Improving Educational Game Design Methods: A Rubric to Assess the Engagement and Educational Value of Educational Games

Ken Hoff
University of Colorado at Boulder
kendall.hoff@colorado.edu

December 1, 2013

Abstract

Existing educational games lack the combination of education and engagement. The objective is to research, synthesize, and test a more effective educational design method. The procedure is to research current literature and existing effective educational games, synthesize an educational game design rubric based off of this research, then apply this method to known educational games. Amazon's Mechanical Turk will be used to gather large amounts of data about the consistency of this rubric, after which the data will be collated and analyzed.

Contents

List of Figures

Chapter 1

Introduction

Educational games lack the combination of education and engagement. In ineffective games, the content being taught is either uninteresting or unrelated to the gameplay. A good example is Math Blaster; players answer simple arithmetic questions to fill up their energy, which they then use to shoot space debris. These games are usually just homework problems, either included as an unrelated component inside the game, or interspersed between gameplay segments in a quiz format. It seems that game designers don't know how to produce effective and fun educational games. They are unaware or unfamiliar with the attributes their educational game should or shouldn't have in order to be as effective and fun as possible.

1.1 Thesis Statement/Solution

I'm going to synthesize an education game design method by creating a list of attributes that educational games should have a high degree of in order to be effective and engaging. The list is not a set of standards, but a set of guidelines; it is possible (though unlikely) for a game to contain all of the attributes and still be ineffective, or for a game to have none of the attributes and be very effective. It is simply meant to infer that game designs that contain these elements are more likely to be effective and engaging educational games.

The question to be answered in this research are the following: Are there attributes that an educational game should have in order to be effective and engaging? In this work, I will develop a rubric of game attributes that educational games should have, and investigate whether some selected educational games incorporate these attributes. In addition, I will test the effectiveness of the selected games' ability to educate the players.

1.2 Approach

I'll begin by examining current educational game design methods, through research of published literature. I'll then synthesize my own educational game design method, a list of attributes found in effective and engaging educational games, influenced heavily by my research and findings. Afterwards, I'll attempt to apply my method to existing educational and semi-educational games, and see if the most effective educational games correlate to my

design method. In contrast, ineffective educational games won't correlate at all to my design method.

1.3 Organization

The rest of this thesis is organized as follows:

Chapter 2 contains background research relevant to this field of research. It explores existing educational frameworks and research that influenced the selection of design elements for the rubric.

Chapter 3 contains the design rubric that games are scored on. It contains 13 game design elements, each of which have been synthesized from the resources in the Research chapter. Each game design element may be found in existing games, to varying degrees described herein.

Chapter 4 contains background information on how our example games were selected, as well as a listing for each game in our study. The game's overview and educational content is described, and screenshots, sources, and notes on implementation are included for each.

Chapter 5 contains all information related to the procedure of our study. It includes how Amazon's Mechanical Turk works, as well as design decisions for the content and process of administering the survey. It also includes the survey that was administered, as well as all permutations and additions.

Chapter 6 contains the visual aggregated results of the administered survey, as well as non-statistical observations on the data by myself. It includes results from the game opinion, quiz, and rubric sections of the survey.

Chapter 7 contains the statistical analysis of the prominent parts of the survey, the quiz and rubric. We use the t-distribution to determine the statistical significance of the quiz score differences, and use Inter-rater reliability to determine the cohesiveness of our design rubric.

Chapter 8 contains conclusions that we've explored in the Results and Analysis sections, as well as a retrospective and future options for additional research.

Chapter 2

Research

2.1 Edutainment vs Educational Games

It's also important for us to differentiate between edutainment and educational games. For this paper, we're going to define edutainment as the simple gamification of a task. The purpose of edutainment is to increase the player's skill at something by getting them to play a game; the player mimics or actually does the skills within the game, and the game rewards them and keeps them engaged in order to continue increasing their skill. These kinds of games, while effective, are ineffective at teaching complex concepts and systems. For that purpose, we use educational games; they have the more rewarding task of teaching complicated content, but it is far more difficult to do so, as they can't utilize the skill and drill method of education. We're going to focus primarily on educational games in this paper.

2.2 Bloom's Taxonomy

Bloom's Taxonomy is one of the most widely cited classifications of learning. It provides milestones for three different domains of learning: Psychomotor, Affective, and Cognitive.

The first domain, Psychomotor, deals with the ability to perform intricate physical tasks, like hammering a nail or operating a complex machine or instrument. Games are currently very good at teaching concepts within the Psychomotor domain. The best examples are ones where the game provides a near-simulation to the task being performed in real life. Music games like Guitar Hero and Rock Band improve coordination and motor control while playing an instrument. Simulation driving games like Forza and Gran Turismo provide players with continuous feedback to help them learn the correct response to stimulus given the car they are driving, to help them improve their driving skills. Other examples include any game that improves motor control; Kinect's Dance Central improves player's bodily movements, and even first person shooter games like Halo or Call Of Duty improve player's hand-eye coordination.

The Affective domain is one that is far harder for games to address. It deals with learning empathy, emotion, and includes things that are commonly learned through interaction with people and not necessarily through formal education. Games commonly expose players to this in two ways. The first is through game narrative, where players are exposed to char-

acters, situations, and choices where they observe and learn about the human condition. Games with extreme character development (Mass Effect, The Walking Dead) typically explore human choices in great detail. There are also extremely minimalistic games that communicate powerful empathic ideas without a large amount of gameplay or graphics, or conversation (Thomas was alone, Journey), as well as completely narrative-based, choose-your-own-adventure Twine games that address extremely empathic concepts (Howling dogs). The second is through putting players in situations with other players where they will need empathy and skills from the Affective domain in order to solve their problems. All multi-player games facilitate this to some degree, but puzzle games (Portal, LittleBigPlanet), and MMO's (WoW, EVE Online) generally encourage players to communicate, negotiate, and work together to achieve a common goal.

When we think of educational games, we commonly think of the Cognitive domain. The Cognitive domain deals with knowledge of facts, associations, and mechanics, and typically consists of the material that educators attempt to teach in schools. This is also the domain that games attempt to address the most; being able to communicate these concepts effectively would reduce the amount of material educators would need to teach, allowing them to focus on more complex concepts. Because games in the Psychomotor domain are well traversed, and games in the Affective domain are far more nebulous as to measuring their educational value, we will be focusing exclusively on the Cognitive domain.

2.3 Tangential Learning

In their paper on Serious Games and Learning, Breuer and Bente (2010) outline a method called Tangential Learning. Originally presented as part of the Extra Credits video series online (Floyd and Portnow, 2008), the presenter gives us a method for which we can encourage significant learning using games. The theory is that, given the proper incentive, players will engage in a form of self-education separate from the game environment.

They base this on the principle that we're able to recall and understand material that we're more interested or passionate about, rather than material that is uninteresting or boring to us. The same principle applies to video games; if we are interested in the context of an educational game, we're far more likely to remember that instead of the content being taught as part of it. The example Extra Credits uses is geography; it's easy for some players to draw the entire map of Azeroth from memory, but they're unable to properly identify US states.

The most prominent field that this could be applied to is history; as most games are set in the past, such as Ancient Rome or World War II, it would be easy for players to learn more about history on their own simply because they were playing the game in the appropriate context. There are other contexts that players can be placed in, in order to educate about other topics. Players in music games can self-educate on music theory and different musical genres, and players in space simulations might self-educate on spaceflight and astronomy.

However, there are numerous other ways players can be educated apart from being placed in the context of the educational goal. An easy-to-implement way would be to include facts as part of the loading screens in the game; instead of having game tips or strategies, designers could include interesting bits of external information to help players retain information. Also,

designers can include subtle references to real-life objects within their games. Extra Credits gives the example of Sephiroth from Final Fantasy, or the Excalibur and the Masamune. By naming characters or objects after real-life characters or objects, designers can encourage players to go out and seek the origin of the character or object; it also helps to have once object in a group be something that's easily recognizable, so that players immediately know that other objects in the group are referential to the real world.

Another excellent way to integrate tangential learning into games is by including an in-game encyclopedia, where players don't even have to leave the game to explore more about the topic they're interested in. Usually, these encyclopedias include information related to playing the game; the stats of an object, or the rules of the game. However, some include information that isn't related to playing the game, like the history or origin of an object or character. A notable example of this is the game Civilization, which includes the Civilopedia, a reference to every unit, building, and wonder in the game, as well as other information related to it. An easy way for games to integrate this would be just including Wikipedia links to every object in the game that has real world roots.

2.4 Flow

Dondlinger (2007) explains why it's extremely important to give players the perception of free will. In educational games, we don't want to give the players no direction as to where to go within the game; we risk players missing the educational goals, and worse, becoming frustrated with the lack of progress. Conversely, we don't want to overly guide the players; too much hand-holding results in the game stifling creativity, as well as not allowing players to really learn by doing. It's important for us to give them the illusion of having an open world, where almost any actions are possible, in order to allow players to experiment and learn by doing.

In parallel to the illusion of free will is the concept of flow.' In games, we want to the player to experience just the right difficulty; games that are too hard will be frustrating, and games that are too easy will be boring. The same is true of education and educational games; material that is far beyond the comprehension of the learner won't be retained, but material that the learner has already covered sufficiently will be boring. We need to maintain a level of difficulty to our game that Csikszentmihalyi (1997) calls flow.' When players are in a state of flow,' the game provides to them a clear set of inputs and outputs; material that they are somewhat familiar with, but also obscure enough to encourage to players to try several options in order to solve the problem.

Following the concept of flow,' we can extrapolate the concept of adaptive difficulty.' It's logical to assume that games that continually adapt their difficulty level to the players will be more effective in retaining the player's interest and educating the player than games that have a fixed difficulty curve. However, not every game can implement adaptive difficulty effectively; games need the players to continuously attempt problems and give feedback on how well they're doing to ascertain how well the difficulty matches up to the player.

2.5 Play

Similar to the illusion that players have a seemingly limitless game world to interact with is the concept of play.' Paras (2005) defines a world where play can happen simply as a world with a series of constraining rules. We don't want the rules to be too constraining, because then we inhibit creativity and discourage alternative solutions to problems, but at the same time, we want the rules to be constraining enough to guide the player towards the solution and mimic the rules of the real world.

The way the game guides players around these rules is extremely important. If the game gives the player copious amounts of tutorials and guidance as to what they can or can't do in the game, the player won't learn them as well. This isn't to say that the player should be encouraged to break the game rules, but instead to encourage the player to explore the world's rules; they won't understand what will and won't work unless they encounter it on their own and are enabled to learn from it.

The game also shouldn't penalize the player for exploring the boundaries of the game world, as it would discourage them from trying new things and discovering additional boundaries. However, we still need to encourage players to solve problems within the game world without breaking any boundaries, just to ensure that they understand the world that they're playing in; that's why rewarding the players for not breaking any rules works much better than penalizing the player for breaking rules.

Chapter 3

Rubric

What follows is a list of properties that make up the educational game design, created by researching and synthesizing properties of existing educational games. Games are not required to have all or any of these properties, but the theory is that games with more of these properties will be considered more effective and engaging educational games.

3.1 Location of Game Encyclopedia

Some educational games include a game encyclopedia as part of the game, either internally or externally as a game manual or wiki. Having such an encyclopedia available within the game greatly increases the chance of players using it for self-directed and self-motivated learning, as part of tangential learning.

1 Game contains no encyclopedia of game content

2 Game information is located online, in a non-central location

3 Game has an outside manual (or wiki) of game content

4 Game doesn't have an in-game encyclopedia, but points to a central location elsewhere

5 Game has an in-game encyclopedia of game content

Examples Examples of games with an encyclopedia with a good location include the Civilization series (for their Civilopedia) and the Total War series.

3.2 Content of Game Encyclopedia

Having a game encyclopedia as part of the game encourages self-motivated players to seek out additional help. If the encyclopedia includes more than just game mechanics as part of its content, players may be more encouraged to self-educate themselves about those topics.

1 Game contains no encyclopedia or Encyclopedia contains no content

2 Encyclopedia contains content only related to game mechanics

3 Encyclopedia contains information about game mechanics, and also limited historical/-factual information

4 Encyclopedia contains content related to game mechanics and historical/factual information

5 Encyclopedia contains content related to game mechanics and historical/factual information and outside links or references

Examples Examples of games with an encyclopedia that contains both mechanics and historical/factual information include Civilization and the Total War series, as well as most historical simulation games.

3.3 Amount of Referential Material

By having many objects or events in the game that are references to real-life objects or events, players may recognize certain objects or events and self-motivate themselves to learn more about those objects or events. More objects like that in the game means more chances for a player to recognize a real-life object or event.

1 No game objects or events are references to real-life objects or events

2 At least one event or object is a reference to a real-world event or object

3 Several non-connected events or objects are references to a real-world event or object

4 At least one group of objects or events are references to real-life objects or events

5 Numerous groups of objects or events are references to real-life objects or events

Examples

3.4 Popularity of Referential Material

Having real-life objects or events in the game means that some players will recognize them and attempt to self-educate themselves about those objects or events. In addition, having a large amount of those references be popular means that more players will recognize the more popular objects/events, and by association, education about the lesser-known objects/events will be the result of self-motivated education about the popular items.

1 Existing references are extremely obscure

2 Some of the references are popular

3 About half of the references are popular

4 Many of the references are popular

5 Most or all of the references are popular

Examples

3.5 Rewards for knowing Referential Material

If games allow players to use their knowledge of the referential material in a positive manner within the game, it reinforces the desire for the player to tangentially learn about all the referential material in the game. This is in contrast to traditional methods, where the content that the player is trying to be educated about serves as a barrier of entry to the later levels of the game, or the game penalizes players that don't know the content; with this, players can still play the entirety of the game, but are incentivized to do better by learning the referential material.

1 Knowing the referential material is purely irrelevant; doesn't affect the gameplay at all

2 Knowing the referential material is a little useful; only affects the gameplay a small amount

3 Knowing the referential material is somewhat useful; moderately affects the player's choices during gameplay

4 Knowing the referential material is very useful; usually affects the player's choices

5 Knowing the referential material always significantly affects gameplay

Examples

3.6 Adaptive Difficulty

In order to incorporate flow as much as possible, the game must use a form of Adaptive Difficulty; the game tries to match the difficulty of the obstacles to the skill level of the player. This can be done in a rudimentary way by allowing the player to select the difficulty themselves, but is most effective when it recognizes how well a player is doing on a certain

puzzle and adjusting the difficulty automatically. This ensures that the player isn't trivially challenged, but at the same time isn't pushed outside the bounds of their ability.

1 Game only has one difficulty

2 Game has several difficulties, but players can only select difficulty at beginning of game

3 Game has several difficulties, and players can change difficulty mid-game

4 Game has several difficulties, and prompts the player to increase or decrease the difficulty as needed

5 Game has several difficulties, and automatically adjusts the difficulty of the game as needed

Examples

3.7 Contextual Tutorials

In keeping with flow, players need to be challenged at their level of skill within every game. This means that players shouldn't be hindered by challenges that are too easy for them, but likewise shouldn't feel completely lost on difficult challenges. The method of introducing these challenges is equally important; players that are forced to go through numerous, easy introductory levels will quickly become bored, but players that skip through tutorials at later stages or forget about valuable information learned early in the game quickly become frustrated. It's important to give the player only information that they need; similar to the concept of adaptive difficulty, the game gives the player helpful hints or suggestions when they are stuck, or when the player doesn't intuitively grasp a new mechanic the first time. Similarly, there needs to be tutorials for just about every concept in the game, to cater to players that might have learning 'gaps' in their history.

1 Tutorials aren't given in the game

2 Tutorials are given at the beginning of the game

3 Tutorials are given every few levels/sections of the game

4 Tutorials are given at the beginning of every level/section

5 Tutorials are offered continuously

Examples

3.8 Low game resource penalty for failure

In order for the game to encourage exploration and play, the player needs to feel free to fail. This means that if the player does fail, there is a minimal amount of game resources (e.g. lives, tries, health points) consumed each time the player fails. With this, the player is comfortable trying different solutions to problems within the game without much penalty.

1 Game resource penalty is extreme (e.g. restart the game)

2 Game resource penalty is large (e.g. 1/3 lives, 10 health points / 100)

3 Game resource penalty is moderate

4 Game resrouce penalty is small

5 Game resource penalty is nonexistent (e.g. unlimited lives)

Examples

3.9 Low "reset time" for failure (time to return to failure point)

In order for the game to encourage exploration and play, the player needs to feel free to fail. This means that if the player does fail, there is a minimal amount of external resources (e.g. time, frustration) consumed each time the player fails. With this, the player is comfortable trying different solutions to problems within the game without much penalty.

1 Reset time is very long (e.g. reloading level takes a long time)

2 Reset time is long (e.g. greater than 10 seconds)

3 Reset time is moderate (e.g. up to 10 seconds)

4 Reset time is short (e.g. a few seconds)

5 Reset time is very short (e.g. near instant)

Examples

3.10 High checkpoint frequency

It's important to have players feel free to fail within games. Having a high frequency of checkpoints reduces both the amount of time/frustration that it takes for the player to return to the point of a failure, as well as in-game resources.

1 Zero checkpoints

2 Checkpoints are few and far between (e.g. levels are the only places to restart)

3 Checkpoints are moderate

4 Checkpoints are numerous (e.g. players can restart at the beginning of each puzzle)

5 Checkpoints are frequent (e.g. players can restart part of the way through puzzles)

Examples

3.11 Freedom of exploration

If players are to be encouraged to explore, the nature of the game must be nonlinear and allow for players to progress in varying sections of the game at their own pace. That isn't saying that they shouldn't have a finite goal to work towards, but the methods or paths with which they work towards that goal should be optional and variable.

1 Players are placed in a strictly linear world or lesson progression

2 Players are allowed just a few large-scale choices in their game world

3 Players have the option to make choices about the direction of their progression in the world, but it is largely linear

4 Players can choose from many choices within the game world to explore, including lessons

5 Players are free to choose the direction they want, both educationally and within the game world; allowed to jump between parallel lessons

Examples

3.12 Iterative feedback

In the case of games with a high degree of free-form exploration, it's easy to see that some players might get lost and become frustrated. In order to maintain flow, these players must receive some kind of feedback that they're on the right track, and not just wandering aimlessly through the game world or puzzles. It's extremely helpful for players to receive feedback at every step they take, even during the completion of puzzles; that way, they can be sure that they're progressing, and easily see the steps ahead of them, enabling flow.

1 Game gives no feedback other than high-level progression through the game

2 Game gives feedback after each level

3 Game gives feedback at various points through a level, after a series of puzzles

4 Game gives feedback after each puzzle

5 Game constantly gives feedback (e.g. during a puzzle)

Examples

3.13 Unorthodox problem solving

Another important element of allowing players to explore and play is to give them the option to solve problems using unorthodox solutions; solutions that the game didn't anticipate, but still validate the puzzle's end condition. Sometimes, the player may be able to figure out how to circumvent certain sections of a puzzle; this should still be considered valid, but it's hoped that the circumvention wasn't because of faulty game logic. This allows players to try unusual solutions to problems, and help them understand the system far better than if they had just been taught the correct solution.

1 There is only one way to solve any given problem, with one given progression that is valid as a solution

2 Some problems within the game may be solved more than one way

3 Multiple solutions are available for each problem, but players are limited to using one of those solutions

4 Players can solve each problem via any solution, so long as they do not circumvent it

5 Players can solve a problem any way they like, or even circumvent the problem, and be given full (or bonus) points

Examples

Chapter 4

Games

4.1 Game Selection

In order to test our rubric, we have to have testers play some educational games. We looked for games marketed as educational games, as well as games that could have been considered educational by including tangential learning. We selected the games from existing knowledge, as well as from references in existing literature.

4.1.1 Web games

Due to the EULA of Mechanical Turk, we could not ask Workers to install any software on their machines as part of our task. This meant that we couldn't select some popular educational games, like Typing of the Dead, because users would have to install the game on their machine.

An option that was considered was having Workers find a YouTube or other video of the game being played, and try to evaluate the game's properties based on the gameplay footage. However, we decided that Workers wouldn't be able to evaluate some of the game's properties (e.g. penalties for failure, or contextual tutorials) based on a gameplay video.

Thus, our primary selection criteria for educational games was that they had to be web-based. The players could not be required to install any software onto their machine in order to play the game. This left us with many interesting and high-quality options, like Light-Bot and Darfur is Dying, but it eliminated some other high-quality options like Logical Journey of the Zoombinis.

It's important to note that we were allowed (or at least, didn't run into any problems) to ask users to install web browser plugins in order to play games. Some of these plugins were commonplace (like the Java runtime), but others were obscure (like the Apple II emulator). This allowed us to find some popular educational games usually only available as a desktop applications, like The Oregon Trail or The Incredible Machine, even if they were very old versions of the games.

4.1.2 Math Blaster

One of the most prominent games that we were unable to include in our survey was the Math Blaster series. Math Blaster's The Search For Spot, released in 1994, was the earliest version of the game, and the franchise continues strong to this day. More editions and expansions were released, all as installable desktop applications.

Recently, the Math Blaster franchise has transitioned into a web-based format, where players can play various Math Blaster games online. This would have been perfect for our study, but the games exist behind a 'login wall,' where players must create accounts and verify their email addresses before they are allowed to play the games. We assume this violates the EULA of Mechanical Turk, but the login wall presents such an accessibility obstacle for Workers that we decided to omit Math Blaster altogether.

4.1.3 Where in the World is Carmen Sandiego?

Another extremely popular and highly-acclaimed educational game series, Where in the World is Carmen Sandiego, started releasing desktop games in 1985, but has since ceased producing sequels. Online, we found numerous emulators for the original game, but the performance and framerate for each of them was very poor. We decided that the average Worker would find it unplayable, and decided to omit it from the study.

4.1.4 Old Games

It's important to note that when we think of good, educational games, we commonly think of classic games; games that were created early in the lifecycle of educational games, and were highly acclaimed for their educational content. Recently, there haven't been many games that have been popularized for their educational content.

Why is that? There's a couple ways that we could explain the lack of popular educational games. It's possible that there exist as many or a greater number of excellent educational games presently, but due to saturation of the gaming industry (both as entertainment and educational games), educational games don't get as much visibility as they do before. For example, the most popular recent game included in our survey, Darfur is Dying, had over 800,000 plays in 6 months and was covered in the media, but received no 'educational game' awards.

Another option is that the nature of education in gaming is changing. Instead of players seeking deliberately edcuational games, where education is the primary goal of the game, players may be more interested in games where education is the secondary goal. Games with education as the secondary goal most likely use tangential learning in order to teach players without deliberately lecturing or quizzing them. Light-bot, one of the games in our survey, is a great example of this.

4.2 Game Listings

4.2.1 Darfur is Dying

URL `http://www.darfurisdying.com/`

Description Darfur is Dying, developed by mtvU, is a activism game released in April of 2006. The game consists of two main sections. In the first section, players must select a member of a Darfuri refugee; the family consists of a male, female, and several children. Once the player has chosen, the player must guide the refugee to a well using only compass and distance directions, while attempting to avoid Janjaweed militia patrols in trucks. If the player is caught, the game describes the fate of the refugee, and the player is prompted to select another refugee. Once the player has made it to the well, the player must return through the same section to their encampment, but lose water during their journey. Once they have returned to the encampment, the player enters a top-down strategy-like game simulation; they can use the water they have retrieved to grow crops and keep the encampment in good condition. However, if they run out of water, they will need to return to the first section and make the well run again. The goal of the game is to keep the encampment alive for 7 days. In addition, the community is constantly under threat from attacks from the militia; if the militia attacks, the encampment is lost, and the player must start again. The player can prevent attacks from the militia by participating in various viral and advocacy campaign tactics, such as inviting their friends to play, posting on social media, or writing to government officials.

Educational Content The educational content of Darfur is Dying is centered almost entirely around awareness. Through playing the game, players learn about the nation of Darfur, including its history, wars, environment, climate, and people. The players are forced to be aware of the troubles that plight Darfuri residents and families, such as militia attacks. Players are encouraged to read the backstories of every man, woman, and child, as well as stories associated with locations within the village that expose various events that have happened, such as being unable to fend off sickness without medical aid, or when a militia recently stormed the village and murdered numerous people.

Notes on Implementation Darfur is Dying was built in Flash by interactive media agency interFUEL.

4.2.2 Light Bot

URL `http://armorgames.com/play/2205/light-bot`

Description Light Bot, made by Danny Yaroslavski, is a programming and robotics puzzle game. It was originally a flash game, but has since been ported to iOS and Android. Players assume control of a robot on a grid of varying sizes and orientations. Each grid square can also have a height. The robot has the ability to move forward, turn left or right,

Figure 4.1: Darfur is Dying's title screen

Figure 4.2: A screenshot from Darfur Is Dying's village management section

jump up one level or down one level, and turn a square "on." The robot also has the ability to call "functions," where the robot can execute sequences of events and repeat functions several times or indefinitely. The goal of the robot is to navigate to all of the blue squares and turn them "on" to a yellow state. The robot can do this in any order and using any sequence they like, so long as it fits within the provided instruction spaces. There are 40 levels to the game, ranging from the simple to extremely difficult.

Educational Content Light Bot's educational purpose focuses on teaching programming at a very simple level. Players learn that the bot will follow sets of instructions. Initially, these instructions will be very simple (e.g. forward, turn, blink), but the player will realize quickly that the bot will follow the instructions explicitly, even if they do not solve the puzzle. This teaches players that computers are very powerful but very simple machines, and will do exactly what they are directed to do, even if it's not what the programmer intends to do. The game also teaches the concepts of functions and loops; players can "call" predefined functions numerous times, as well as have a function call itself to loop the function indefinitely. These programming concepts, while simple, are a wonderful introduction to programming for students.

Notes on Implementation Light Bot was built in Flash by Danny Yaroslavski, and the iOS and Android versions were built using Haxe 3 and OpenFL.

4.2.3 Pandemic 2

URL `http://www.crazymonkeygames.com/Pandemic-2.html#game`

Description Pandemic 2 is a flash-based strategy game involving infectious diseases, viruses, and bacteria. The player is in charge of designing and mutating an infectious organism, which infects the world population. The objective is to have the infection spread to and kill every human being on the planet, rendering the human race extinct. The virus starts out as being only mildly visible, lethal, and infectious, and can be mutated to more effective versions through "upgrades," received as more humans are infected and die. To combat the spread of the infection, world nations begin to close their borders, set up quarantines, and close off trade routes, cutting off the transmission of the infection to their nation and making it more difficult or nearly impossible for the disease to spread. The player typically alters between the disease upgrade screen and the world monitoring screen, which includes notable headlines and the statuses of the nations. Global high scores are given to players that successfully eliminate the human race in the shortest amount of time.

Educational Content Pandemic has two limited educational aspects to it. The first is the notion of learning about infectious diseases and organisms. While there isn't much science within the game behind mutating an organism to be more deadly, there are plenty of terms and game mechanics that the player can familiarize themselves with, such as organism's resistance to humidity, or how airborne diseases differ from waterborne. There's also an element of strategizing, risk-taking, and planning ahead associated with playing the game;

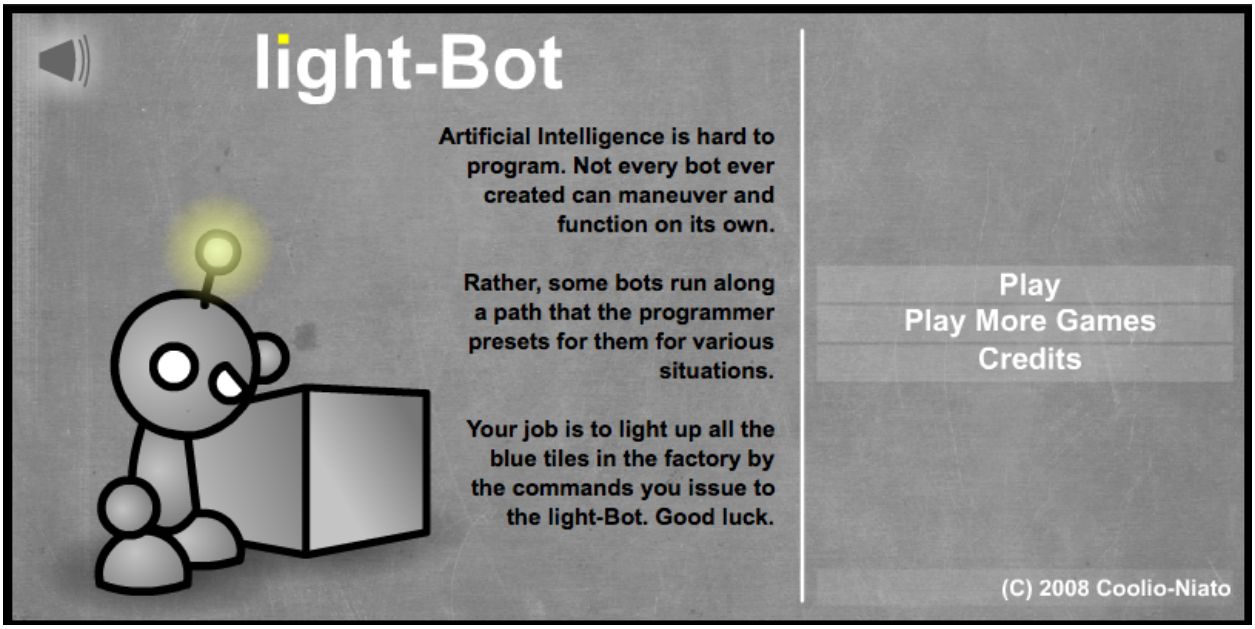

Figure 4.3: Light Bot's main menu

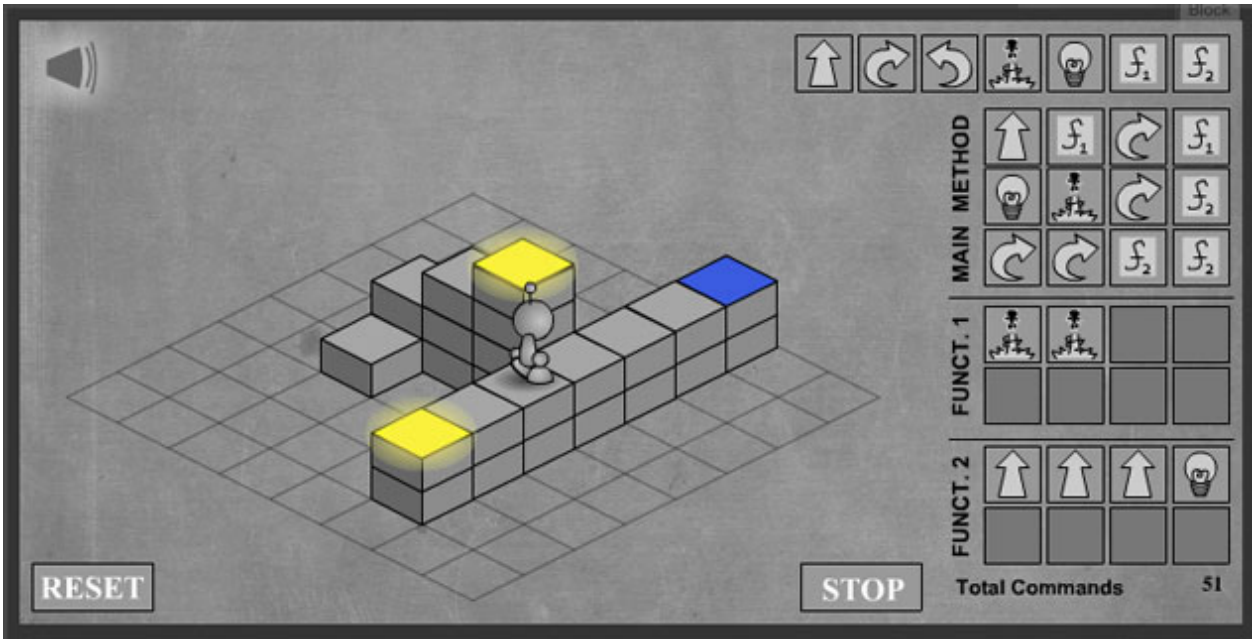

Figure 4.4: A screenshot from one of Light Bot's levels

if players run into a problem with a certain method, they may be able to solve the problem a different way, or circumvent the problem on another playthrough.

Notes on Implementation Pandemic 2 was built in Flash by Dark Realm Studios.

4.2.4 Oregon Trail

URL `http://www.virtualapple.org/oregontraildisk.html`

Description The Oregon Trail is a series of games detailing the experience of a pioneers traveling the Oregon Trail, a wagon route from the Missouri river to Oregon in 1848. The original game, developed for the Apple II in December of 1971, turned out to be a extremely well received by middle and high school students. In the game, the player first selects an identity which determines how much starting money they have. They then select supplies to buy for their journey; wagons, spare parts, oxen, food, and other supplies. Once they embark on their journey, time passes and food is automatically consumed. Occasionally, players have the chance to hunt, where they can play a simple top-down 2D game to shoot animals that provide food to their party. In addition, numerous events will happen that the player needs to make descisions for, such as a party member falling ill, a wagon part breaking or oxen becoming injured, or needing to cross a river. The player's objective is to travel the entire Oregon Trail with using the minimum amount of resources; the player receives more points at the end for having living party members, items in inventory, and number of dollars, as well as receiving a multiplier if they started the game with less cash.

Educational Content Oregon Trail's educational value comes in two forms. The first is the most apparent one; though not explicitly sitting players down and teaching them, the game educates students on life in the 19th century, as well as the hardships and trials endured by explorers of the early Oregon Trail. It teaches them about the kinds of materials that were used in everyday life, like wagons and oxen, as well as the diseases that commonly plagued explorers (diarrhea, dysentery), and what day-to-day activities were like on the trails, such as maintaining the wagons, crossing rivers, and hunting for food. The other educational aspect that Oregon Trail focuses on is planning and risk management; though not explicitly teaching players how to assess the risk of various actions, players who properly evaluate their initial inventory options as well as the options they have during events on the trail will end up doing better than players who don't.

Notes on Implementation This is the very first edition of The Oregon Trail, for the Apple II. It's played in a browser-based Apple II emulator.

4.2.5 Number Munchers

URL `http://wallofgame.com/free-online-games/arcade/988/Number_Munchers.html`

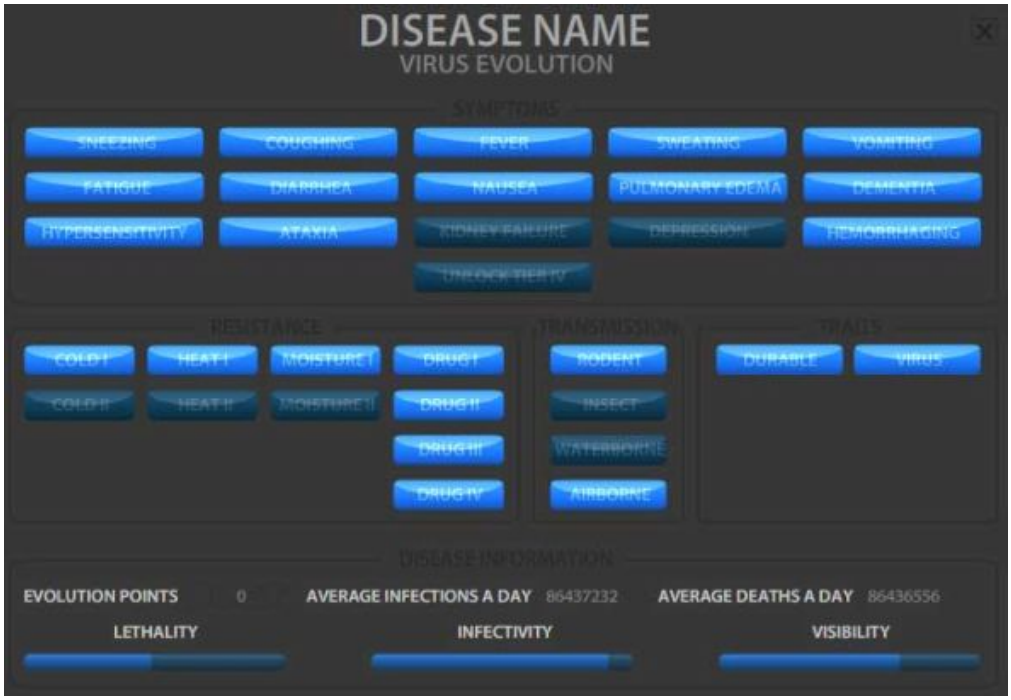

Figure 4.5: A screenshot from Pandemic 2's evolution view

Figure 4.6: A screenshot from the opening sequence of Oregon Trail

Description Number Munchers is part of a "Munchers" series that began as a strictly educational game around teaching children basic arithmetic and other numerical concepts. The series then expanded to teach words and other knowledge. The player first selects the educational content that they'd like to play, including grade level. Administrative controls are available for parents and teachers to restrict what kind of content is available; for example, restricting higher-grade students to their level, instead of allowing them to play at trivial levels. Once in the game, the player assumes control of a green 'Muncher,' which can move up, down, left or right. The game board is a grid, with each space containing a number that is a solution to the level they're playing (e.g. equal to 12, multiples of 3, divisible by 5). The player moves onto the appropriate space, and presses the space key to 'eat' the number. If the number matches the level criteria, the player gets points; if it does not, the player loses one of three lives. The player also has to deal with 'troggles,' which cross the grid slowly at random intervals. If the player occupies the same space as a troggle, the player loses a life and is reset to a different grid space. The level ends once all the appropriate numbers are eaten, and the player recieves a bonus for time.

Educational Content The educational content of Number Munchers is extremely straightforward. Players will learn arithmetic, primes, and other numerical concepts while playing this game. Players who learn the concepts will be able to identify the correct solutions faster, and consequently achieve a higher score while playing this game.

Notes on Implementation This is not an official version of Number Munchers. It's a Flash-based, Number Munchers-inspired game created by user Authorblues. It's functionally equivalent to the original edition of Number Munchers.

4.2.6 BotLogic

URL `http://botlogic.us/`

Description BotLogic is an educational programming game. In the game, the players assume control of a robot whose task is to return home. The game takes place on a grid, with the robot located on one space, the home on another space some distance away, and a number of obstacles in between. The player can direct the robot up, down, left, or right, and does so by queueing up commands before running the program. The player can wait for the program to run, then add more commands before running the rest of the program, which allows players to incrementally build their program. However, the robot has a limited amount of energy, which limits the number of moves the player can take. Later in the game, more obstacles and powerups are introduced, such as electric fences, buttons, and recharging stations. The game contains 20 levels.

Educational Content Botlogic teaches players about simple programming concepts, namely passing a sequence of instructions to a robot and watching them run. However, the game doesn't include any functional or object-oriented programming concepts, and is only slightly abstracted away from the player having direct directional control of the robot.

Figure 4.7: A screenshot from the gameplay of Oregon Trail

Figure 4.8: A screenshot from the gameplay of Number Munchers

Notes on Implementation BotLogic is built in HTML5/Javascript and developed by Dolphin Micro, a web development consulting firm located in Colorado.

4.2.7 Math Baseball

URL `http://www.funbrain.com/math/index.html`

Description Math Baseball is an educational math game. Initially, the player selects what type of arithmetic they'd like to do, as well as the grade level. Then, they assume the role of a baseball player at bat. For each throw, the players are given unlimited time to answer one arithmetic question. If they get the question right, the player earns a randomly selected single, double, or triple and gets a runner on base, as well as advancing any other runners. The player's score is determined by how many runs they get in. If they don't get the question right, then they recieve a strike. After three strikes, the player is "out," and the game ends.

Educational Content The education in "Math Baseball" is very straightforward. Players can learn addition, subtraction, multiplication, and division, with the player's knowledge of the topic reflected in their higher score of the game.

Notes on Implementation Math Baseball is written in HTML and an unspecified server framework.

4.2.8 Notpron

URL `http://notpron.org/notpron/`

Description Notpron is an ARG (alternate-reality game) for the browser. The player begins Level 1 with an image of a house with a partially open door in front, as well as some slightly opaque text that says "Enter the door." The player needs to click the door (not the image, but the door itself) to advance to Level 2. In Level 2, a finger points to the address bar, where the player can replace "level2.htm" with "level3.htm" to advance to the next level. In Level 3, the player must change "false" in the URL to "true" to advance to the next level. The game continues like this, adding in new elements each level. There are a total of 140 levels, and only 31 people have completed all 140 levels, out of about 16 million players.

Educational Content From the Notpron site:

> "[Players who finish the game] have persisted with a broad range of complex ways of thinking, while maintaining focus and dedication over a long period. [Their] detective skills have been tested to the limits, yet the smallest hint proved sufficient to solve the most complicated tasks. Furthermore, competence in the following areas have been displayed: Sound editing, Graphic editing, Musical understanding, Insight into HTML programming, Rapid learning of new programs,

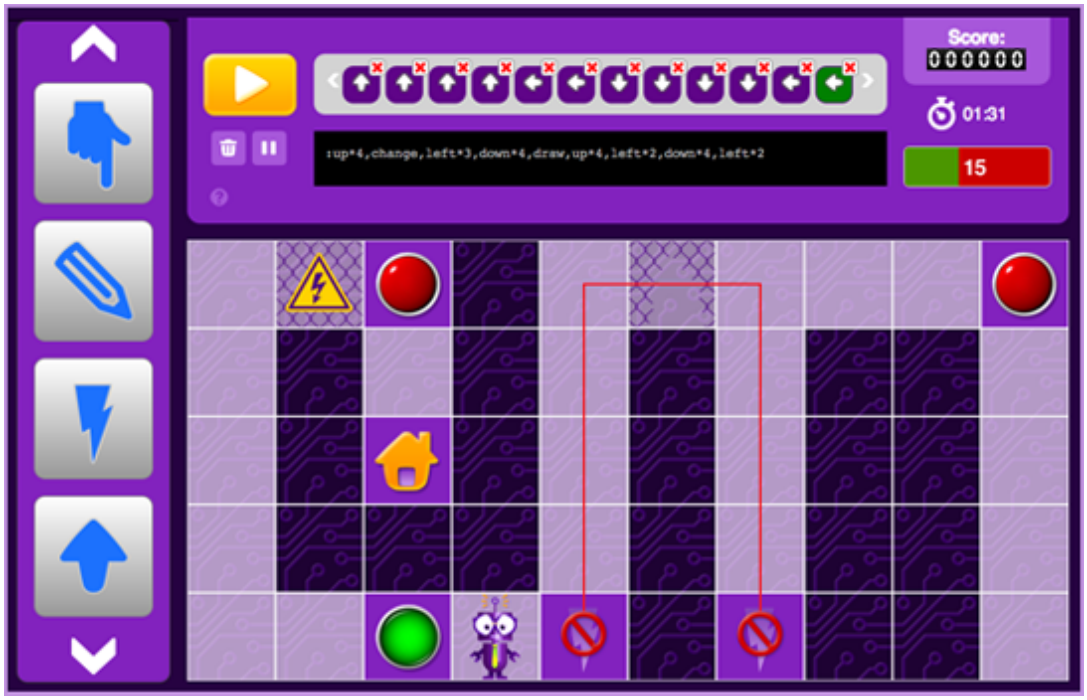

Figure 4.9: One of the levels from BotLogic

Figure 4.10: A screenshot from the gameplay of Math Baseball

Efficient online research techniques, [and] Insight into the complex workings of a computer."

Notes on Implementation Notpron is written in HTML and an unspecified server framework.

4.2.9 Lemmings

URL `http://www.elizium.nu/scripts/lemmings/`

Description Lemmings, a PC game, was originally developed in 1991. The game plays from a 2D side-scrolling perspective. The player directs "lemmings," small, humanoid creatures that are reminiscient of the mammal in their behavior. They operate almost entirely on their own, walking in one direction until they run into something, then reversing direction. They begin by dropping out of the entrance door, and successfully exit the level through another door in the level. However, the lemmings are very susceptible to dying; falling too far without a parachute will kill them, and numerous obstacles litter the courses, such as spike pits and smashers. The game contains four difficulty levels, with roughly 20 levels per difficulty level.

Educational Content Lemmings teaches players problem-solving, multitasking, and resource management. Because a fixed number of lemmings are required to finish the level, players must learn the proper methods for getting around obstacles by using the minimum number of lemmings possible.

Notes on Implementation This implementation of Lemmings is a Javascript port of the original version of the game.

4.2.10 The Incredible Machine

URL `http://www.classicdosgames.com/online/timdemo.html`

Description The Incredible Machine, originally developed in 1993, is a side-view 2D construction game. During a level, the player will have an objective (e.g. get the ball into the basket). The play area will already have some parts set up, so the player can use the parts that they have in reserve to construct the rest of the Rube Goldberg-style machine to accomplish the objective. There were around 80 levels in the game. The game was extremely successful, and spawned numerous sequels and ports.

Figure 4.11: The first level of Notpron

Figure 4.12: One of the levels from Lemmings

Educational Content The Incredible Machine teaches players about physics and problem solving. Players learn about the physical properties of various objects (for example, the tennis ball might not knock down the board, but the bowling ball might), and learn how to use limited combinations of those objects together to solve the puzzles.

Notes on Implementation This is the original edition of The Incredible Machine, run inside a Java container in the browser.

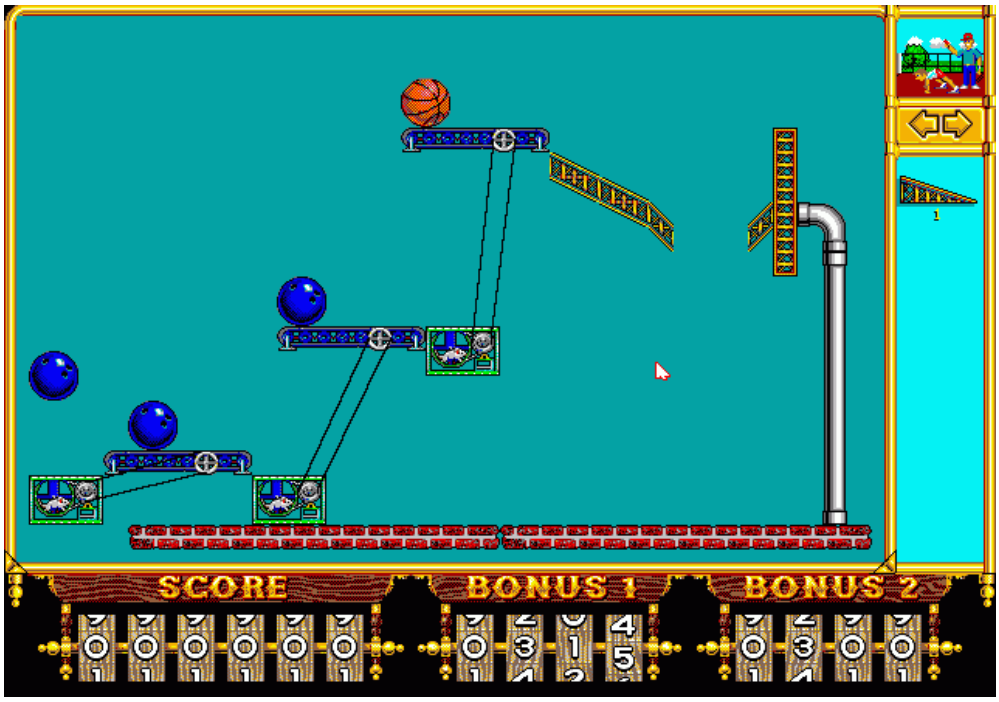

Figure 4.13: One of the levels from The Incredible Machine

Chapter 5

Procedure

5.1 Survey Design

In order to test the effectiveness of the rubric on determining good educational game design elements, it must be used by real players on real educational games. To get the best results possible, it would need to be evaluated by a large number of players across a variety of games.

Our procedure, then, would be very simple. A player would need to play a game for a short period of time, then fill out the survey, indicating what elements the game had. There's no need for additional feedback or contextual observation of the player as they play the game. This means that we can administer our survey remotely, through online quiz or survey-taking systems.

5.2 Mechanical Turk

I decided on Amazon's Mechanical Turk to adminster my surveys. Mechanical Turk is an online marketplace where Requesters (the people who want results from surveys or tasks) can submit HITs (Human Intelligence Tasks) into the marketplace, where Workers (people who fill out surveys or perform other tasks) can complete them, usually receiving some kind of small monetary reward for them. It's commonly used as a survey-taking platform, but Requesters also use it for tasks that computers are not yet good at doing, like determining the content of an image, writing a summary of an article, or finding the website of a lesser-known business.

In order to only allow a worker to take a survey once, I had to set up 10 different HITs, one for each game that we were testing. I also allocated a number of responses that I wanted to get for each HIT; this ensured that a worker would only be allowed to fill out a survey once, but was allowed to complete survey for all of the games if they wanted. I allocated 20 responses for each game, for a total of 200 survey responses. The initial expiration date of the HITs was 3 days later, but I extended the deadline a week.

For 6 HITs, the HIT was composed of the "Basic" survey included below. It

contained some simple demographic questions, some Likert-scale questions about their opinions on fun and educational games, and some freeform text response areas where they could leave feedback about the games and survey. It also included a link to one of 6 games (The Incredible Machine, Lemmings, Notpron, Math Baseball, BotLogic, or Pandemic 2), as well as a section with all of the rubric items, where players were asked to rate each game on all of the rubric items.

The other 4 HITs included a pre- and post-quiz in addition to the basic survey. The 4 games (Darfur is Dying, The Oregon Trail, Number Munchers, and LightBot) each had their own quiz, designed to test the knowledge that the games were intended to teach. The quizzes were 10 questions each, with either 4 or 5 multiple choice answers, or True/False answers. It's important to note that the pre- and post-quizzes were the exact same; the exact same questions and answers appeared before and after the players played and rated the game. The players were not given feedback on how well they did on either of the quizzes. Each quiz is included below.

5.3 Survey

5.3.1 Basic Survey

What is your gender?

- Male
- Female

What is your age?

(An empty text box where workers can only enter numbers)

How many years have you been playing video games?

(An empty text box where workers can only enter numbers)

Fun games can be educational.

- Strongly Agree
- Agree
- Disagree
- Strongly Disagree

Most fun games are educational.

- Strongly Agree

- Agree
- Disagree
- Strongly Disagree

Games are more fun when they are cooperative (instead of competitive).

- Strongly Agree
- Agree
- Disagree
- Strongly Disagree

Games are more fun when you play by yourself (instead of with others).

- Strongly Agree
- Agree
- Disagree
- Strongly Disagree

Games are more fun when you play online (instead of with others in the same room).

- Strongly Agree
- Agree
- Disagree
- Strongly Disagree

Educational games can be fun.

- Strongly Agree
- Agree
- Disagree
- Strongly Disagree

Most educational games are fun.

- Strongly Agree
- Agree
- Disagree
- Strongly Disagree

Games are more educational when they are cooperative (instead of competitive).

- Strongly Agree
- Agree
- Disagree
- Strongly Disagree

Games are more educational when you play by yourself (instead of with others).

- Strongly Agree
- Agree
- Disagree
- Strongly Disagree

Games are more educational when you play online (instead of with others in the same room).

- Strongly Agree
- Agree
- Disagree
- Strongly Disagree

List some fun games.

(An empty text box where workers can write anything)

List some educational games.

(An empty text box where workers can write anything)

5.3.2 Rubric

Does this game have an encyclopedia of game content? If so, where?

1. Game contains no encyclopedia of game content
2. Game information is located online, in a non-central location
3. Game has an outside manual (or wiki) of game content
4. Game doesn't have an in-game encyclopedia, but points to a central location elsewhere
5. Game has an in-game encyclopedia of game content

If this game contains an encyclopedia of game content, what kind of content does it contain?

1. Game contains no encyclopedia or Encyclopedia contains no content
2. Encyclopedia contains content only related to game mechanics
3. Encyclopedia contains information about game mechanics, and also limited historical/factual information
4. Encyclopedia contains content related to game mechanics and historical/-factual information
5. Encyclopedia contains content related to game mechanics and historical/-factual information and outside links or references

How many game objects or events reference real-life objects or events?

1. No game objects or events are references to real-life objects or events
2. At least one event or object is a reference to a real-world event or object
3. Several non-connected events or objects are references to a real-world event or object
4. At least one group of objects or events are references to real-life objects or events
5. Numerous groups of objects or events are references to real-life objects or events

Of the objects/events that are in the game, how recognizable/popular are they?

1. Existing references are extremely obscure
2. Some of the references are popular
3. About half of the references are popular
4. Many of the references are popular
5. Most or all of the references are popular

How much does knowing the real-life objects/events affect the gameplay?

1. Knowing the referential material is purely irrelevant; doesn't affect the gameplay at all
2. Knowing the referential material is a little useful; only affects the gameplay a small amount
3. Knowing the referential material is somewhat useful; moderately affects the player's choices during gameplay

4. Knowing the referential material is very useful; usually affects the player's choices
5. Knowing the referential material always significantly affects gameplay

How many difficulty levels are there in the game, and how is the difficulty changed?

1. Game only has one difficulty
2. Game has several difficulties, but players can only select difficulty at beginning of game
3. Game has several difficulties, and players can change difficulty mid-game
4. Game has several difficulties, and prompts the player to increase or decrease the difficulty as needed
5. Game has several difficulties, and automatically adjusts the difficulty of the game as needed

How often are tutorials offered in the game?

1. Tutorials aren't given in the game
2. Tutorials are given at the beginning of the game
3. Tutorials are given every few levels/sections of the game
4. Tutorials are given at the beginning of every level/section
5. Tutorials are offered continuously

If the player fails, how many game resources do they lose?

1. Game resource penalty is extreme (e.g. restart the game)
2. Game resource penalty is large (e.g. 1/3 lives, 10 health points / 100)
3. Game resource penalty is moderate
4. Game resrouce penalty is small
5. Game resource penalty is nonexistent (e.g. unlimited lives)

If the player fails, how long is the wait until they restart play?

1. Reset time is very long (e.g. reloading level takes a long time)
2. Reset time is long (e.g. greater than 10 seconds)
3. Reset time is moderate (e.g. up to 10 seconds)
4. Reset time is short (e.g. a few seconds)
5. Reset time is very short (e.g. near instant)

How often are there in-game checkpoints?

1. Zero checkpoints
2. Checkpoints are few and far between (e.g. levels are the only places to restart)
3. Checkpoints are moderate
4. Checkpoints are numerous (e.g. players can restart at the beginning of each puzzle)
5. Checkpoints are frequent (e.g. players can restart part of the way through puzzles)

How much freedom of choice is there in the game, including both game world and choice of lessons?

1. Players are placed in a strictly linear world or lesson progression
2. Players are allowed just a few large-scale choices in their game world
3. Players have the option to make choices about the direction of their progression in the world, but it is largely linear
4. Players can choose from many choices within the game world to explore, including lessons
5. Players are free to choose the direction they want, both educationally and within the game world; allowed to jump between parallel lessons

How much and how often is there feedback regarding progress on the game?

1. Game gives no feedback other than high-level progression through the game
2. Game gives feedback after each level
3. Game gives feedback at various points through a level, after a series of puzzles
4. Game gives feedback after each puzzle
5. Game constantly gives feedback (e.g. during a puzzle)

Are problems able to be solved multiple ways, or circumvented entirely?

1. There is only one way to solve any given problem, with one given progression that is valid as a solution
2. Some problems within the game may be solved more than one way
3. Multiple solutions are available for each problem, but players are limited to using one of those solutions

4. Players can solve each problem via any solution, so long as they do not circumvent it
5. Players can solve a problem any way they like, or even circumvent the problem, and be given full (or bonus) points

This game was fun.

- Strongly Agree
- Agree
- Disagree
- Strongly Disagree

I had fun playing this game.

- Strongly Agree
- Agree
- Disagree
- Strongly Disagree

This game was educational.

- Strongly Agree
- Agree
- Disagree
- Strongly Disagree

I learned something from playing this game.

- Strongly Agree
- Agree
- Disagree
- Strongly Disagree

What did you learn from playing this game?

(An empty text box where workers can write anything)

Provide some comments and feedback on the game.

(An empty text box where workers can write anything)

Provide some comments and feedback on this survey.

(An empty text box where workers can write anything)

Estimate the number of minutes that you've played this game.

(An empty text box where workers can only enter numbers)

Estimate your highest score/level from your play session.

(An empty text box where workers can only enter numbers)

5.3.3 Darfur is Dying Quiz

When did the violence in Darfur first start?

1. 2001
2. 2002
3. 2003
4. 2004
5. 2005

Who is the fighting between?

1. two different tribes
2. arabs and non-arabs
3. Hutu and Tutsi
4. Blacks and Whites
5. rich and poor

A genocide is defined as killing, causing serious bodily or mental harm and preventing births of ...

1. a specific national group
2. a specific ethnical group
3. a specific racial group
4. a specific religious group
5. all of the above

How did the conflict start?

1. bombing in a main city
2. government's drafting of soldiers
3. accidental killing of 5 civilians by police
4. rebellion attacks of government targets
5. public protest against new law

How many people have died as a result of the violence, hunger, and disease?

1. up to 3,000
2. up to 30,000
3. up to 300,000
4. up to 3,000,000
5. up to 3.5 milion

Where is Darfur?

1. northern Sudan
2. southern Sudan
3. eastern Sudan
4. western Sudan
5. central Sudan

When was the height of the violence in Darfur?

1. 2003-2004
2. 2003-2005
3. 2004-2005
4. 2004-2006
5. 2005-2006

What group is referred to as Devils on horseback?

1. Sunnis
2. Shiite
3. Tumoils
4. Janjaweed

Darfur is a region in the African country of Sudan. Which countries border Sudan?

1. Rwanda, Somalia, Tanzania
2. Algeria, Morocco, Tunisia
3. Central African Republic, Chad, Libya
4. Kenya, Ethiopia, Malawi

Which two major powers united with the Sudanese government to keep the United Nations out of Darfur?

1. United States and Canada
2. Iraq and Iran
3. Egypt and Libya
4. Russia and China

5.3.4 The Oregon Trail Quiz

How many miles long was the Oregon Trail?

1. 500 Miles
2. 1000 Miles
3. 2000 Miles
4. 3000 Miles
5. 20000 Miles

What is the name of a disease like malaria that the pioneers might catch?

1. ague
2. pneumonia
3. epilepsy
4. measles

What year was the first Oregon Trail wagon train organized?

1. 1847
2. 1843
3. 1899
4. 1876
5. 1836

What disease killed more people on the trail than any other?

1. smallpox
2. plague
3. cholera
4. scarlet fever

When was the first transcontinental railroad finished that eventually ended the Oregon Trail?

1. 1867
2. 1870
3. 1899
4. 1869

How many modern states did the travelers travel through when crossing the trail?

1. 6
2. 8
3. 3
4. 10

How many people died on the Oregon Trail?

1. 50,000-60,000
2. 20,000-30,000
3. 90,000-100,000
4. 10,000-20,000

Where did the Oregon Trail begin?

1. Independence, Mississippi
2. Independence, Missouri
3. Independence, Michigan
4. Independence, Montana
5. Independence, Massachusetts

Where did the Oregon Trail end?

1. Vancouver, Washington
2. The Dalles, Oregon
3. Portland, Oregon
4. Stevenson, Washington
5. Oregon City, Oregon

How many people came west on the Oregon Trail?

1. around 500
2. around 5000
3. around 50000
4. around 500000
5. around 5000000

5.3.5 Number Munchers Quiz

39 + 33 =

1. 67
2. 87
3. 73
4. 81
5. 72

15 + 82 =

1. 112
2. 106
3. 97
4. 81
5. 113

-77 - -94 =

1. 14
2. 36
3. 35
4. 17
5. 10

68 - 92 =

1. -8
2. -15
3. -24
4. -29
5. -38

18 * 9 =

1. 172
2. 162
3. 150
4. 173
5. 157

15 * 8 =

1. 121
2. 131
3. 122
4. 120
5. 119

14 * 6 =

1. 88
2. 71
3. 84
4. 68
5. 72

192 / 16 =

1. 24
2. 18
3. 27
4. 25
5. 12

143 / 11 =

1. 23
2. 13
3. 31
4. 30
5. 14

276 / 12 =

1. 38
2. 42
3. 41
4. 23
5. 20

5.3.6 Lightbot Quiz

What does this conditional evaluate to? (not ((true) and (false))) and (not ((false) or (not false)))

1. true
2. false

What does this conditional evaluate to? ((not false) and (not true)) and ((false) and (not true))

1. true
2. false

What does this conditional evaluate to? not (((not true) and (false)) and ((true) and (false)))

1. true
2. false

What does this conditional evaluate to? ((true) and (true)) or ((false) and (not true))

1. true
2. false

Examine the following code:
int count = 0; while (count <= 6) { System.out.print(count + " "); count = count + 2; } System.out.println( );
What does this code print on the monitor?

1. 1 2 3 4 5 6
2. 0 2 4 6 8
3. 0 2 4
4. 0 2 4 6

Examine the following code:
int count = 7; while (count >= 4) { System.out.print(count + " "); count = count - 1; } System.out.println( );
What does this code print on the monitor?

1. 1 2 3 4 5 6 7
2. 7 6 5 4
3. 6 5 4 3
4. 7 6 5 4 3

Examine the following code:
int count = -2; while (count < 3) { System.out.print(count + " "); count = count + 1; } System.out.println( );
What does this code print on the monitor?

1. -2 -1 1 2 3 4
2. -2 -1 1 2 3
3. -3 -4 -5 -6 -7
4. -2 -1 0 1 2

Examine the following code:

int count = 1; while (count < 5) { System.out.print(count + " "); } System.out.println( );

What does this code print on the monitor?

1. 1 2 3 4
2. 1 2 3 4 5
3. 2 3 4
4. 1 1 1 1 1 1 1 1 1 1 . . . .

Examine the following code:

function foo() { bar(); print('foo'); } function bar() { print('bar'); }

What does the code "foo(); bar()" print on the monitor?

1. foobar
2. barfoobar
3. foobarfoo
4. barbarfoo

Examine the following code:

function foo() { bar(); print('foo'); } function bar() { print('bar'); }

What does the code "bar(); foo()" print on the monitor?

1. foobar
2. barfoobar
3. foobarfoo
4. barbarfoo

Chapter 6

Results

6.1 Opinions on educational and fun games

6.2 Quiz Results

6.2.1 Aggregate Quiz Scores

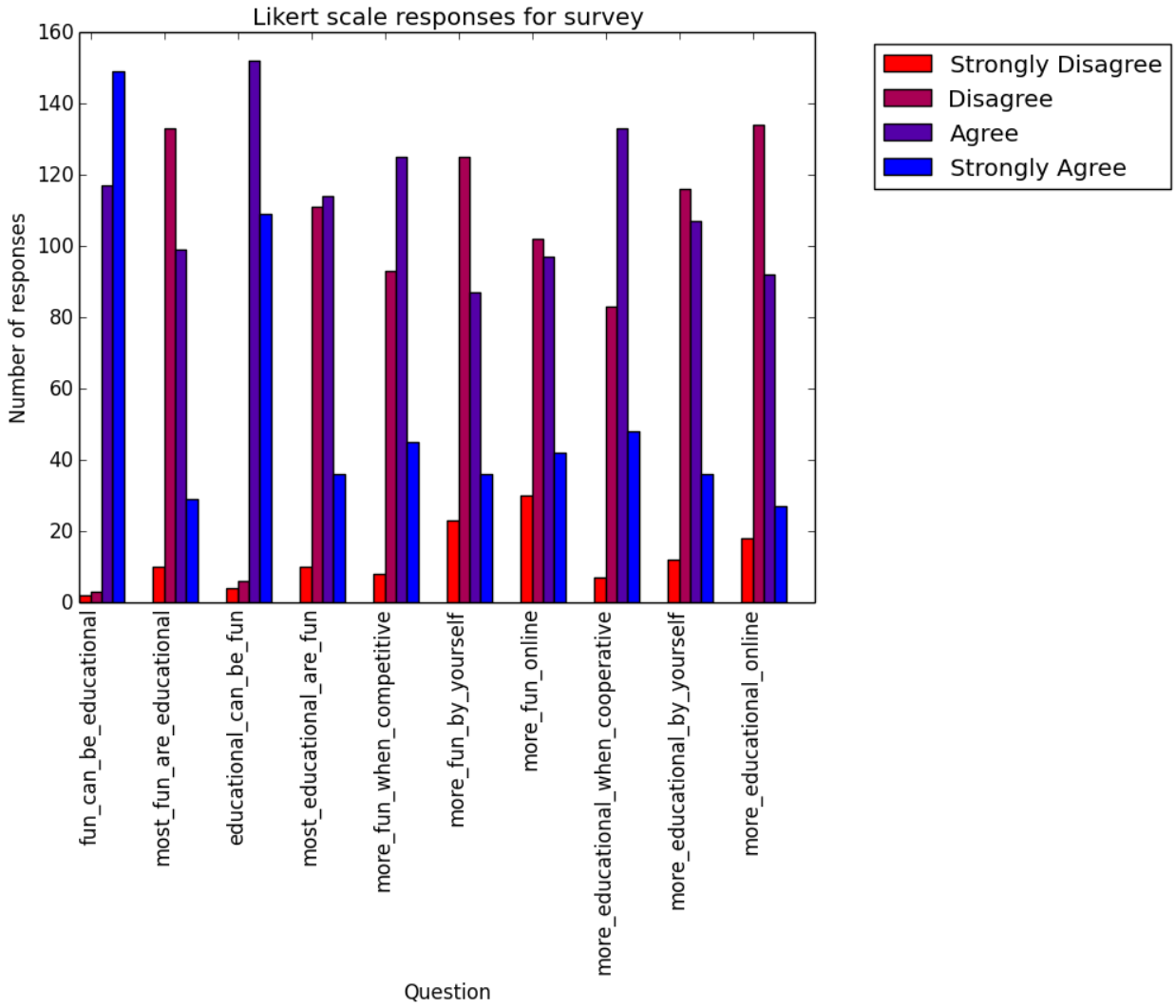

Figure 6.1: Likert scale responses for several questions on fun and educational games.

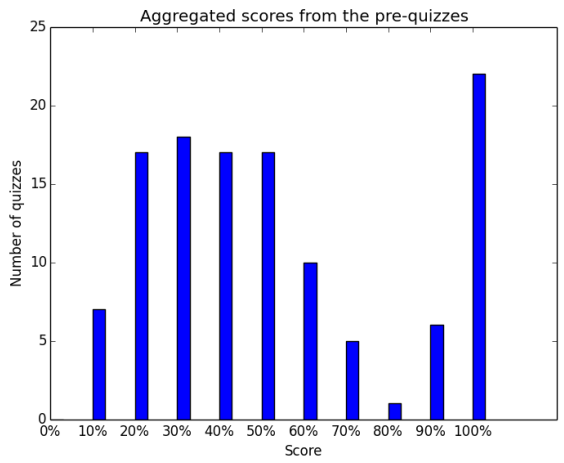

Figure 6.2: The aggregate pre-quiz scores across all games.

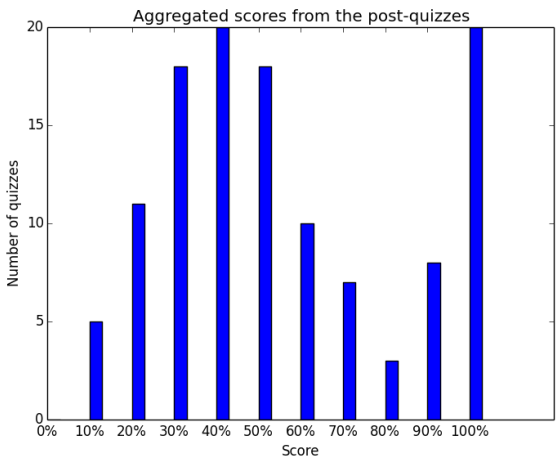

Figure 6.3: The aggregate post-quiz scores across all games.

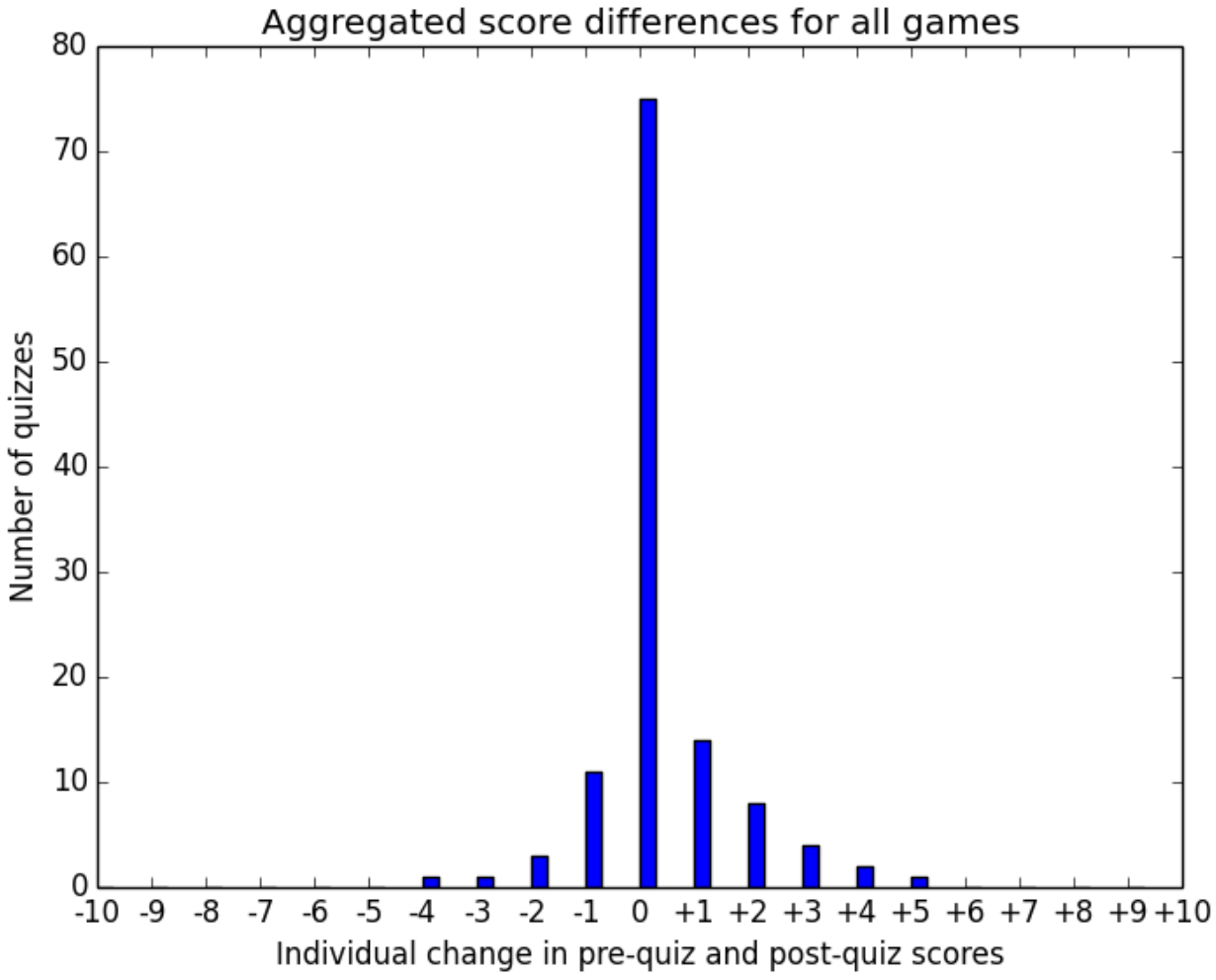

Figure 6.4: The aggregate score differences across all games.

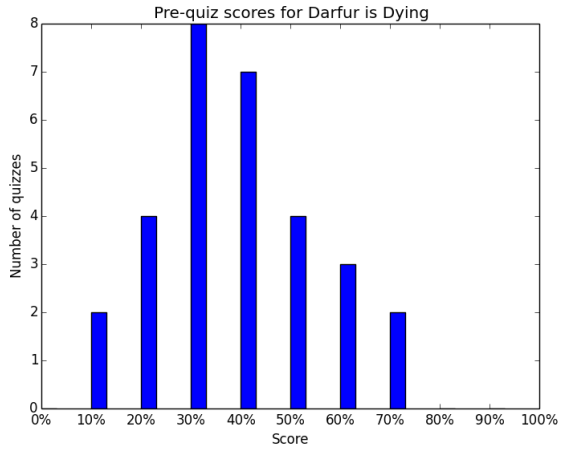

Figure 6.5: The pre-quiz scores for Darfur is Dying.

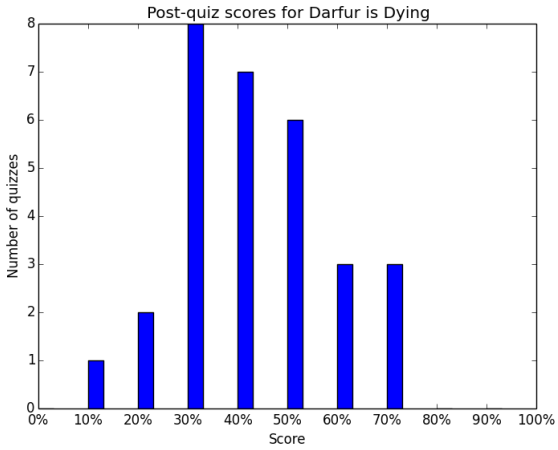

Figure 6.6: The post-quiz scores for Darfur is Dying.

6.2.2 Darfur Quiz Scores

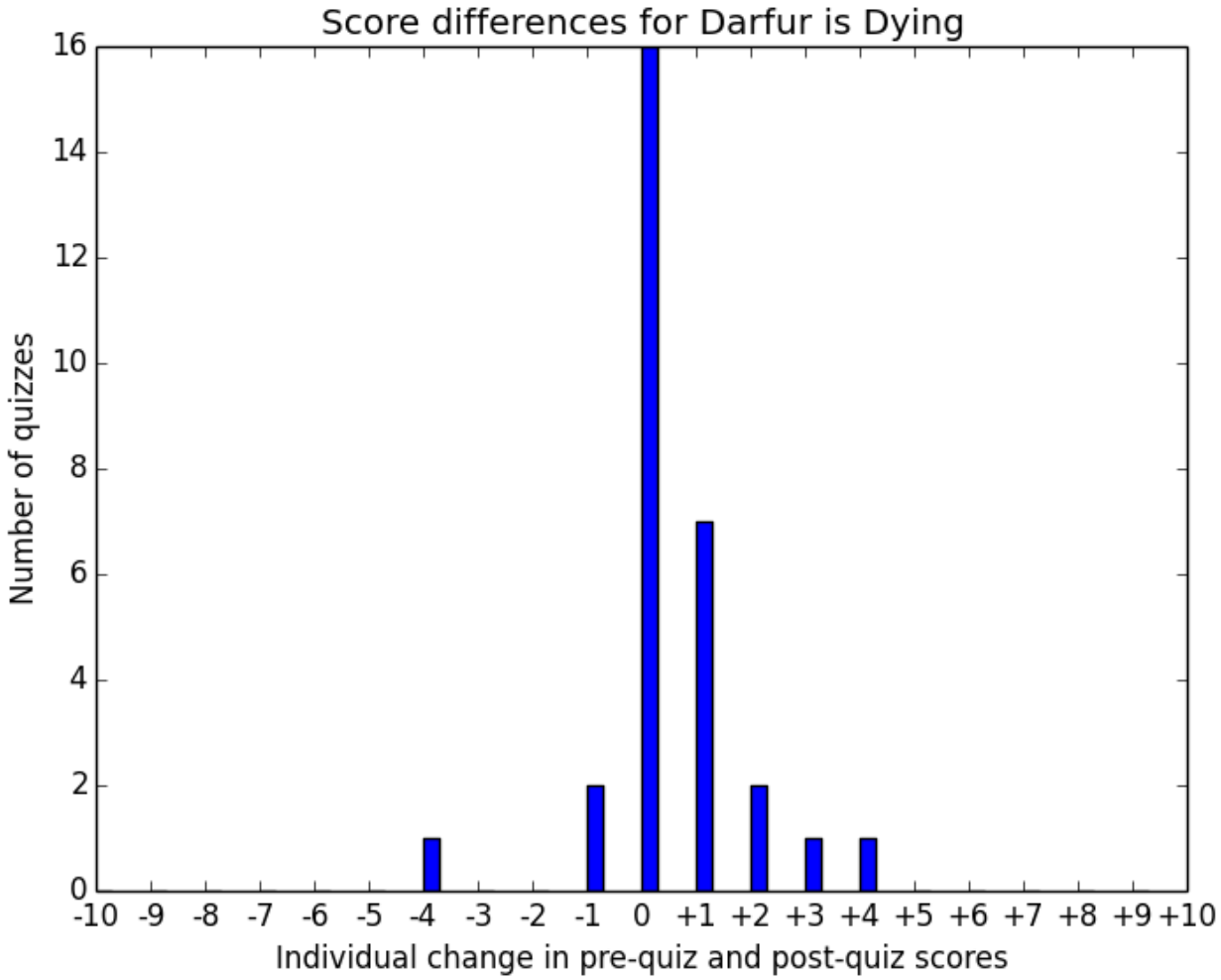

Figure 6.7: The score differences for Darfur is Dying.

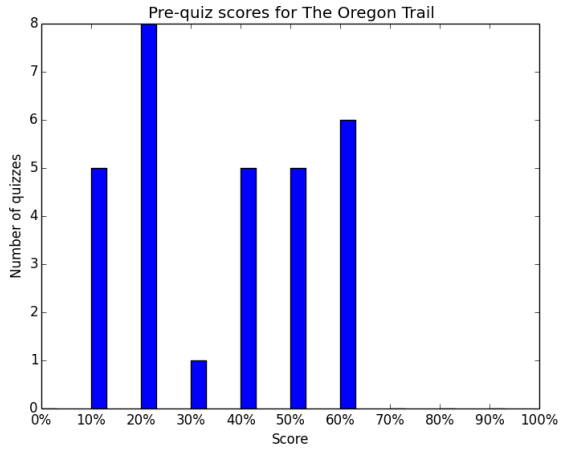

Figure 6.8: The pre-quiz scores for The Oregon Trail.

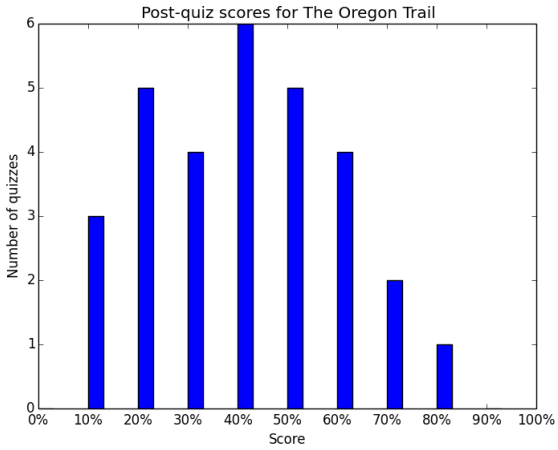

Figure 6.9: The post-quiz scores for The Oregon Trail.

6.2.3 The Oregon Trail Quiz Scores

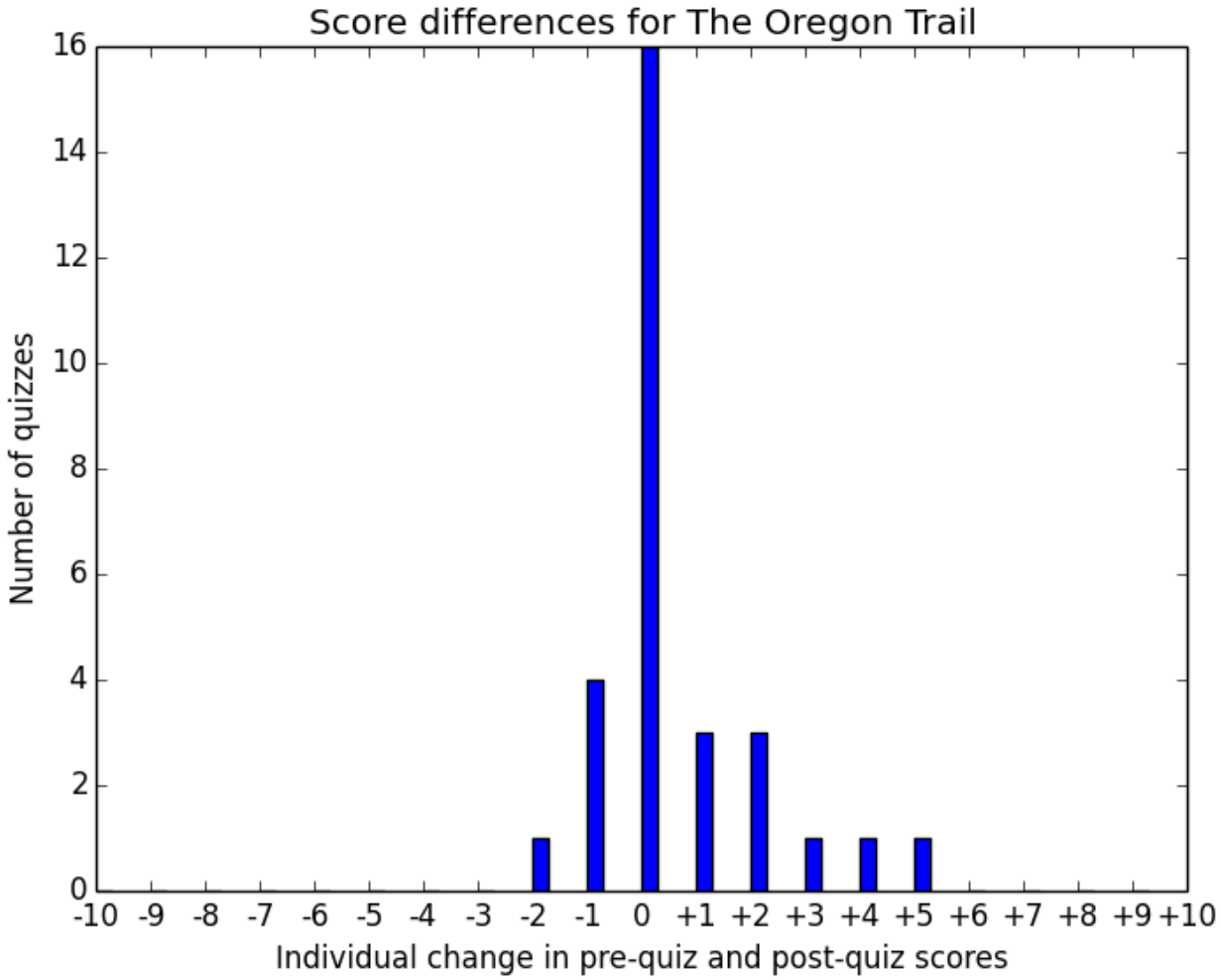

Figure 6.10: The score differences for The Oregon Trail.

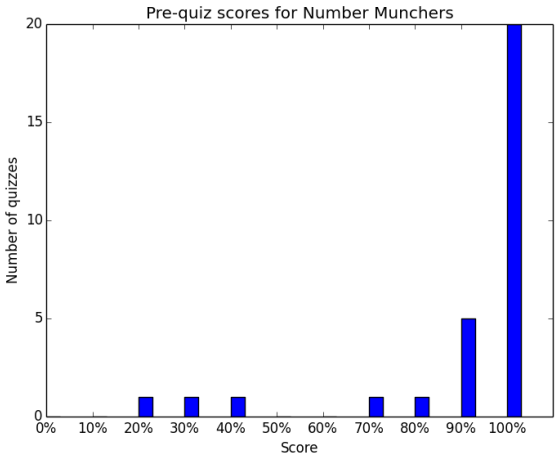

Figure 6.11: The pre-quiz scores for Number Munchers.

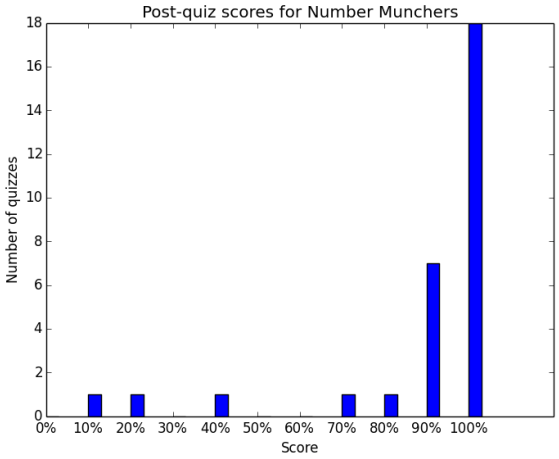

Figure 6.12: The post-quiz scores for Number Munchers.

6.2.4 Number Munchers Quiz Scores

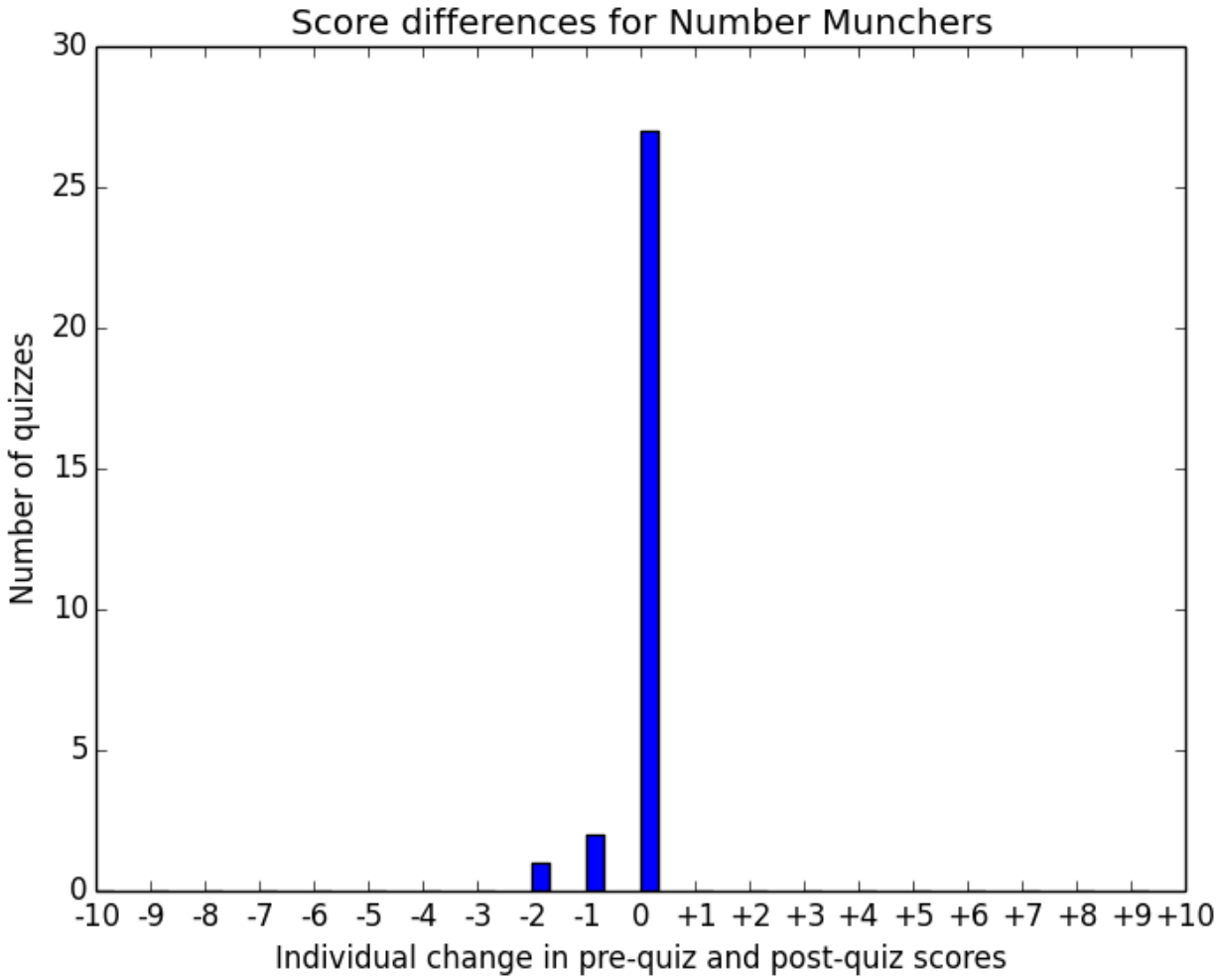

Figure 6.13: The score differences for Number Munchers.

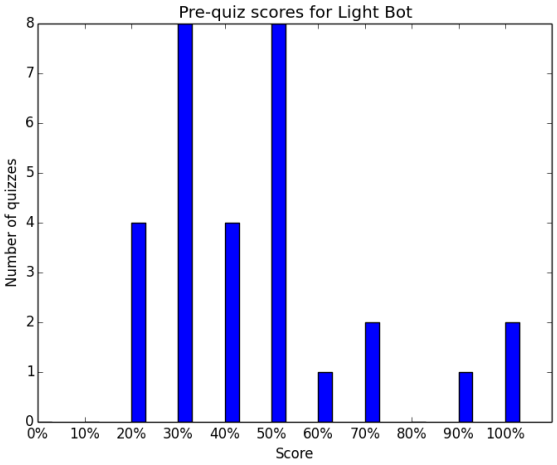

Figure 6.14: The pre-quiz scores for Light Bot.

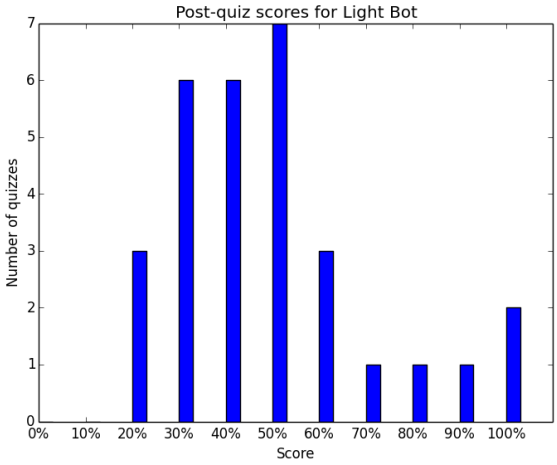

Figure 6.15: The post-quiz scores for Light Bot.

6.2.5 Light Bot Quiz Scores

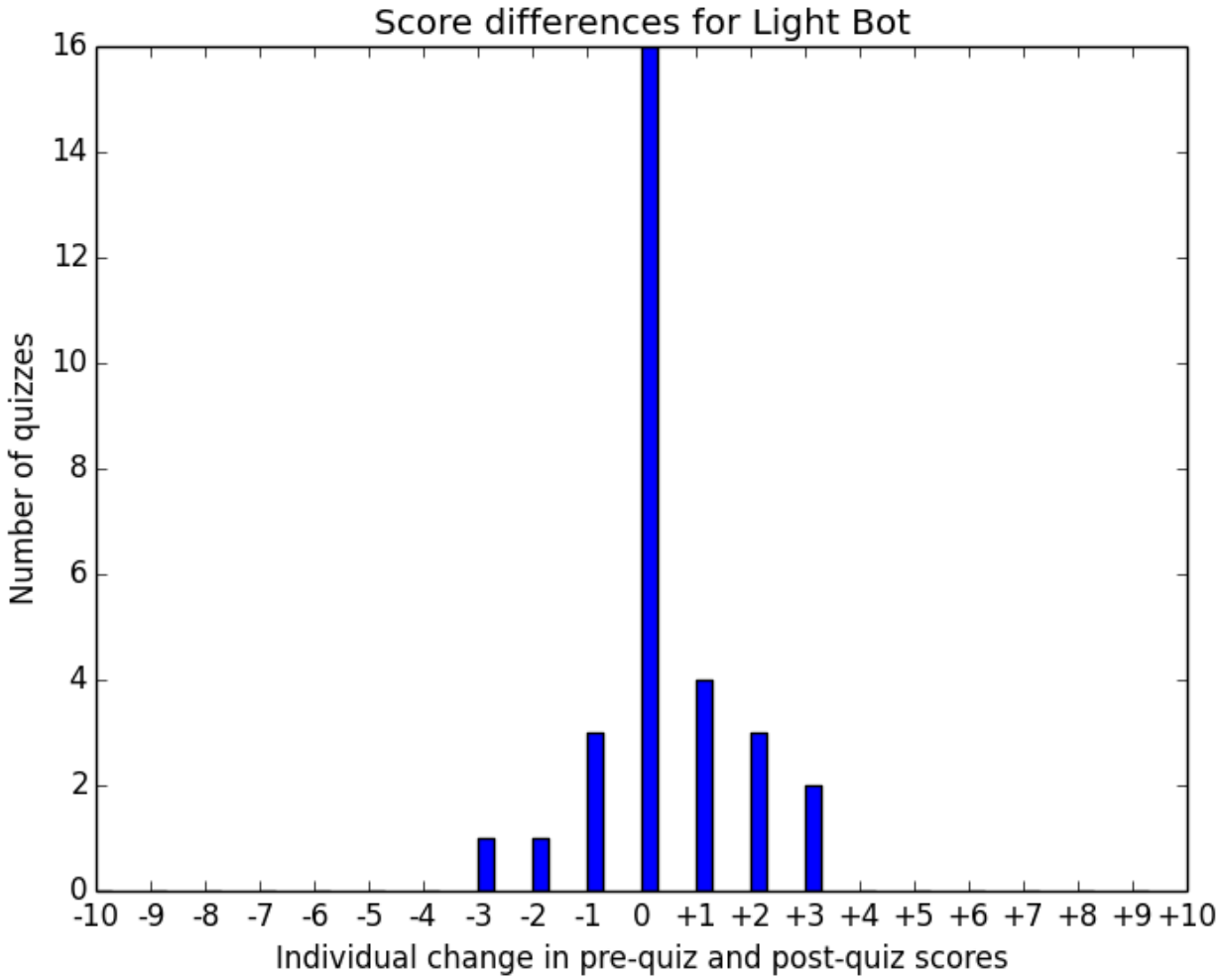

Figure 6.16: The score differences for Light Bot.

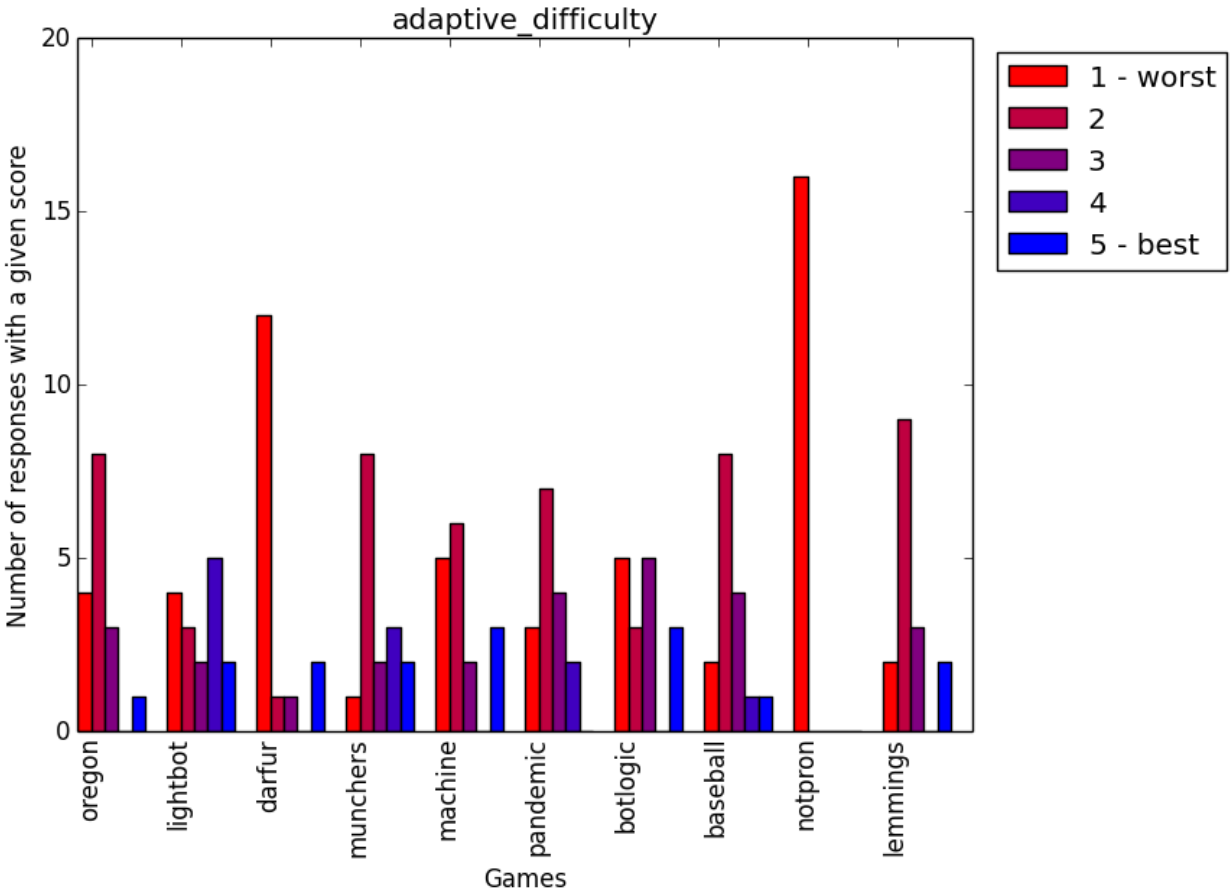

Figure 6.17: Adaptive Difficulty

6.3 Rubric Scores

This section contains the scoring results for each rubric item. For each rubric item, a bar graph is given for each game. The bar graph contains the scores that the game recieved for that rubric item.

An alternate visualization of this data, where each game has a bar graph for all of the rubric items, can be found in the next section.

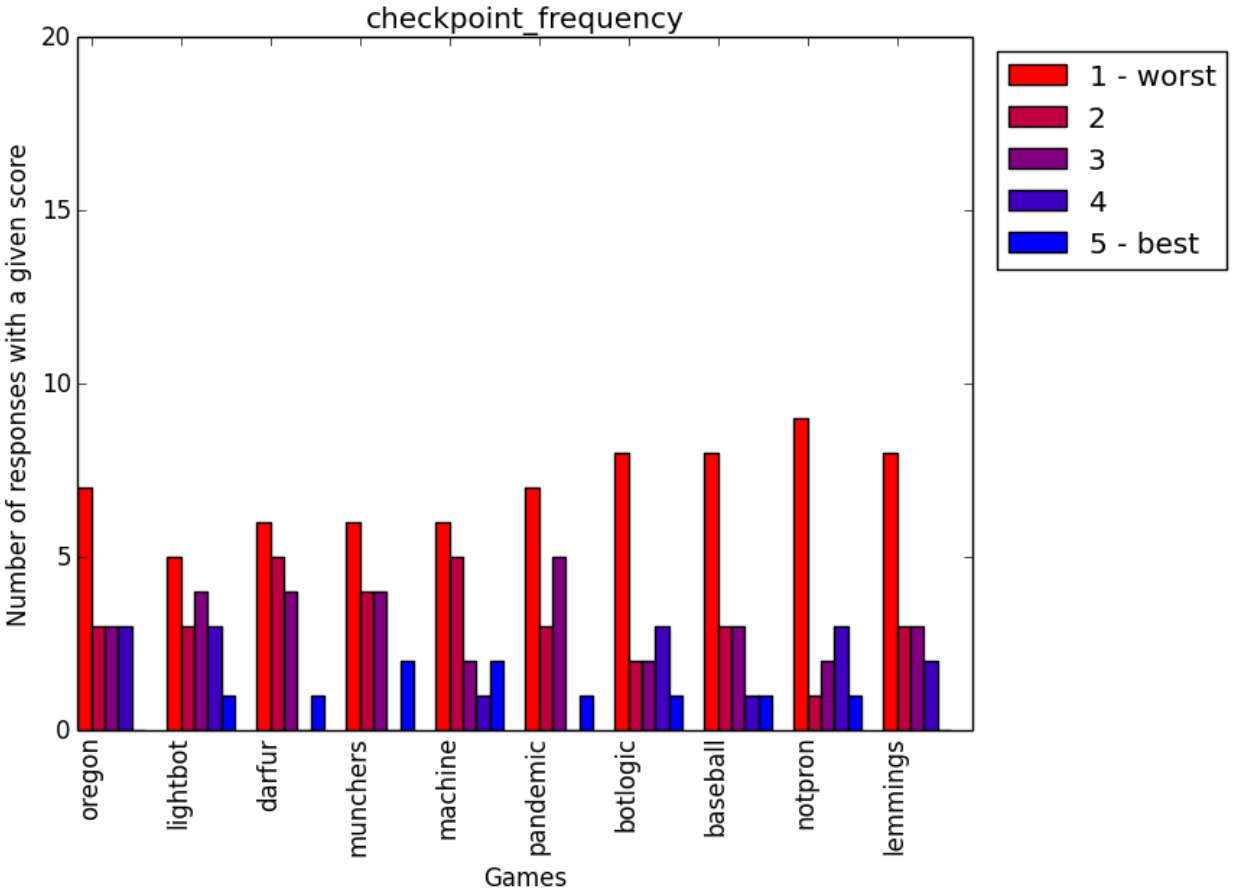

Figure 6.18: Checkpoint Frequency

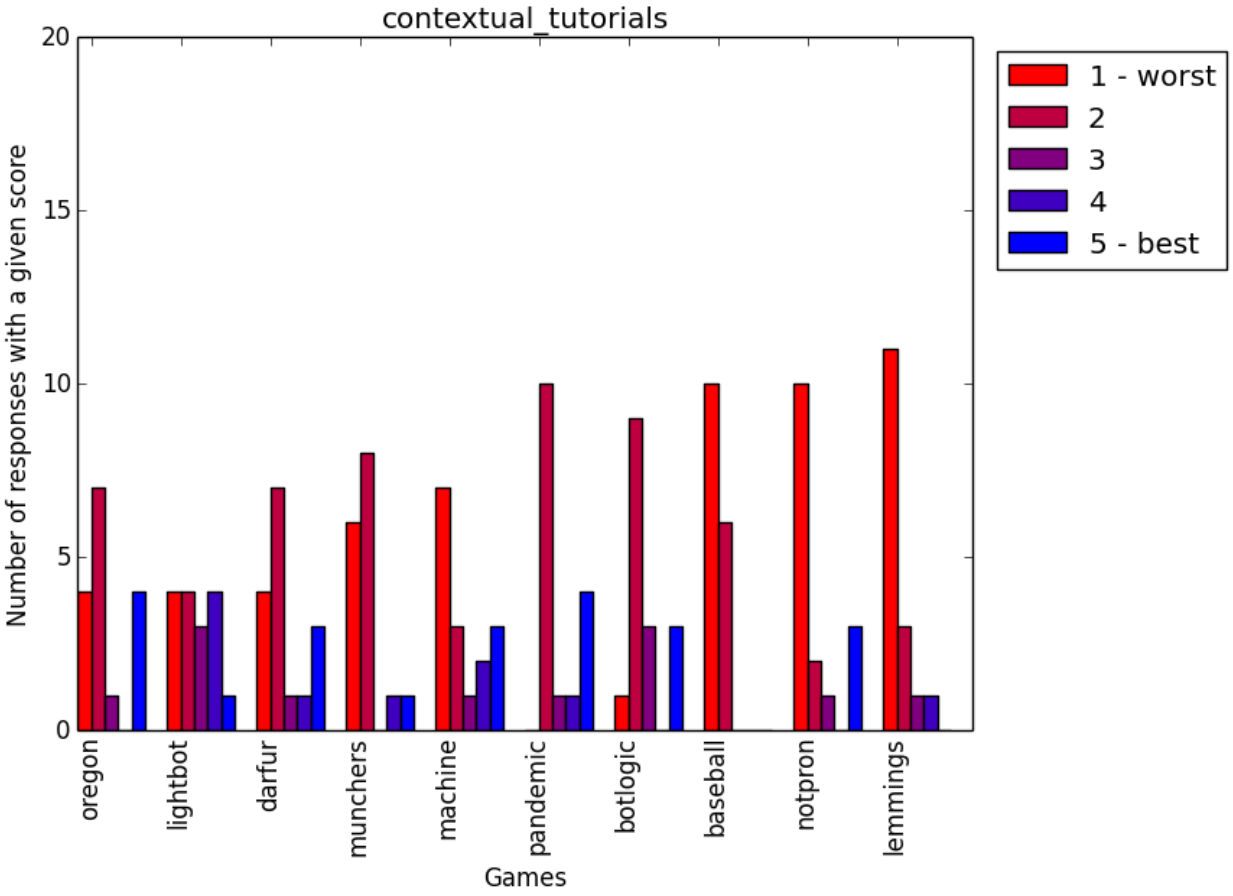

Figure 6.19: Contextual Tutorials

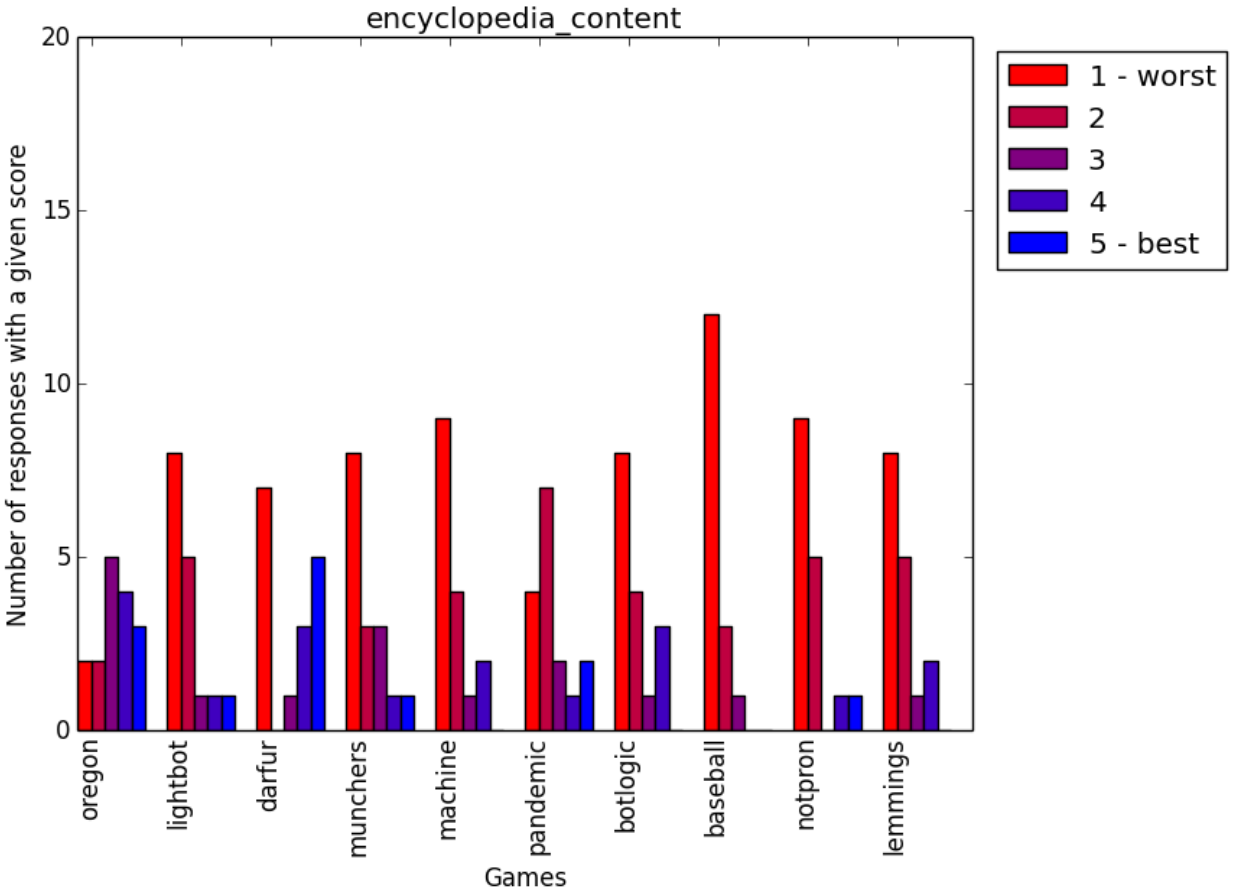

Figure 6.20: Encyclopedia Content

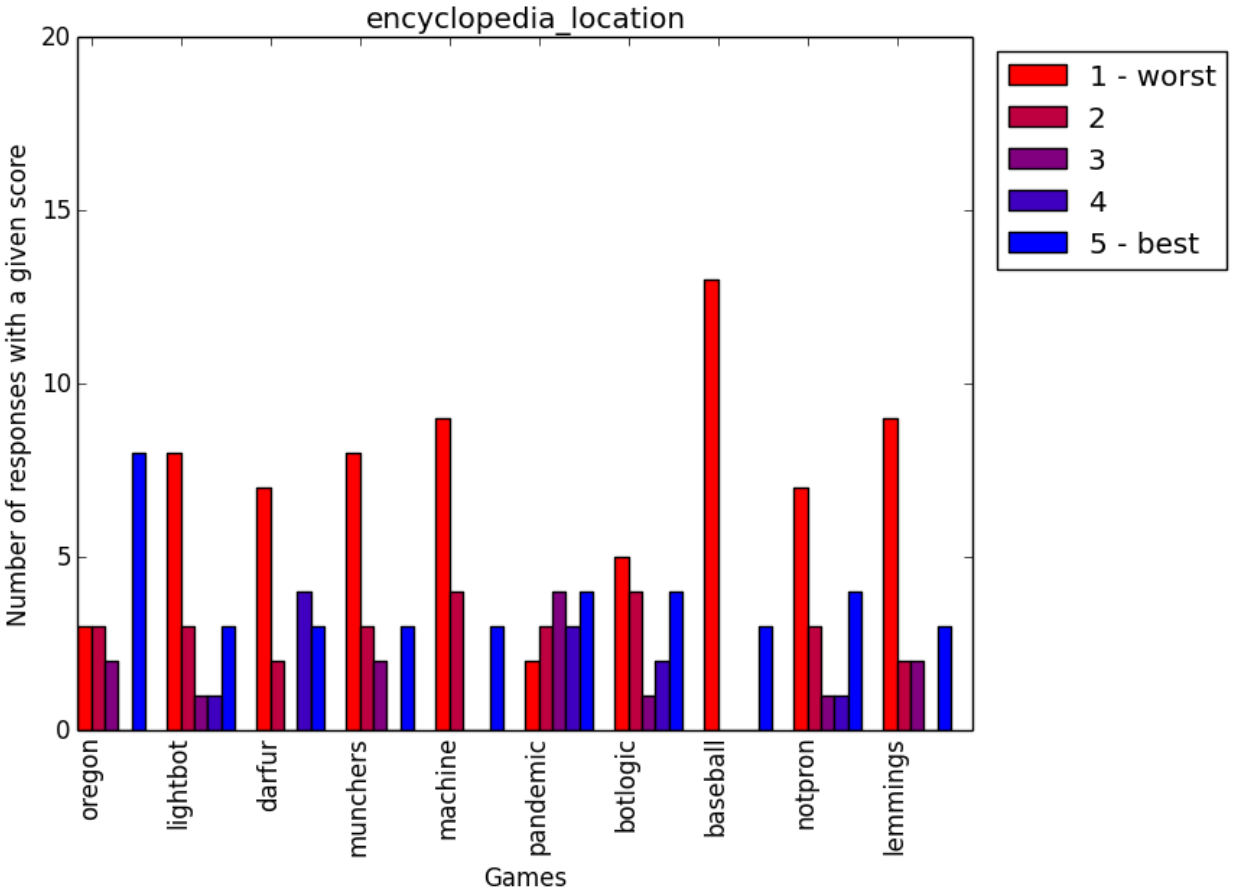

Figure 6.21: Encyclopedia Location

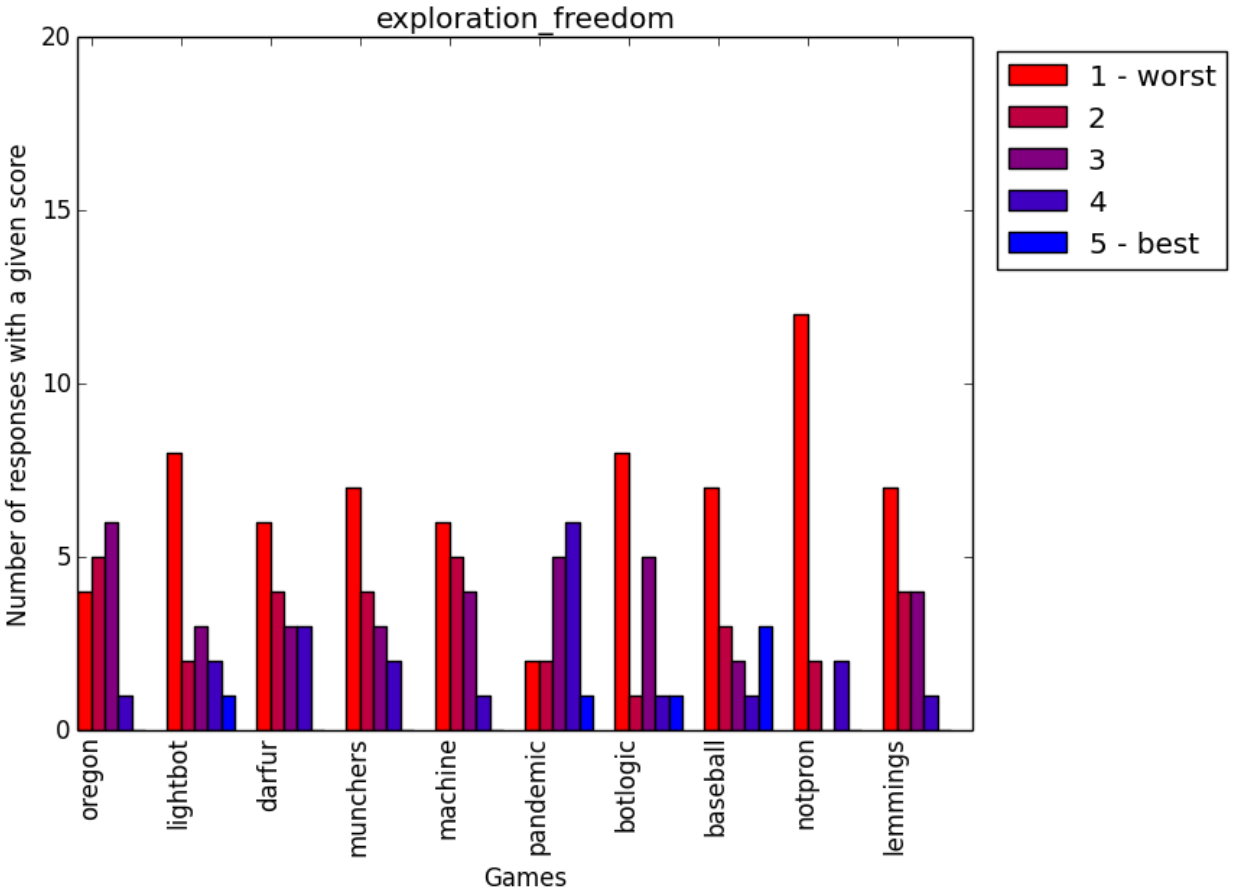

Figure 6.22: Freedom of exploration

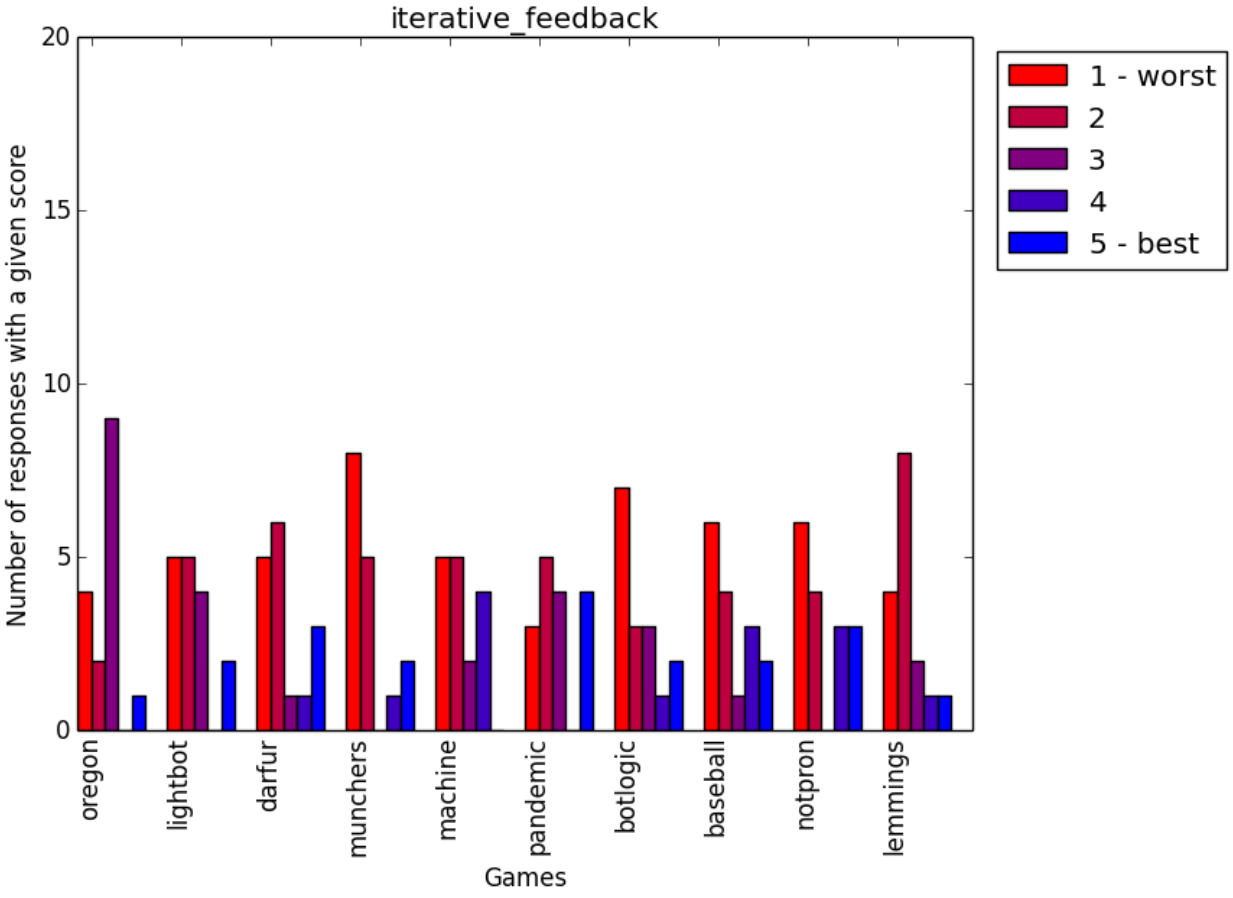

Figure 6.23: Iterative feedback

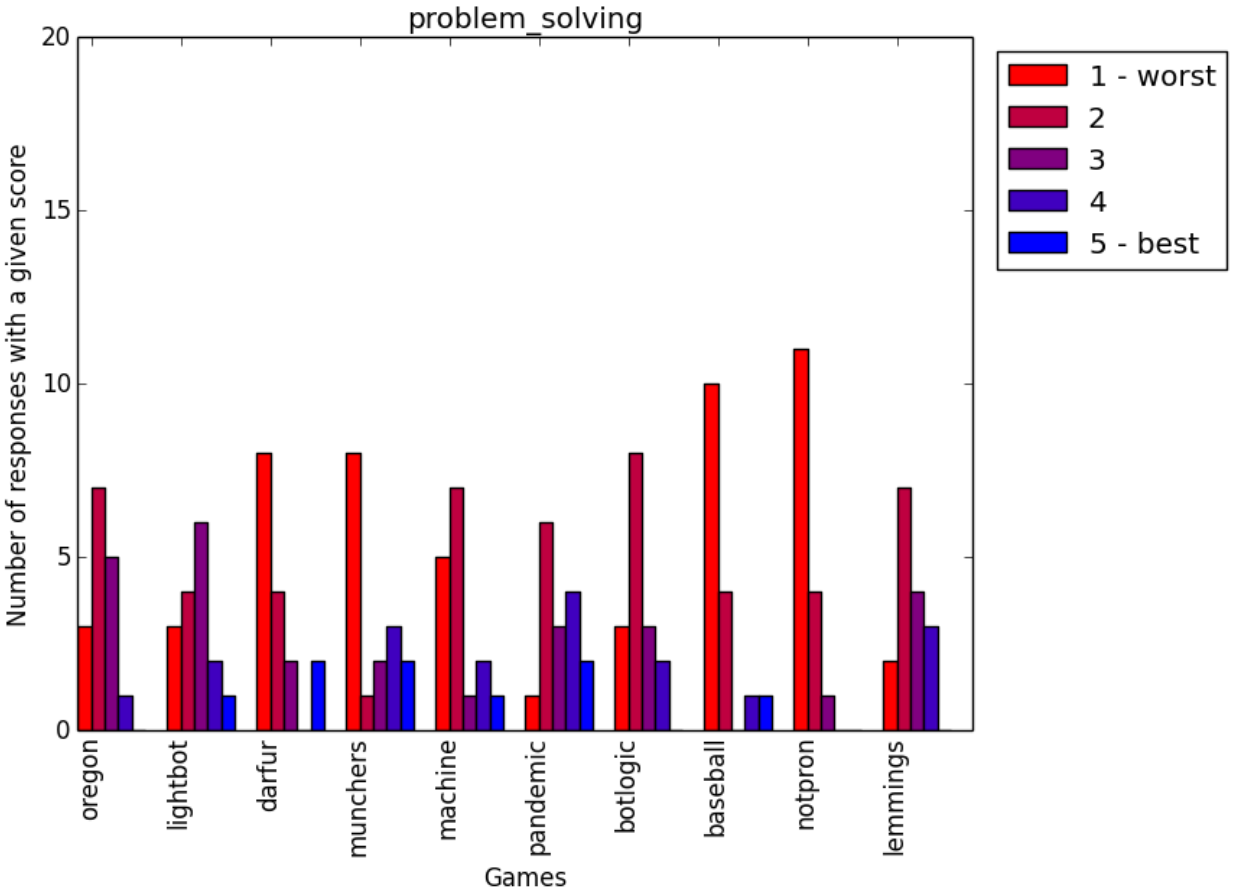

Figure 6.24: Unorthodox problem-solving

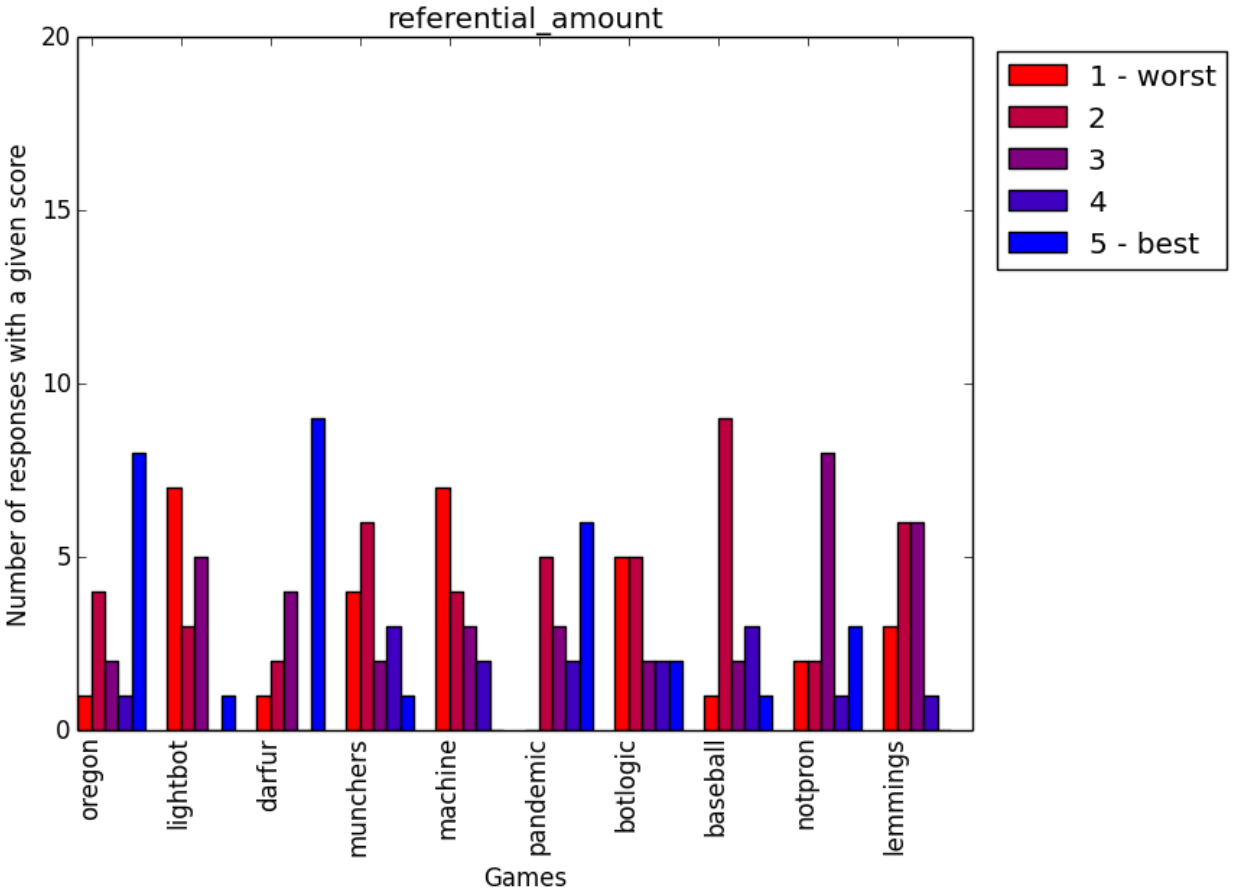

Figure 6.25: Amount of referential material

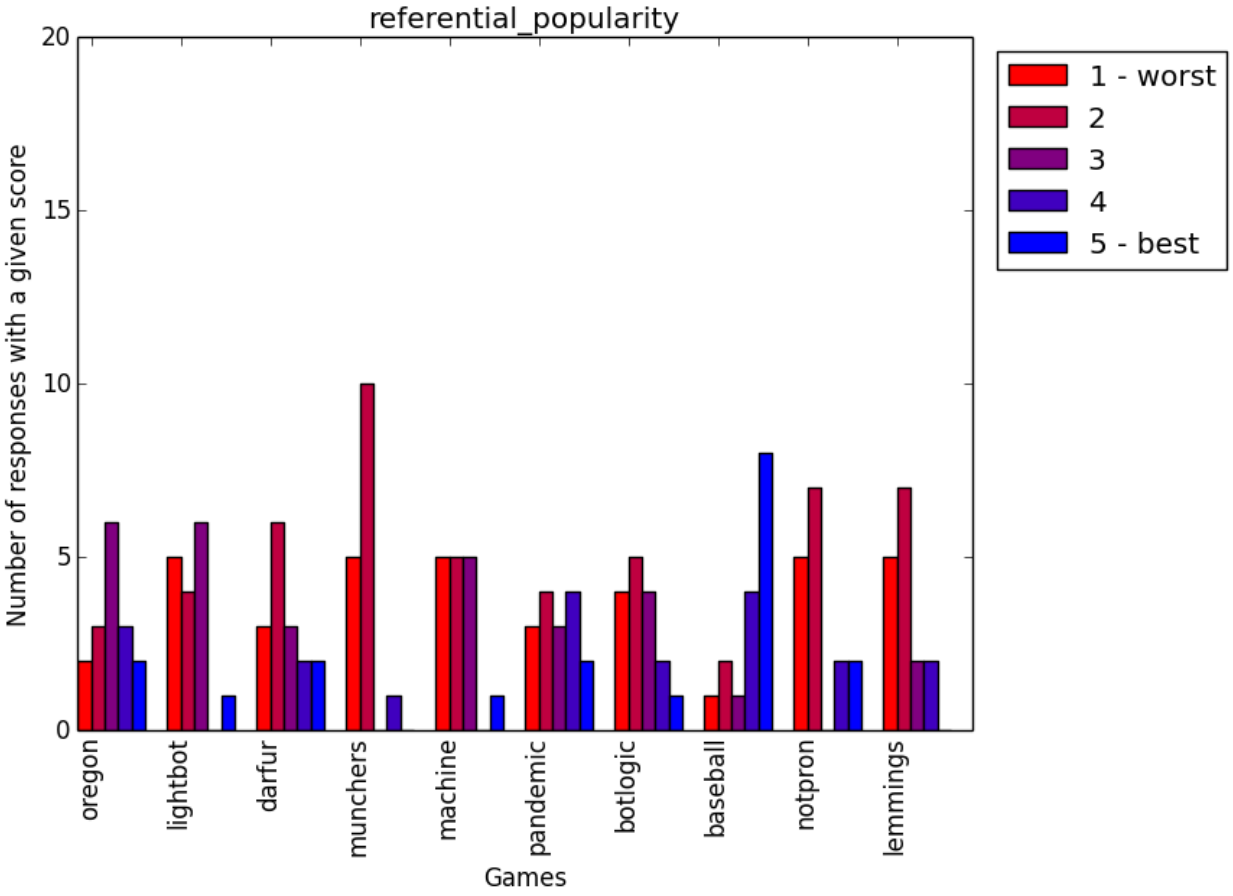

Figure 6.26: Popularity of referential material

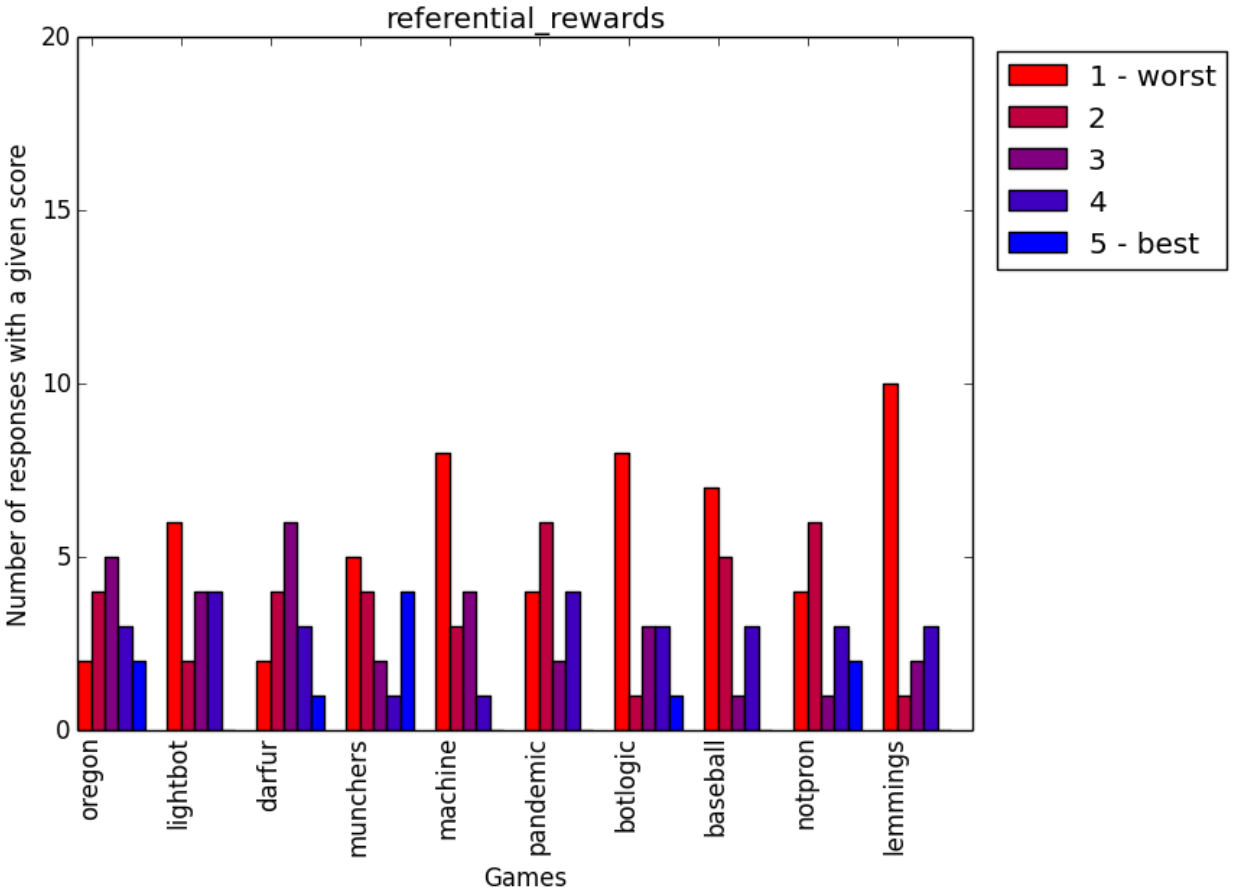

Figure 6.27: Rewards for knowing referential material

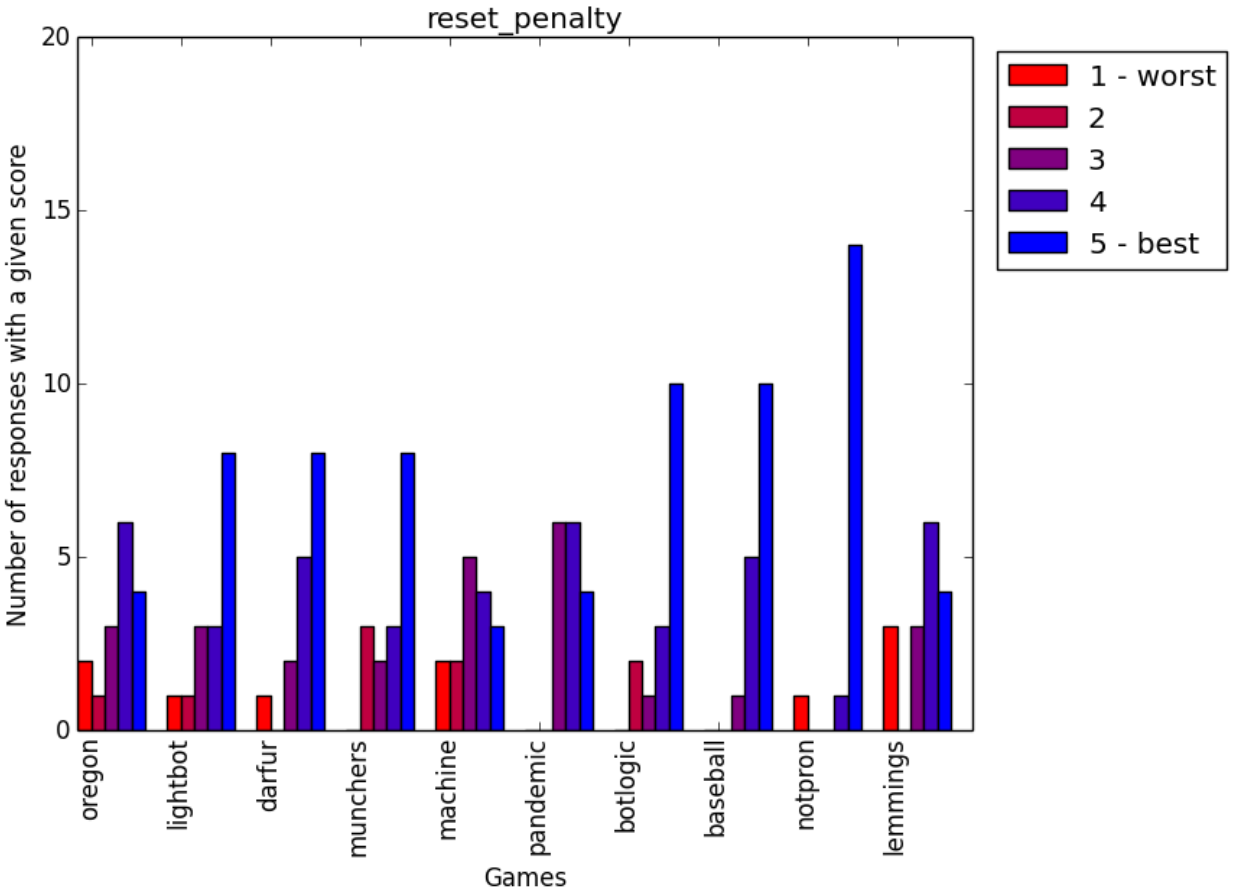

Figure 6.28: Reset time penalty for failure

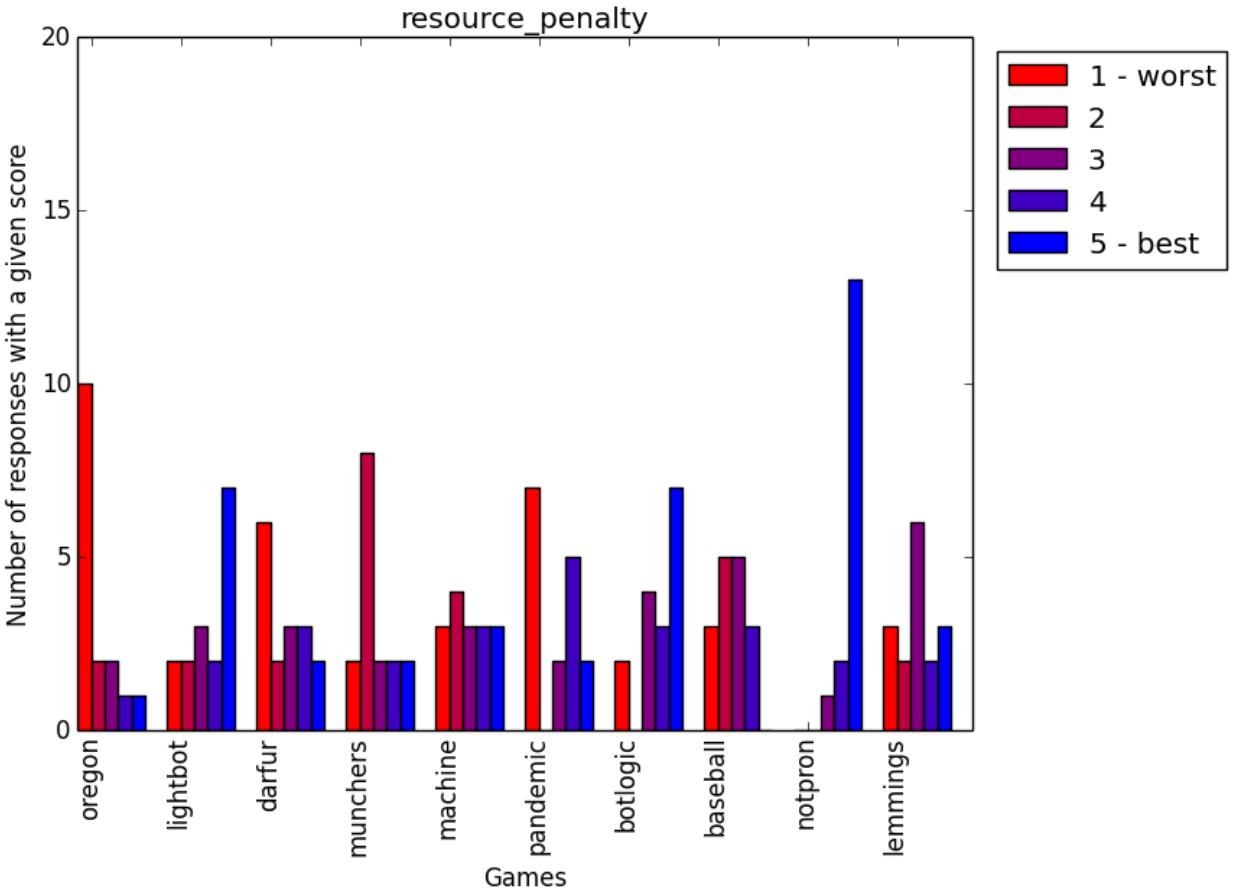

Figure 6.29: Game resource penalty for failure

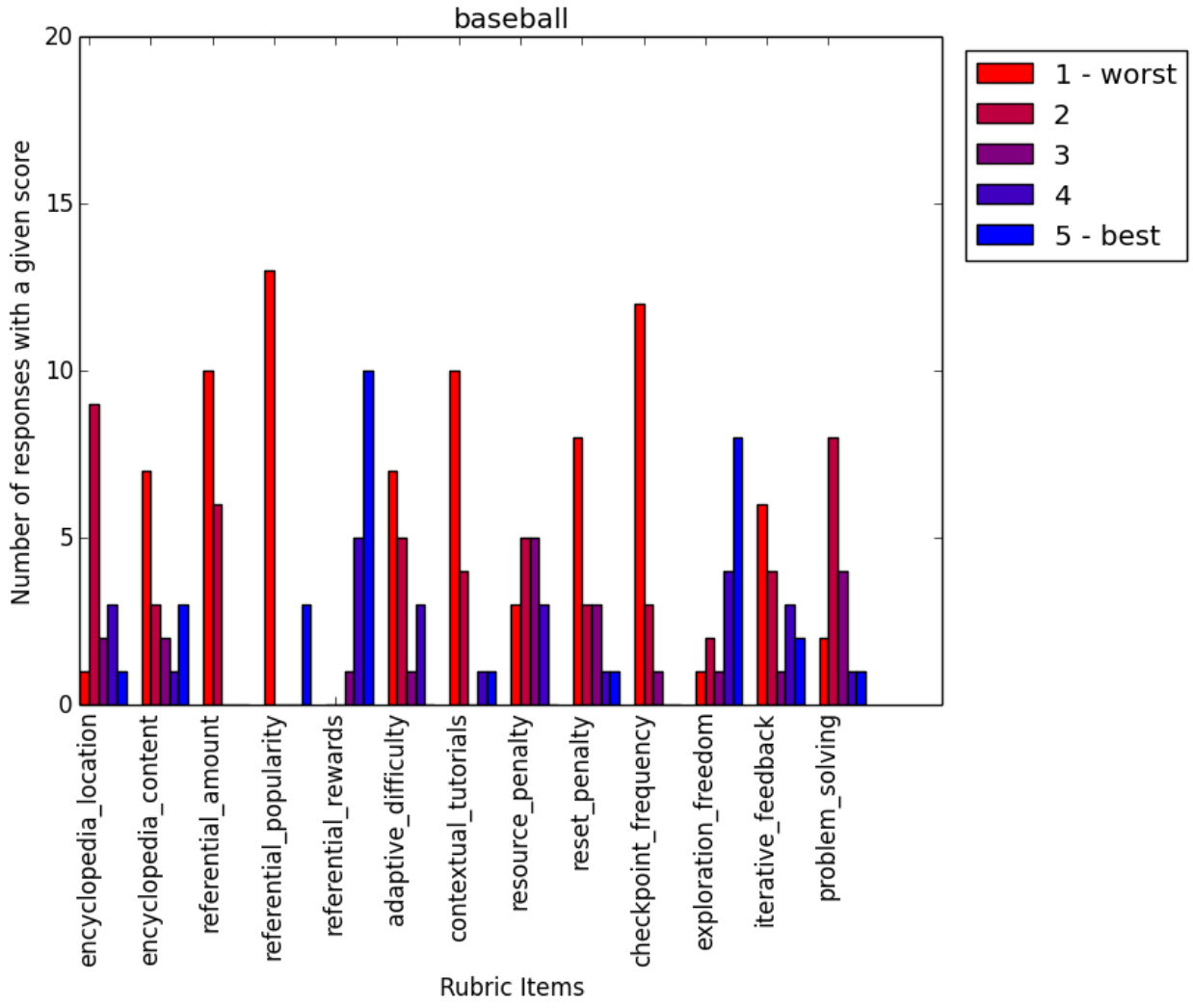

Figure 6.30: Math Baseball

6.4 Game Scores

This section contains the scoring results for each game. For each game, a bar graph is given for each rubric item. The bar graph contains the scores that the rubric item recieved for that game.

An alternate visualization of this data, where each rubric item has a bar graph for all of the game, can be found in the previous section.

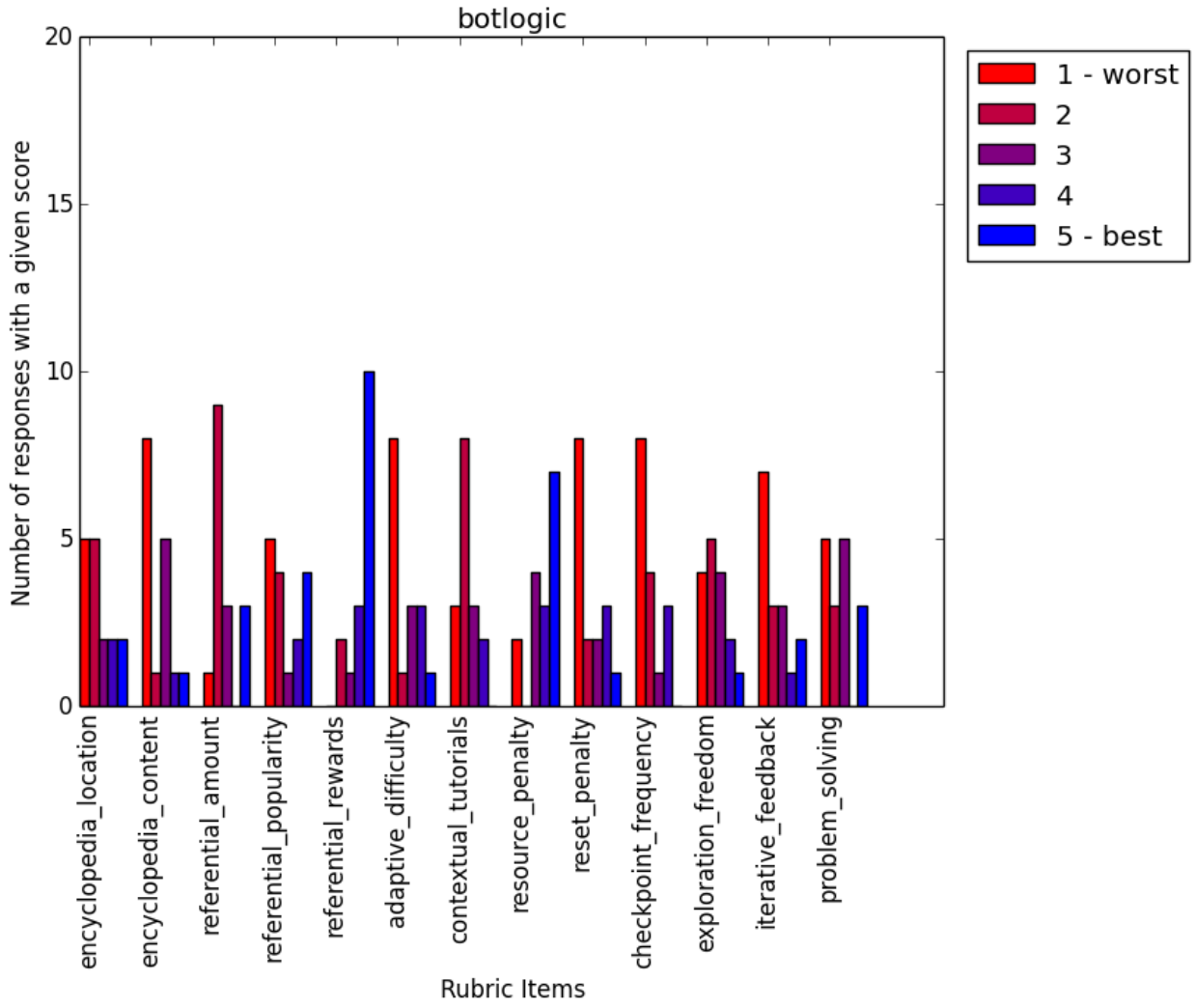

Figure 6.31: Botlogic

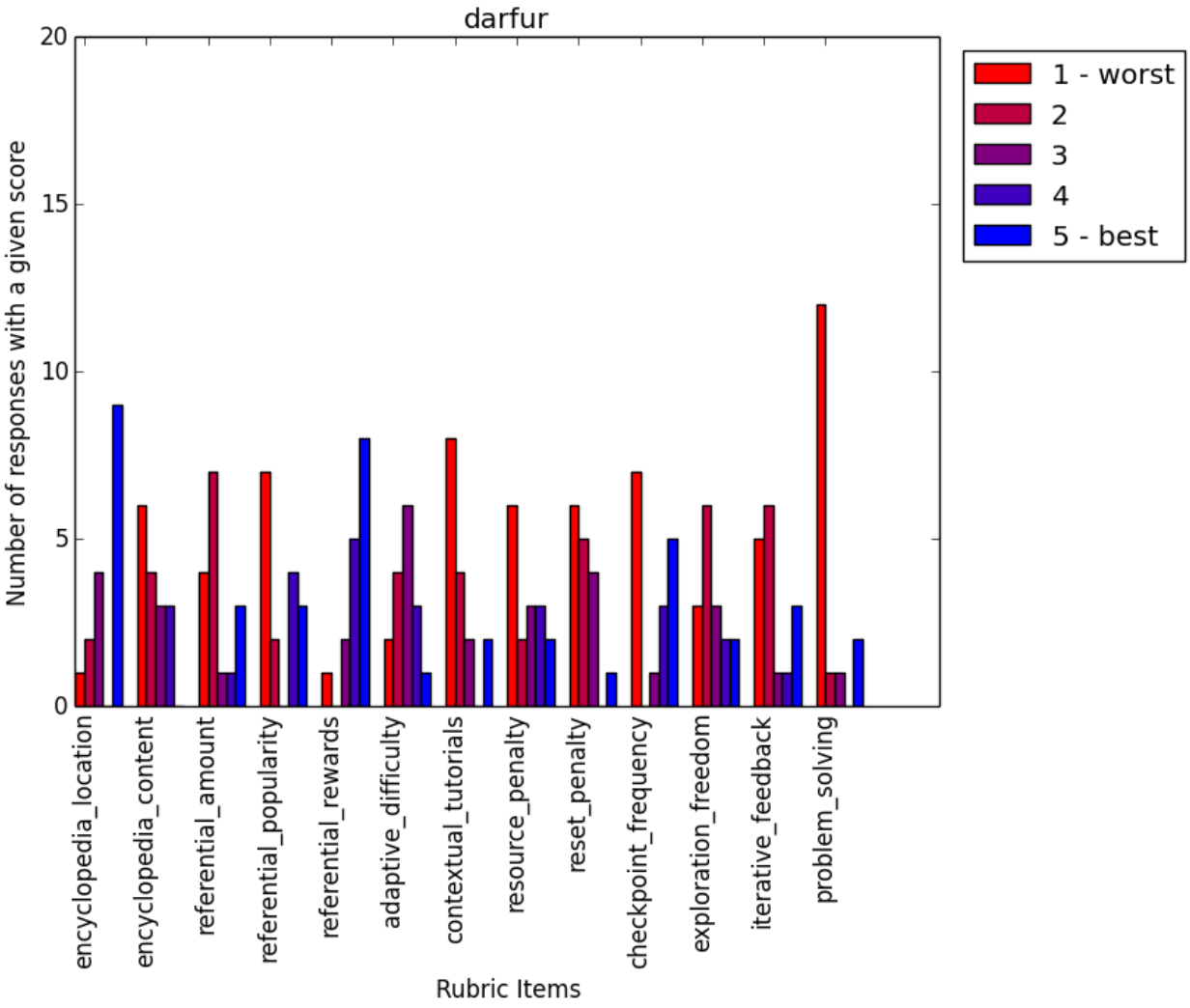

Figure 6.32: Darfur is Dying

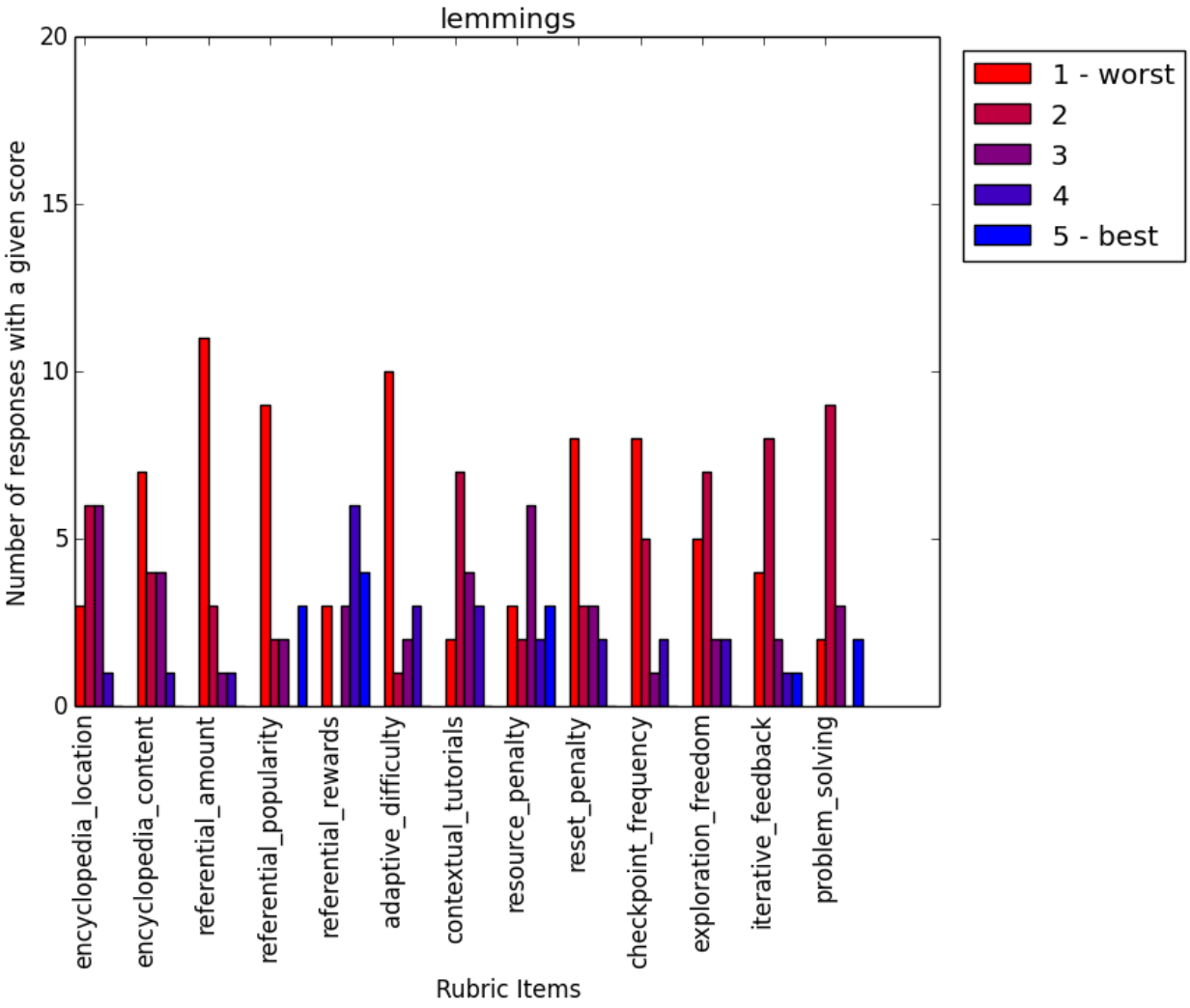

Figure 6.33: Lemmings

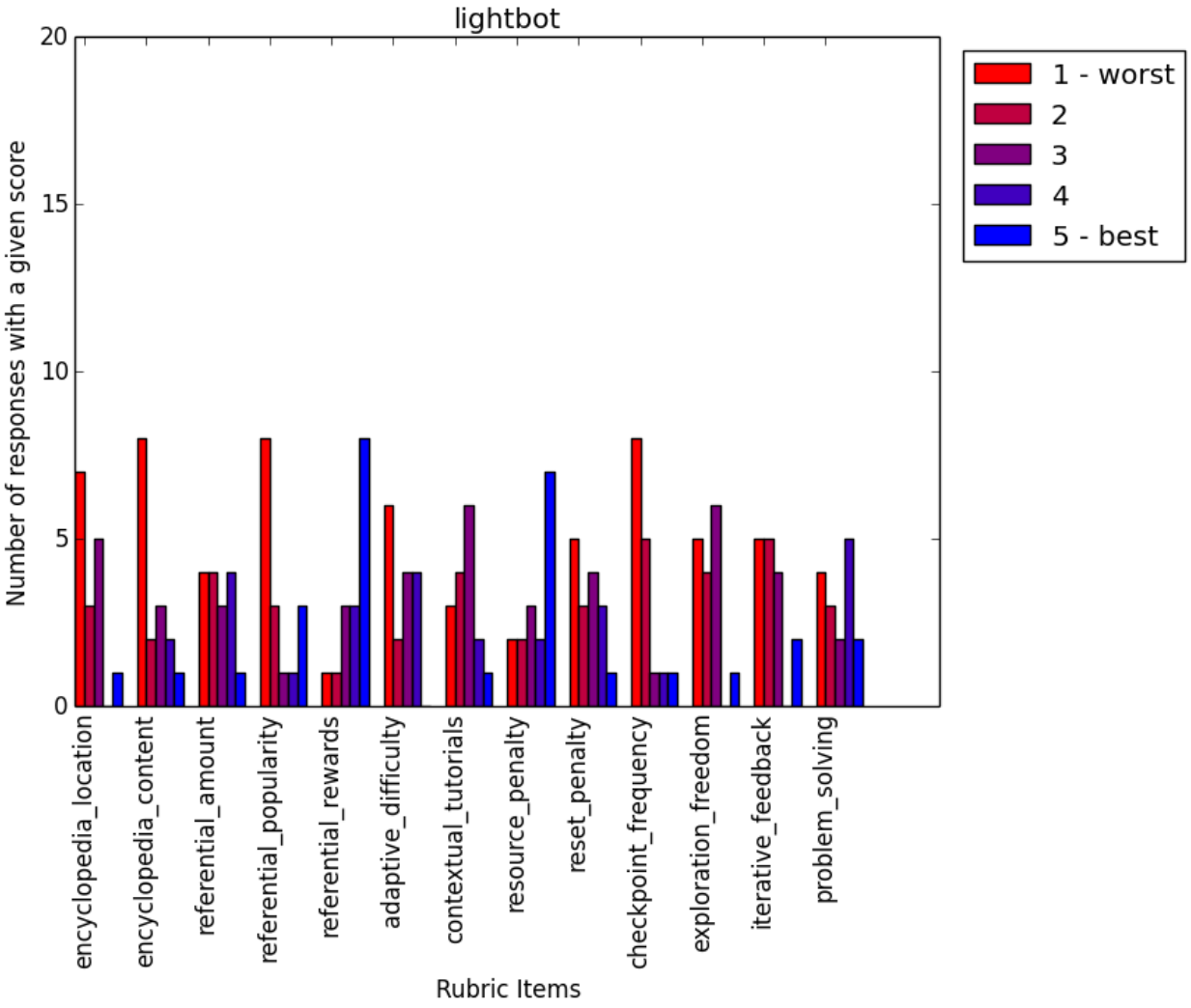

Figure 6.34: Light Bot

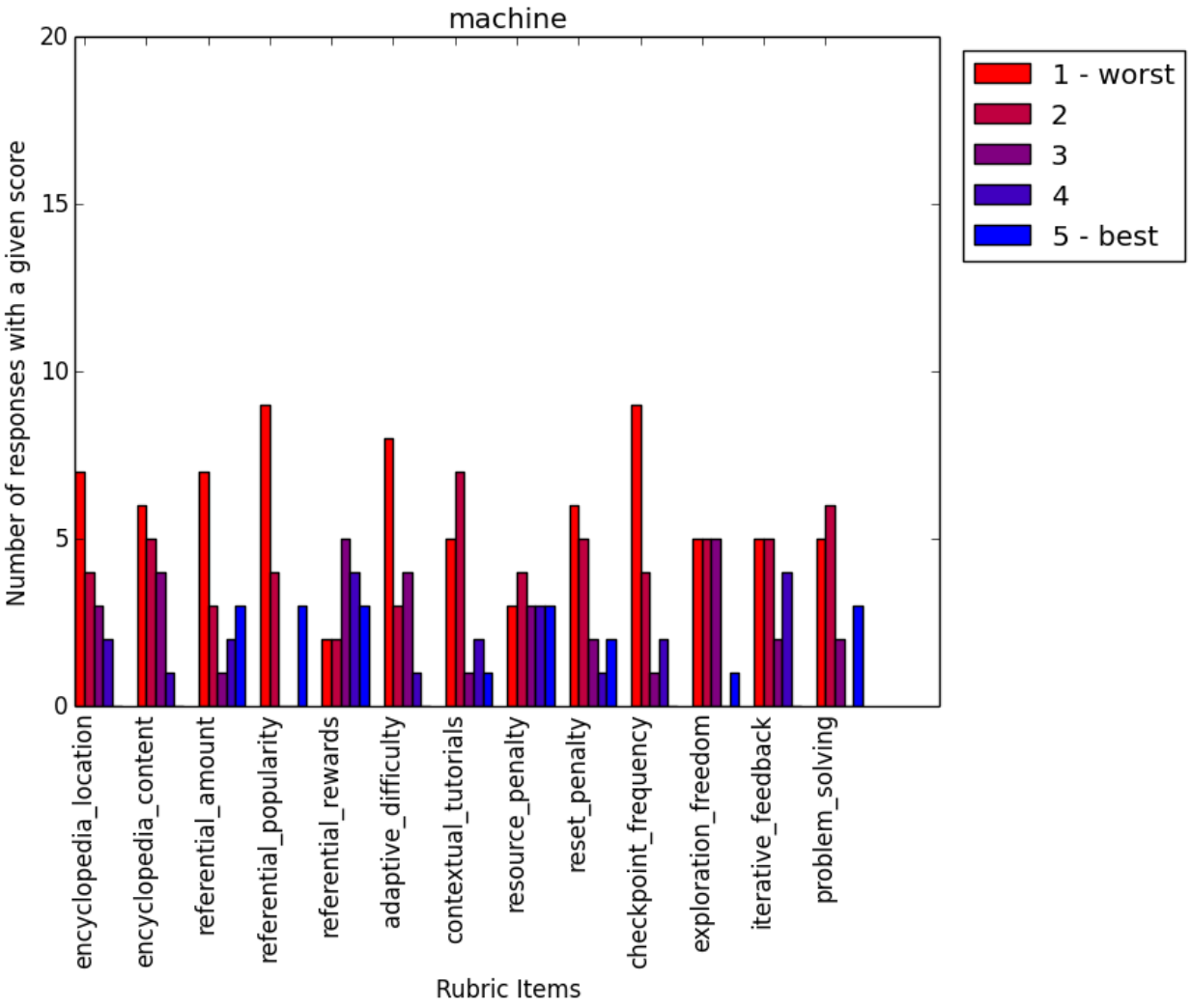

Figure 6.35: The Incredible Machine

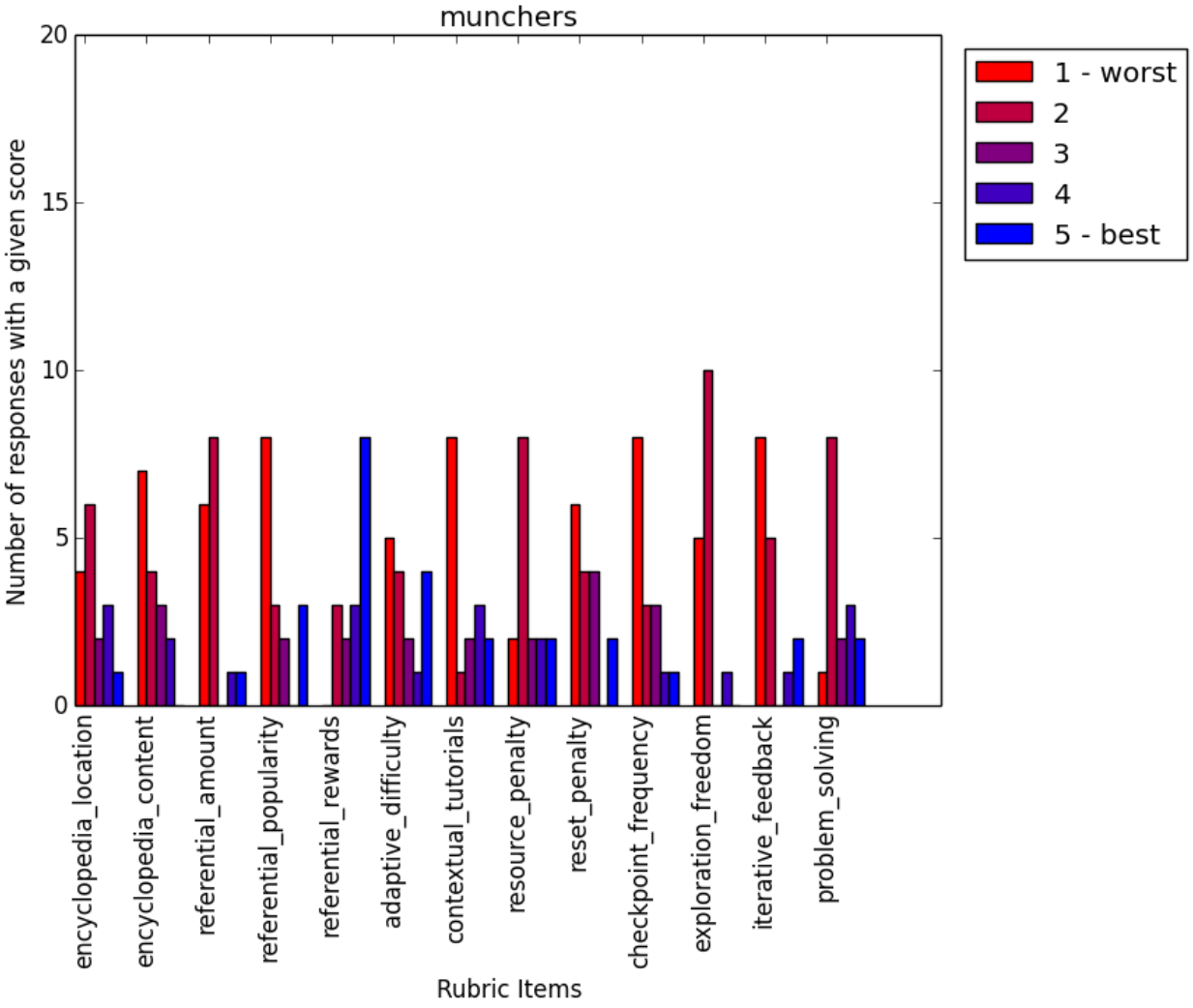

Figure 6.36: Number Munchers

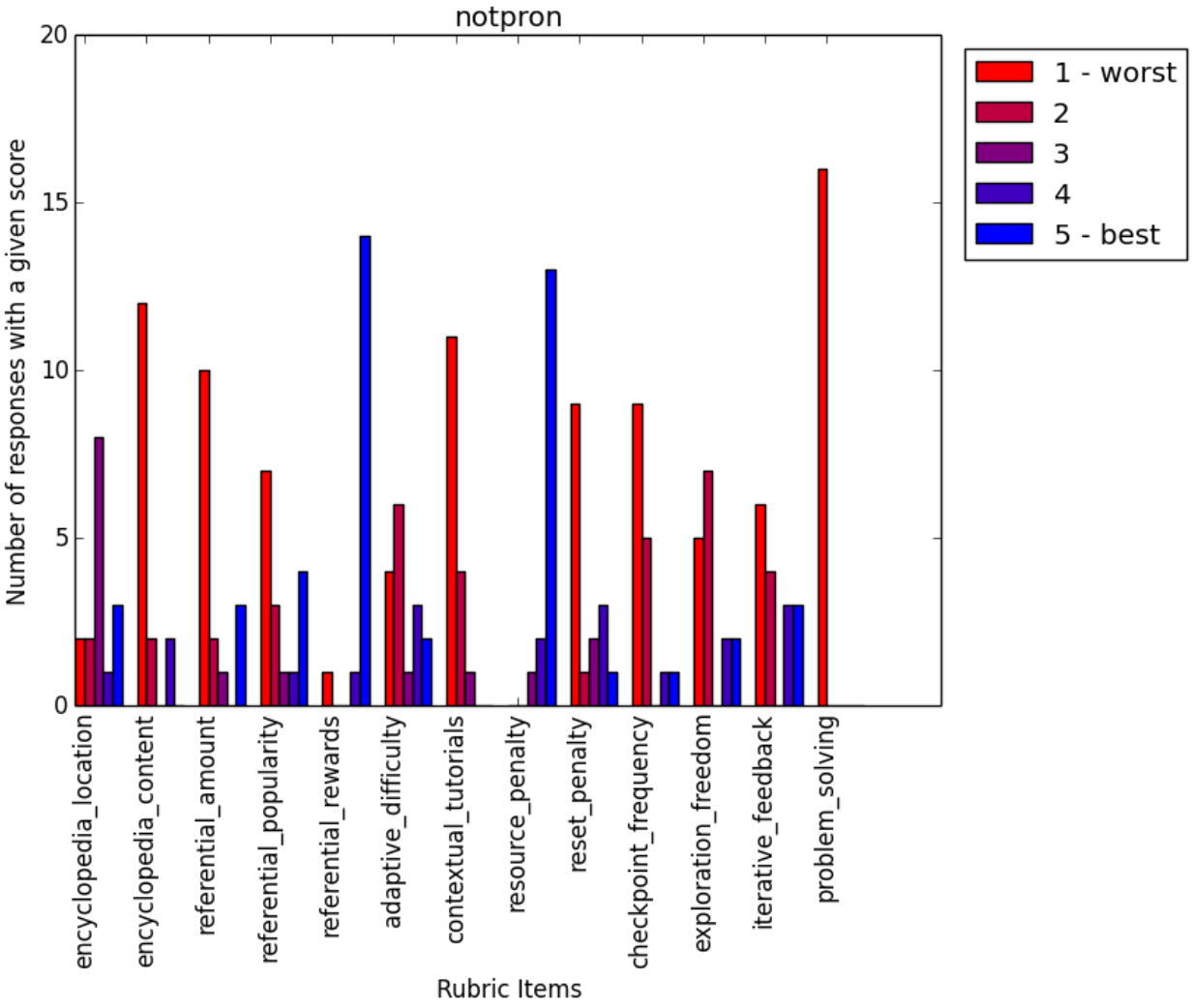

Figure 6.37: Notpron

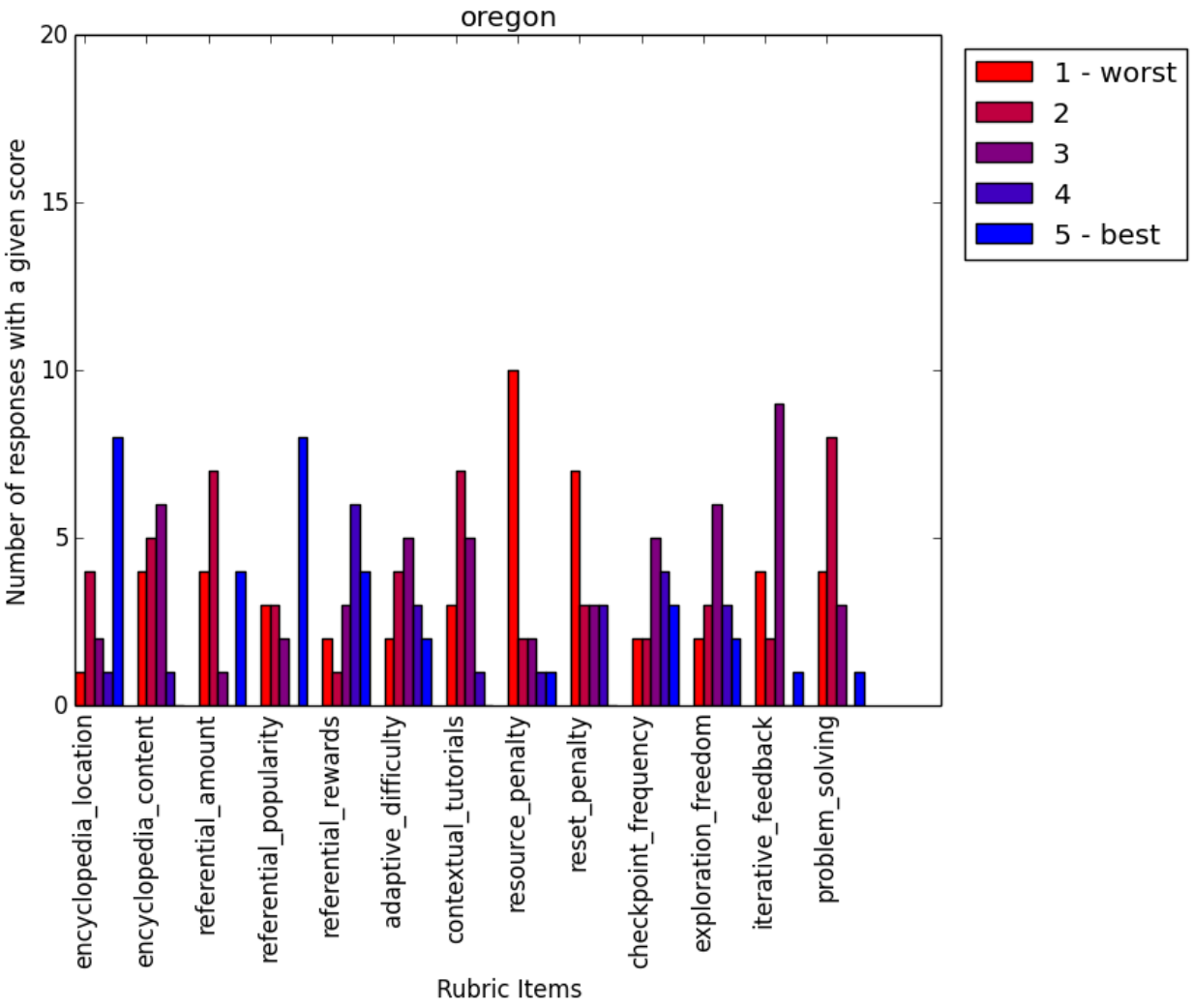

Figure 6.38: The Oregon Trail

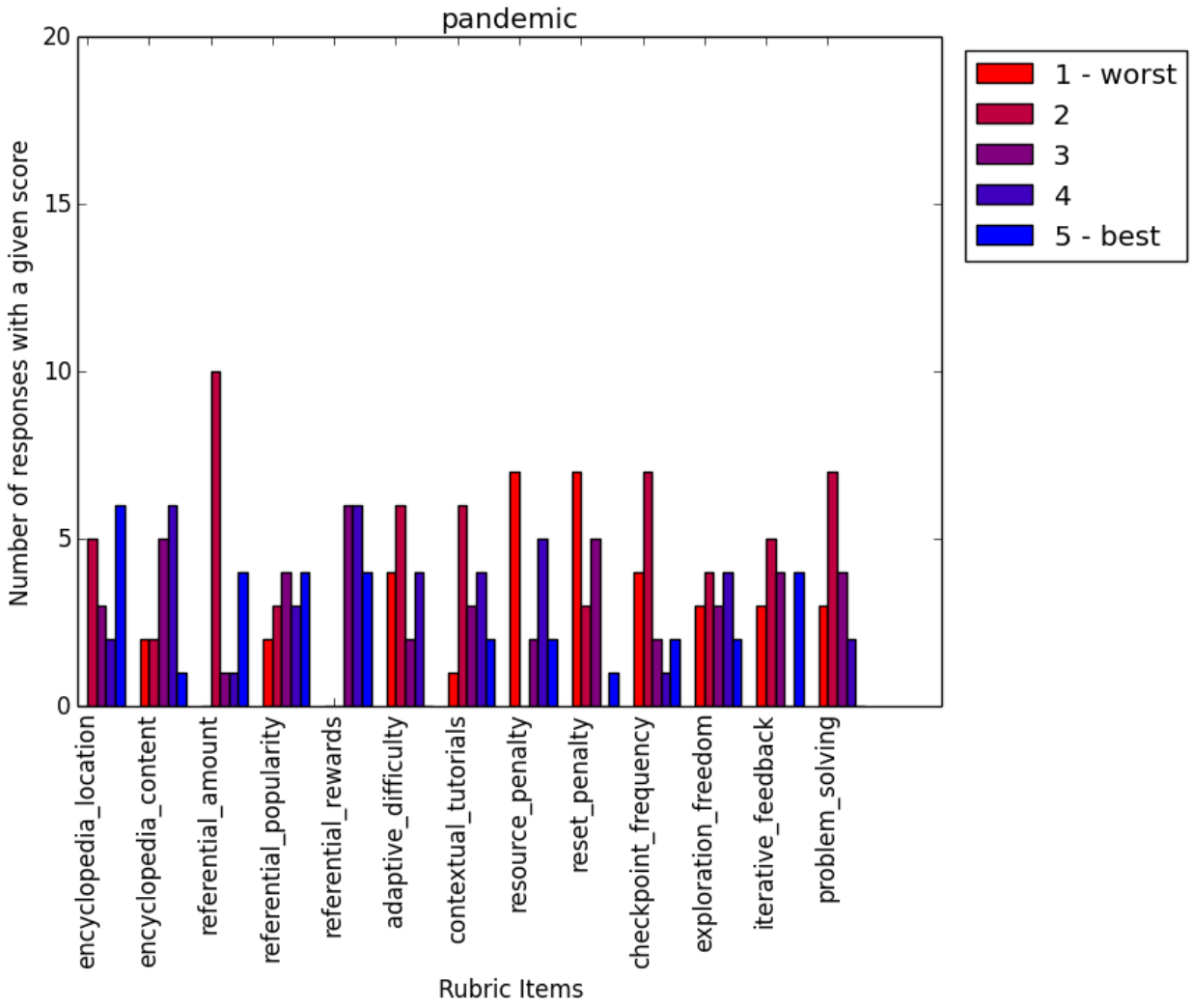

Figure 6.39: Pandemic 2

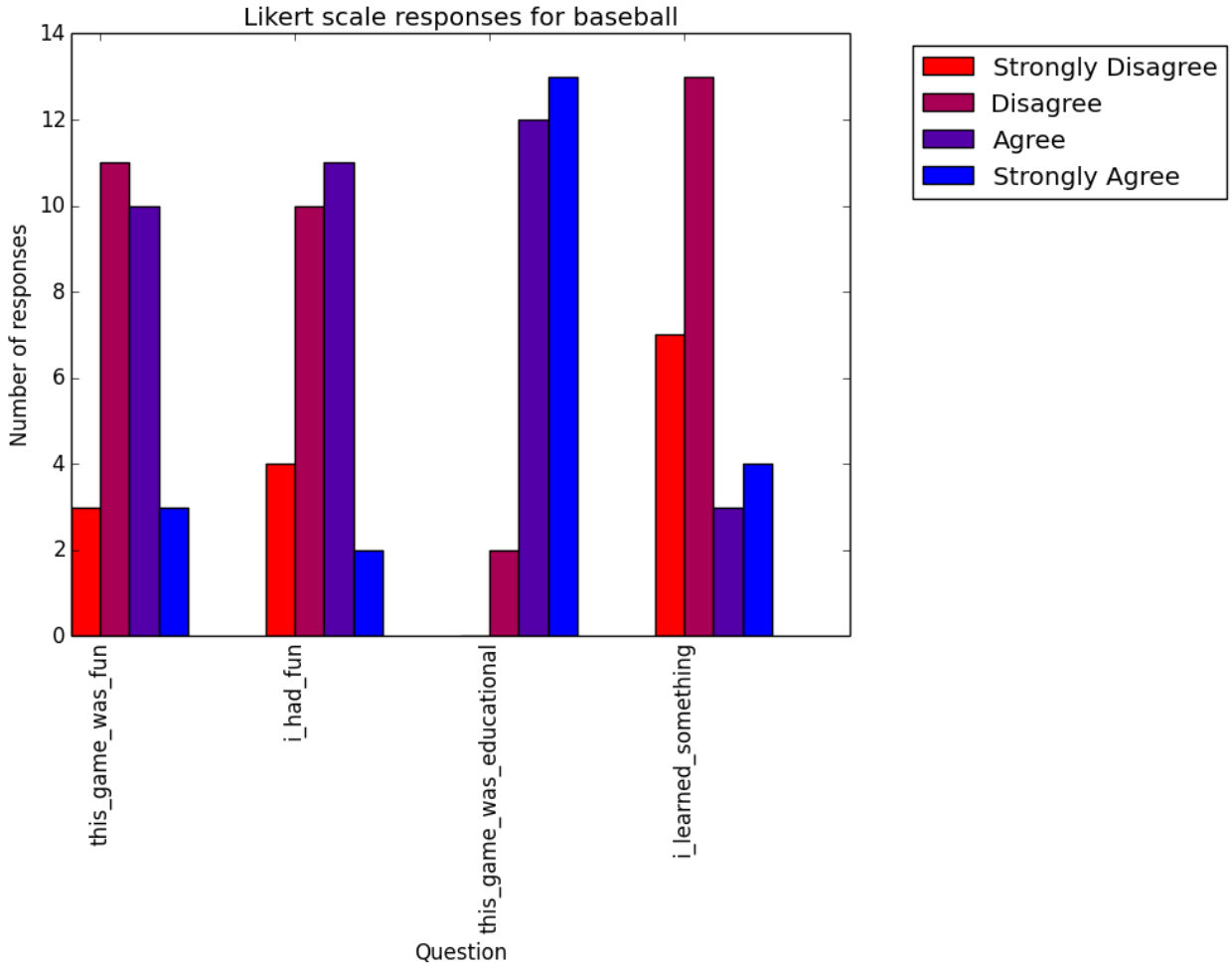

Figure 6.40: Math Baseball

6.5 Opinions on the fun and educational value of each game

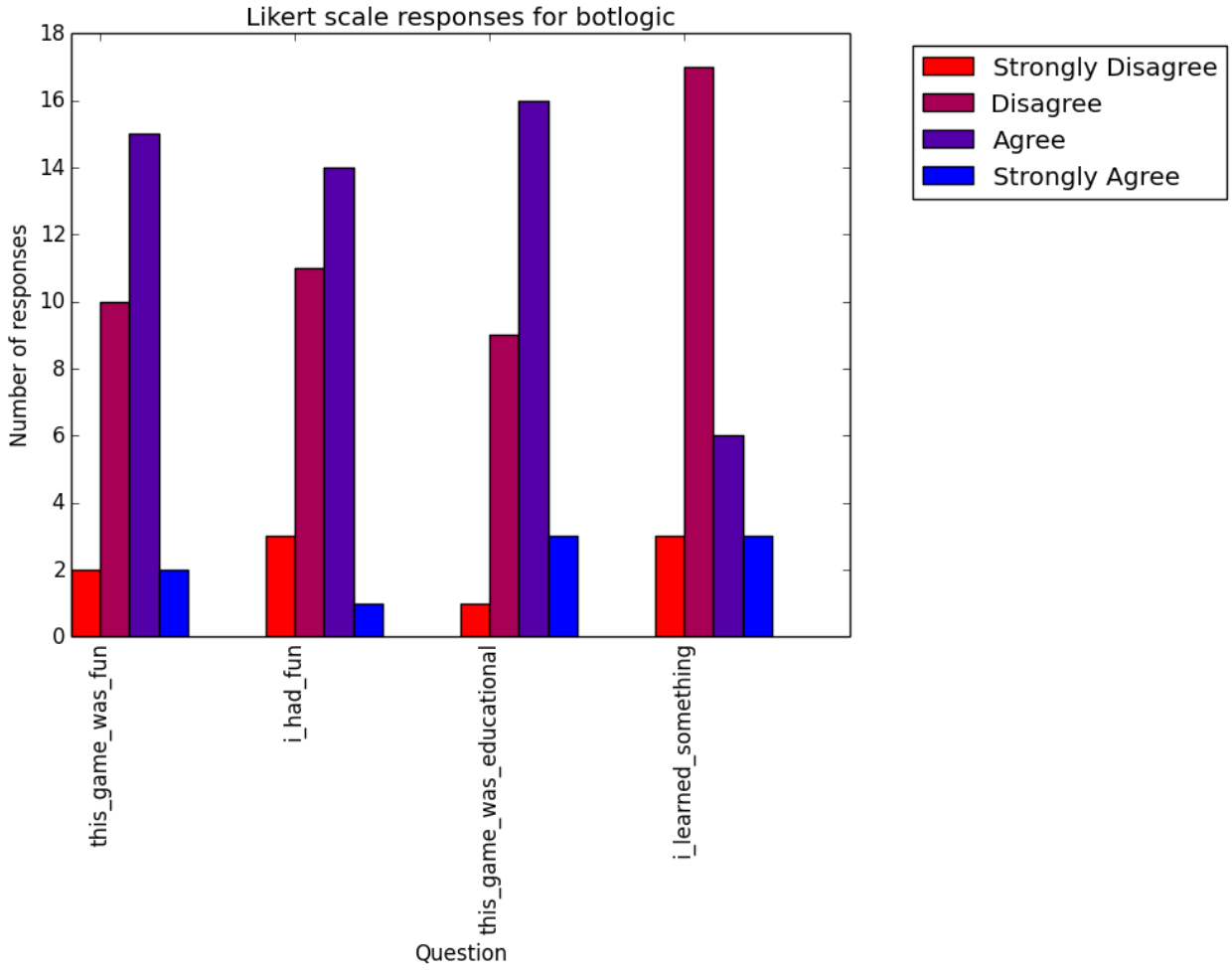

Figure 6.41: Botlogic

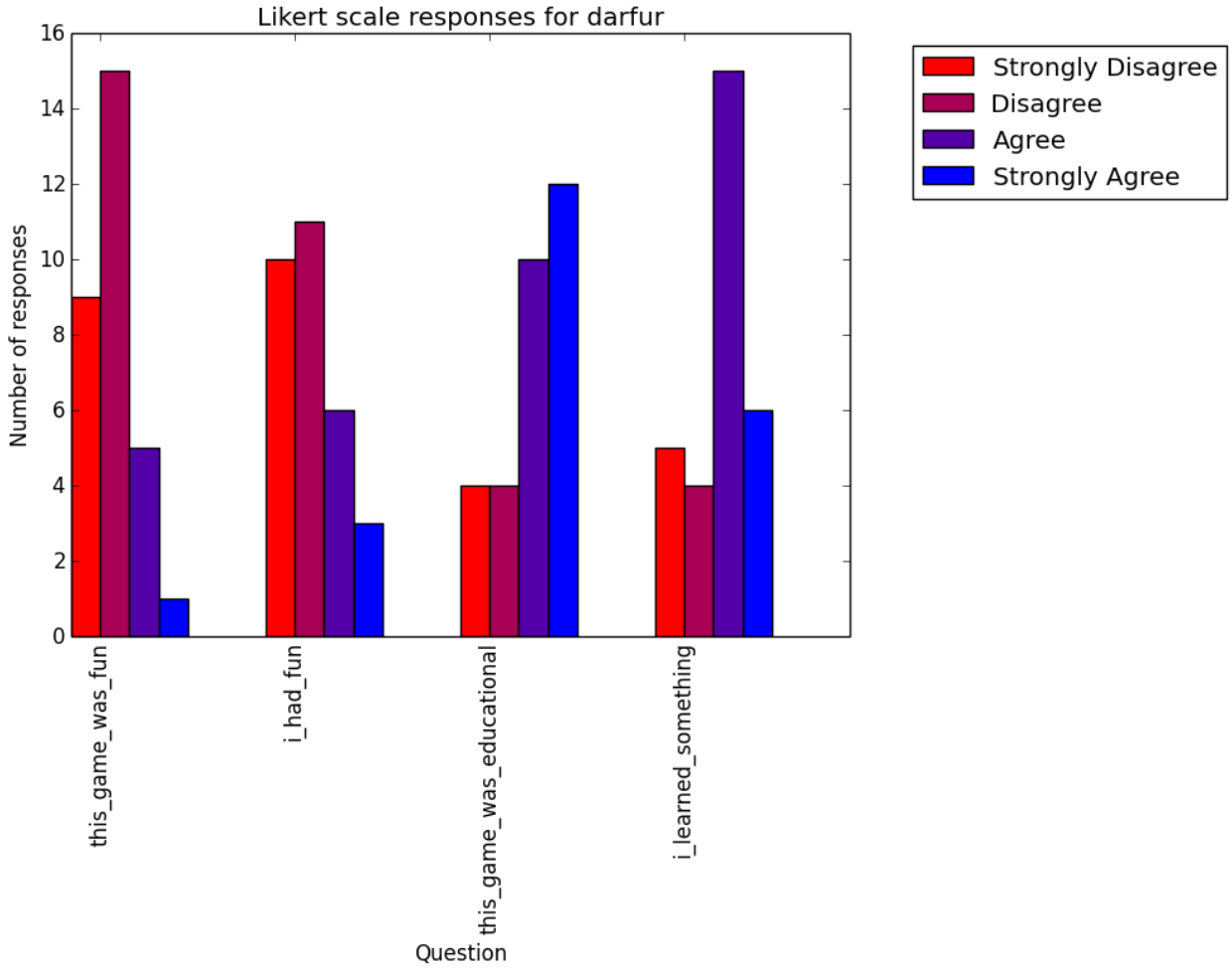

Figure 6.42: Darfur is Dying

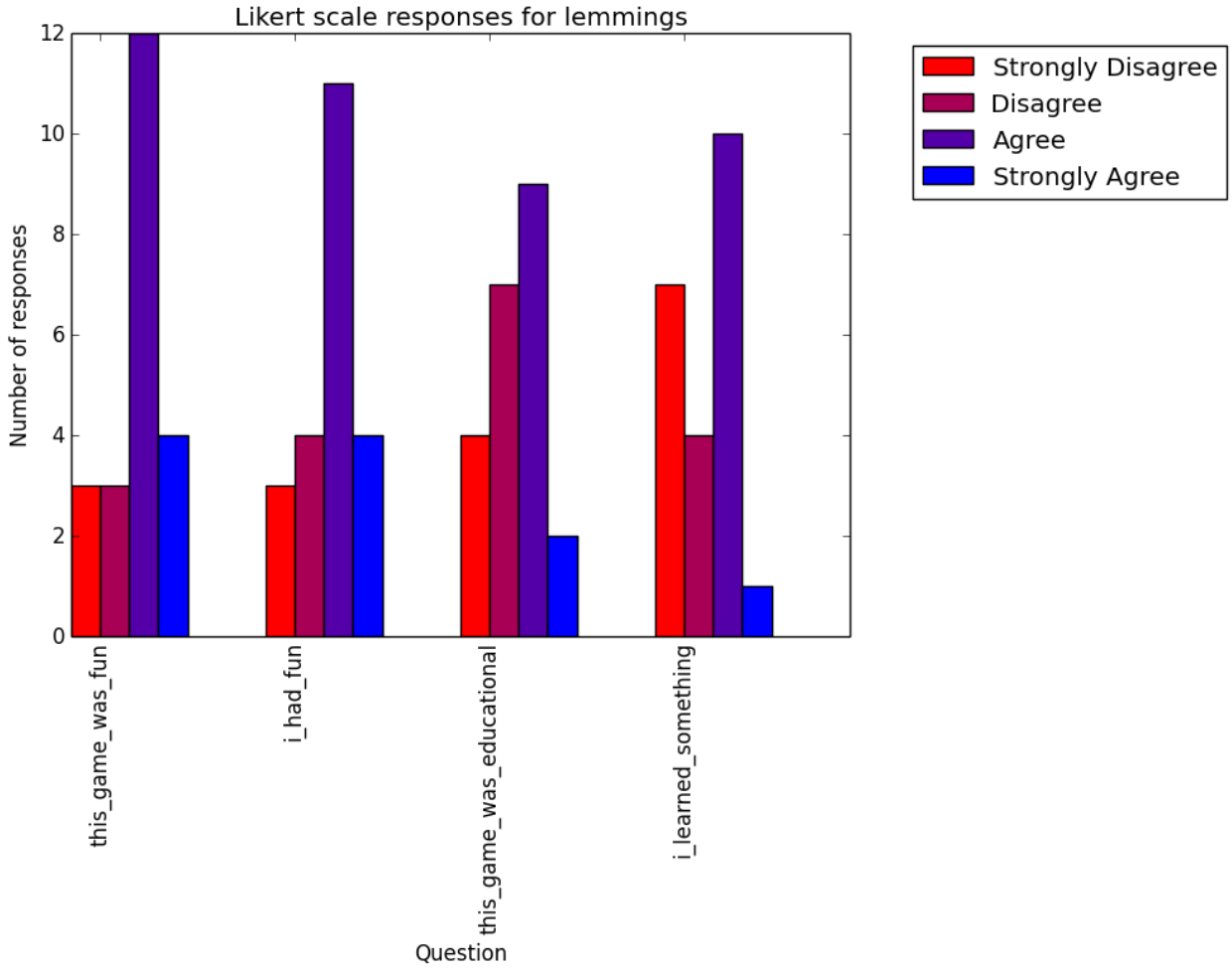

Figure 6.43: Lemmings

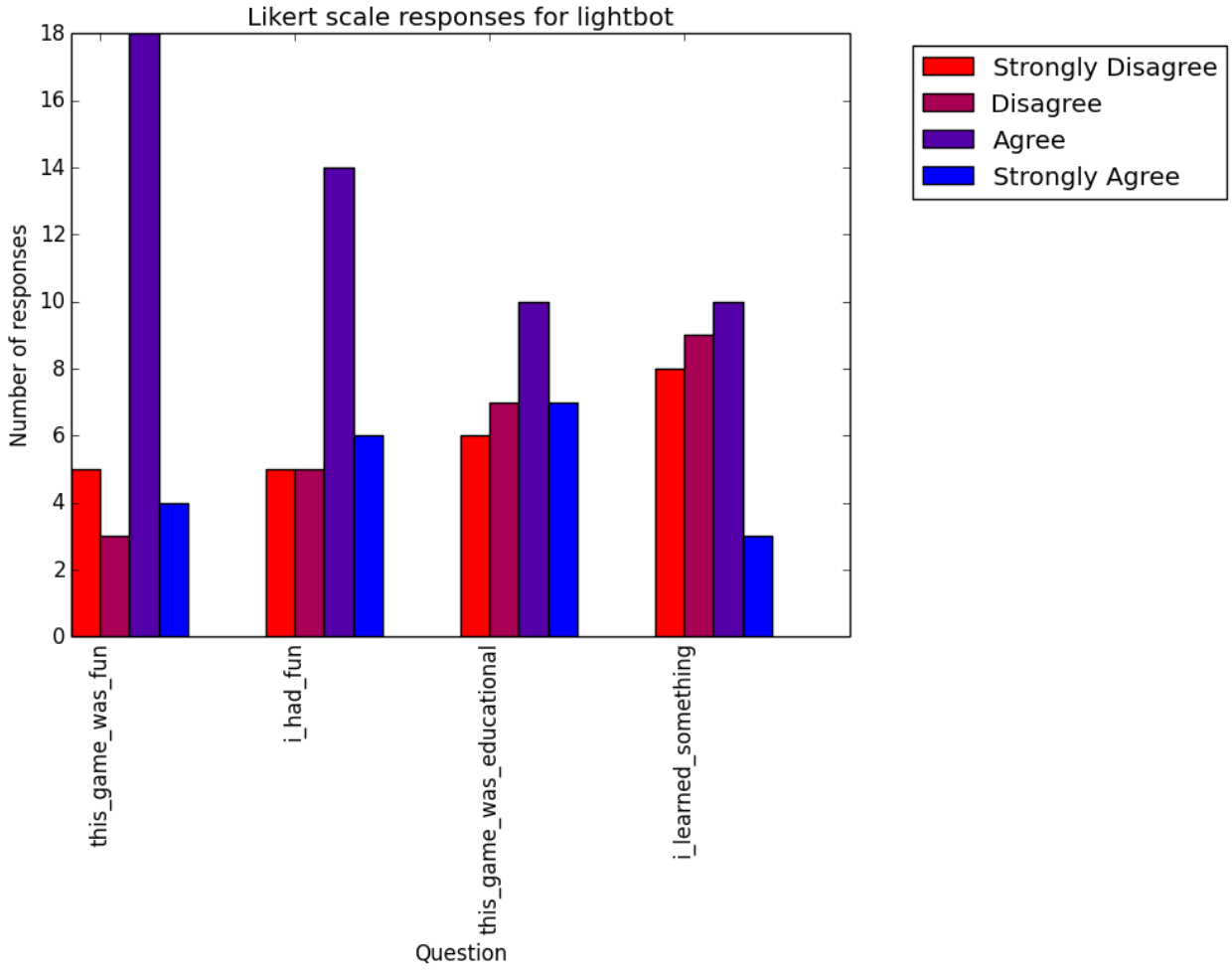

Figure 6.44: Light Bot

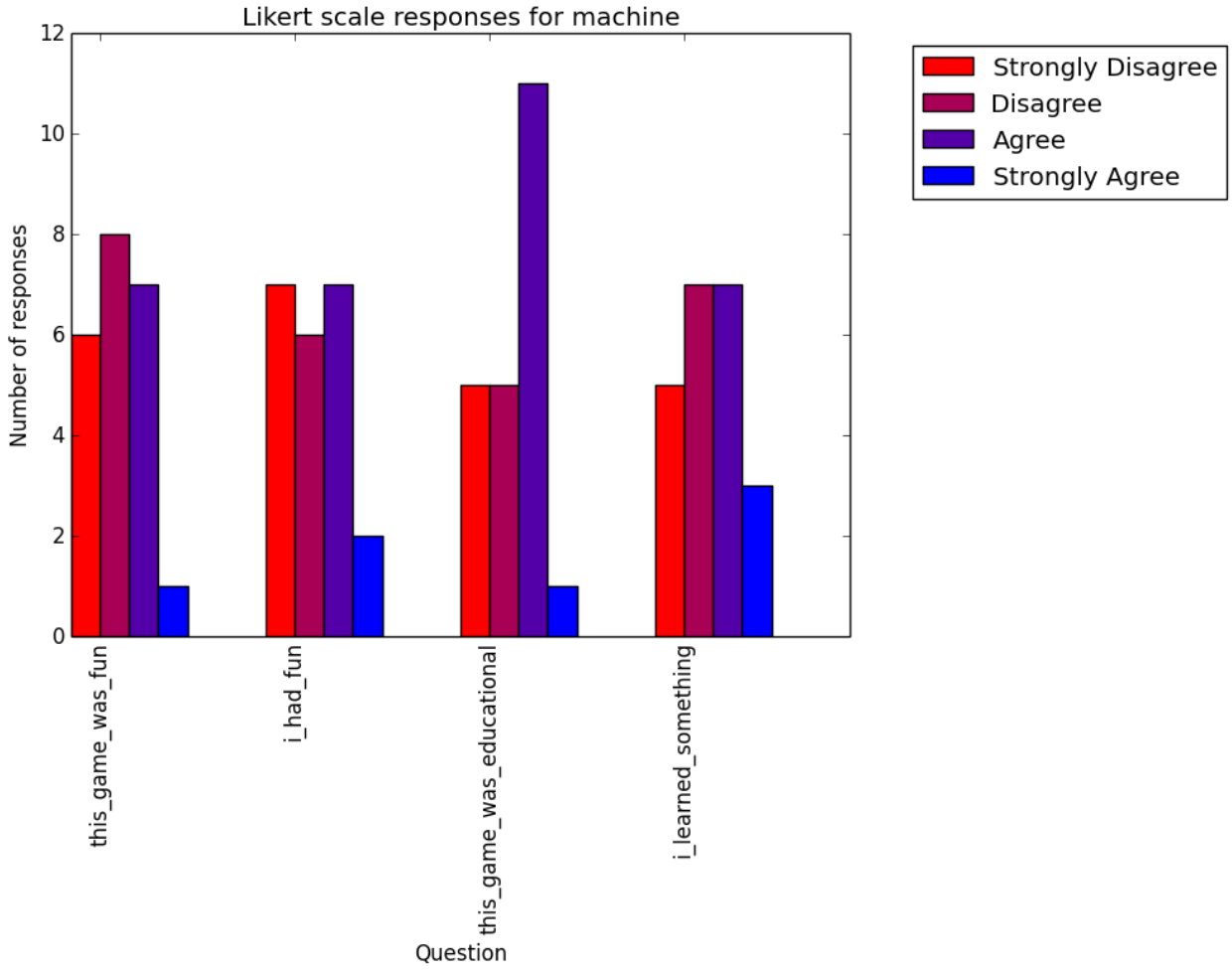

Figure 6.45: The Incredible Machine

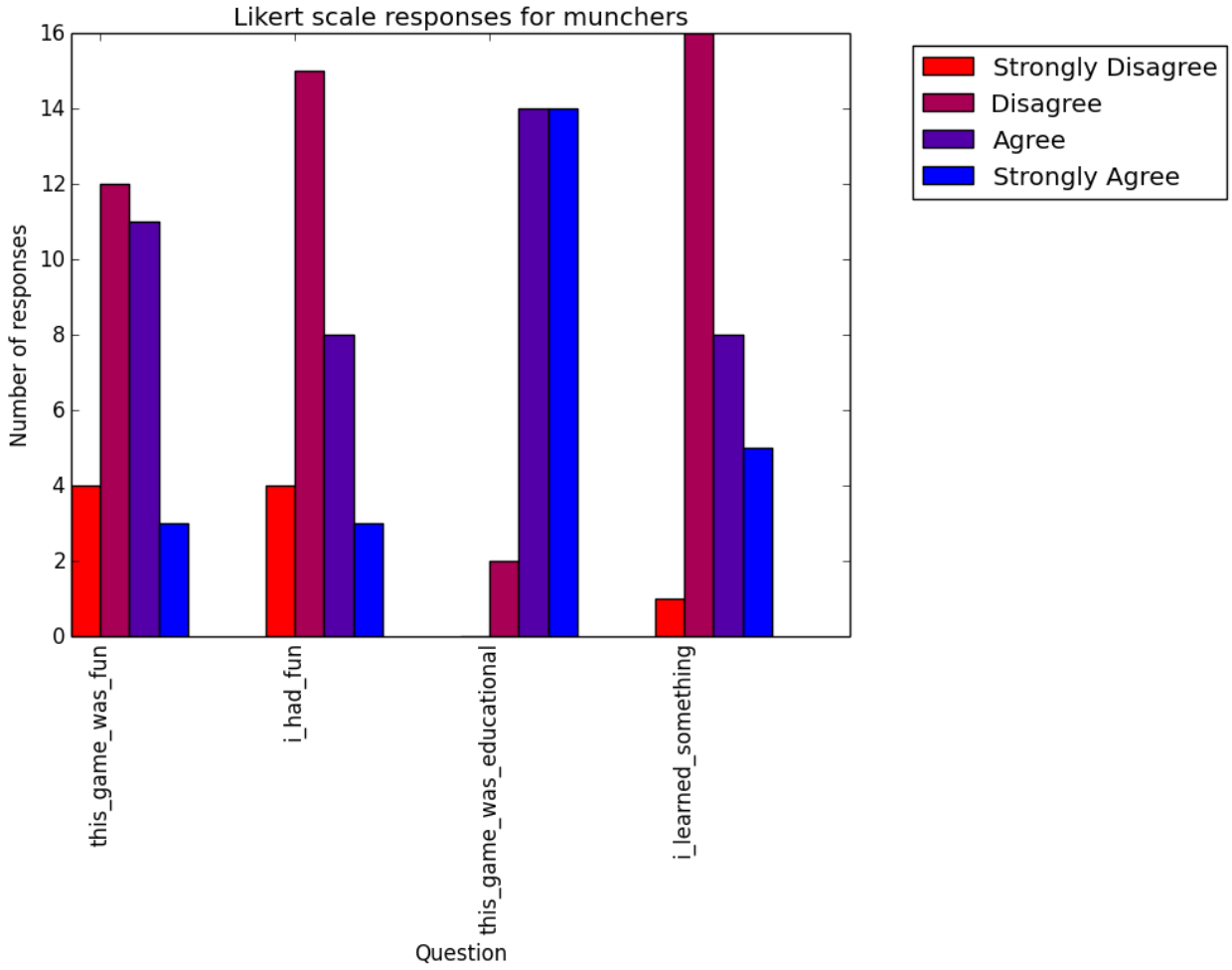

Figure 6.46: Number Munchers

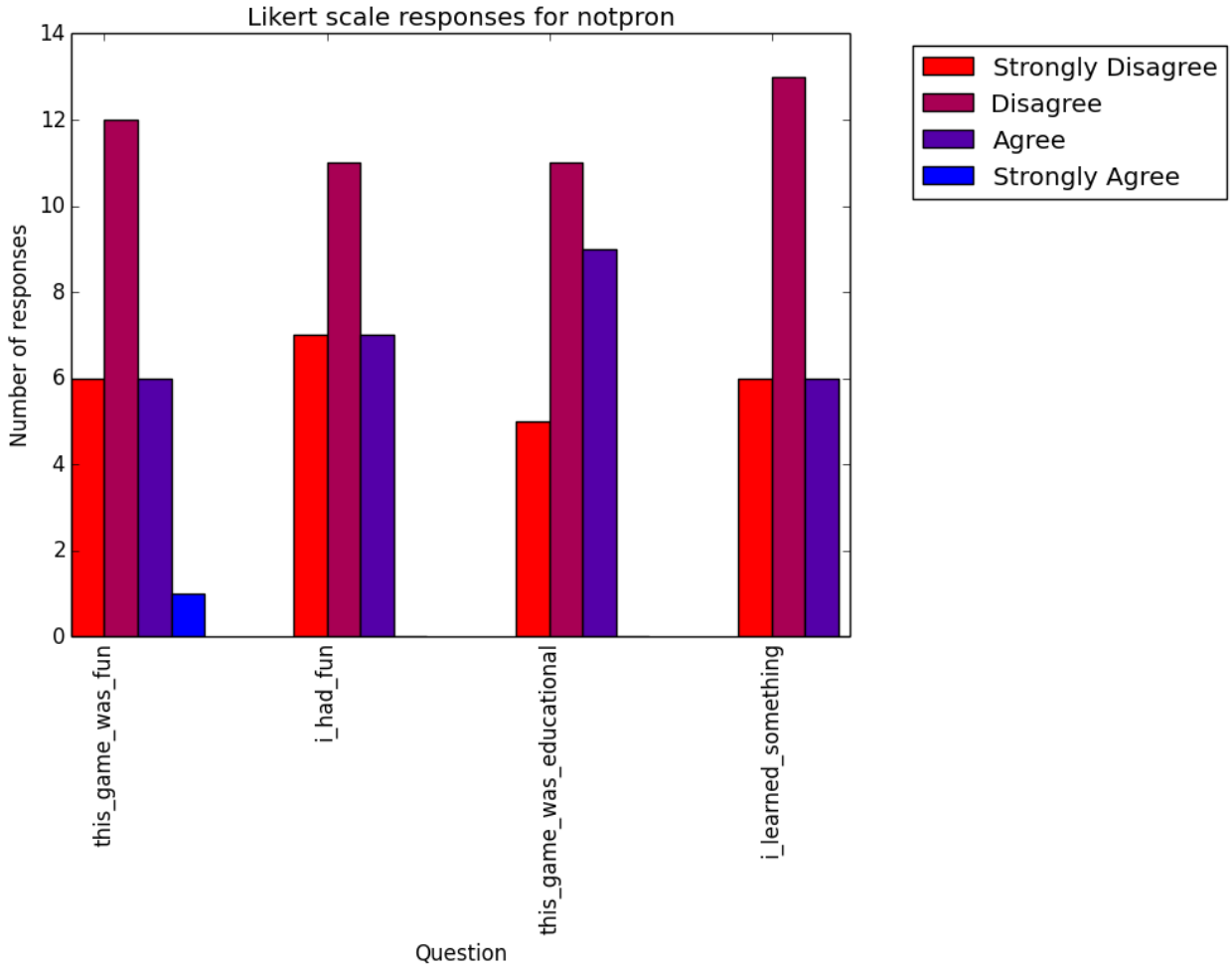

Figure 6.47: Notpron

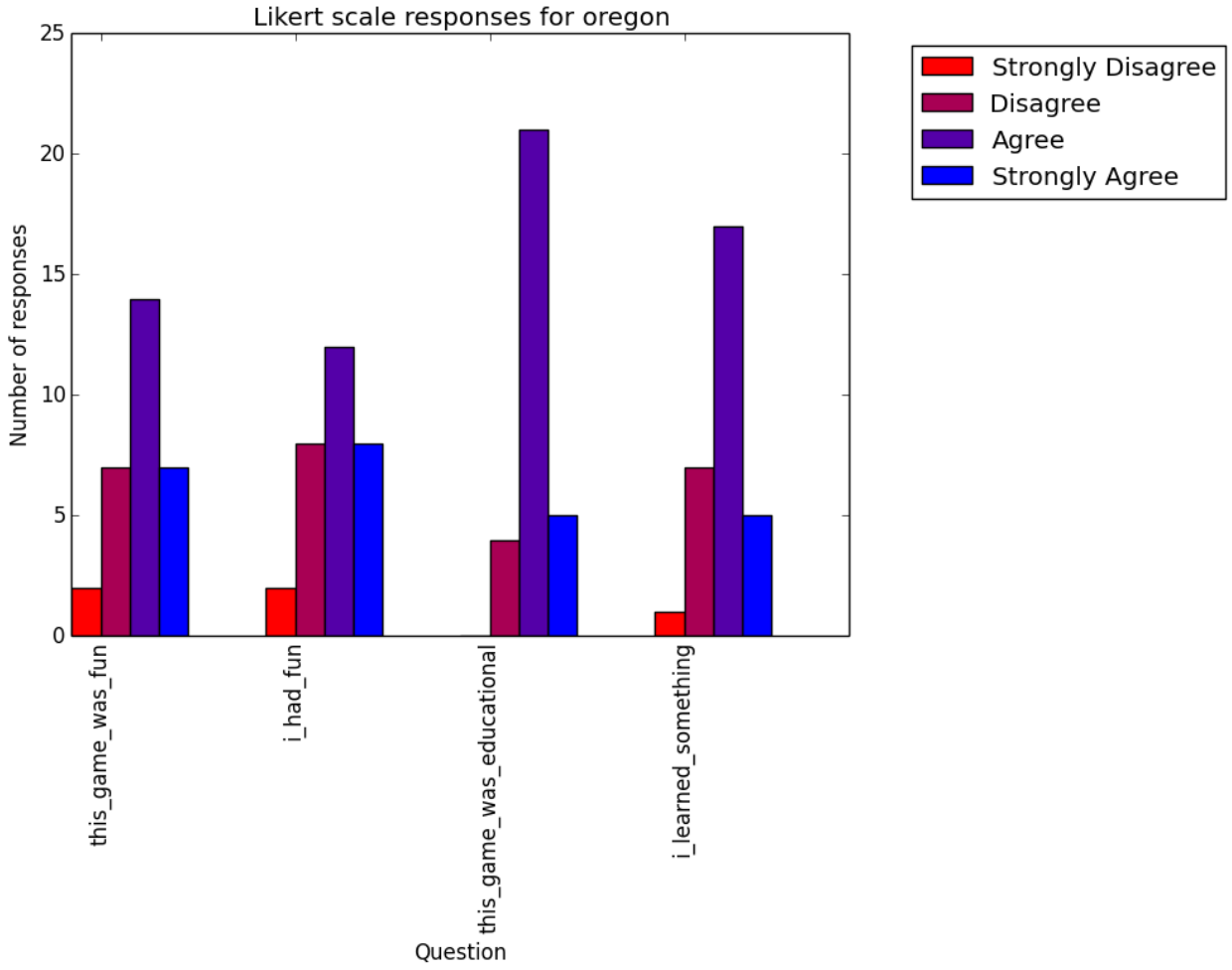

Figure 6.48: The Oregon Trail

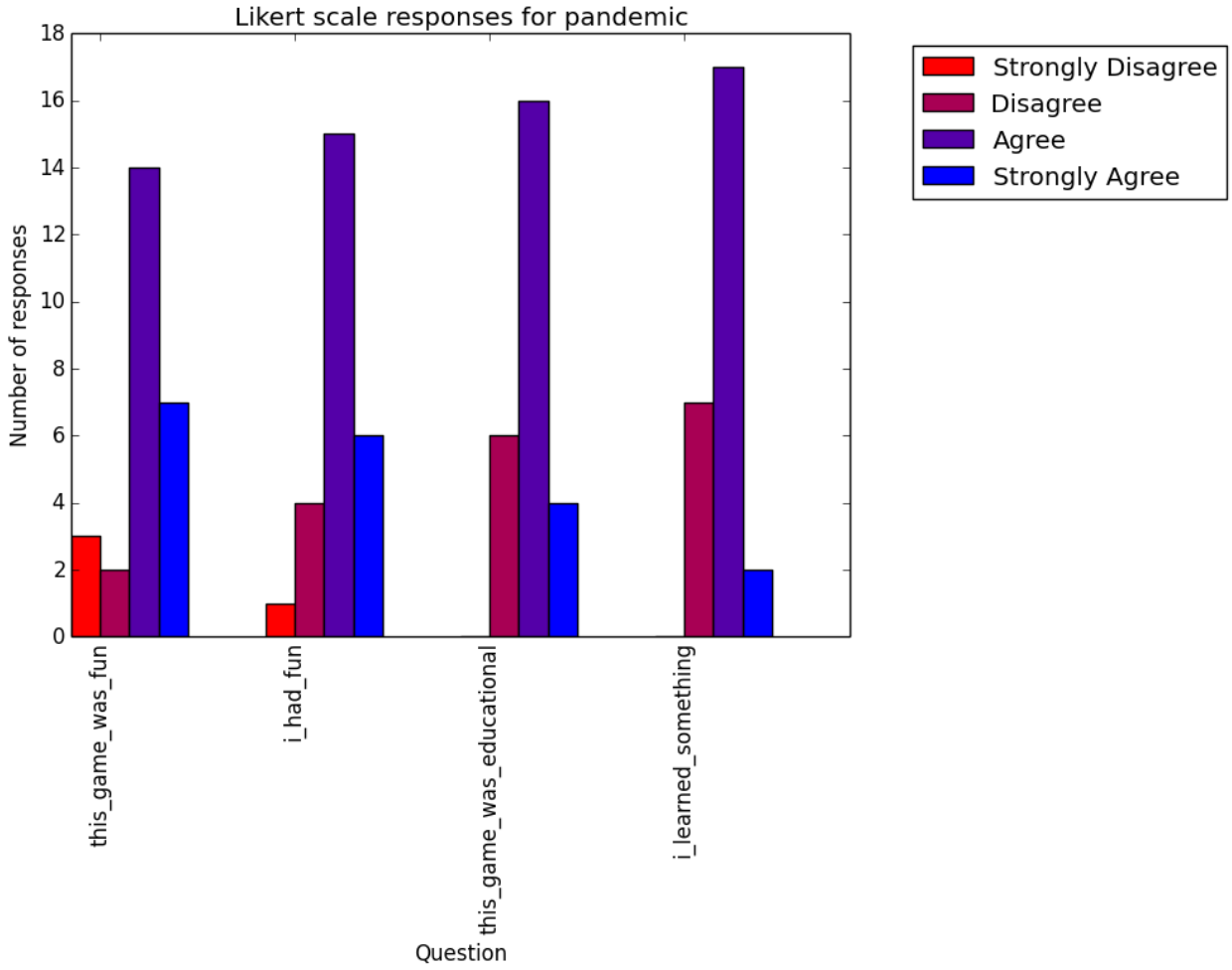

Figure 6.49: Pandemic 2

Chapter 7

Analysis

7.1 Quiz Scores

There are 4 games with quizzes, each with 20 responses each, 80 responses total. I graded each pre- and post-quiz, and ended up with the percent change for each response. After averaging these changes, I found that, after playing an educational game for a short period of time, the average player's score went up by 3%.

For the number of responses I've received, this is not a statistically signficant number. I could allocate more surveys, get hundreds or thousands more responses in order to try to make it significant, but my efforts are better spent trying to make a more accurate survey.

7.1.1 How analysis using a t-distribution works

7.1.2 Analysis of Quiz Scores using t-distributions

7.2 Rubric Inter-rater Reliability

7.2.1 Quiz Postmortem

There are a number of reasons why the results may have fallen so flat.

The quiz itself may have been faulty; perhaps the questions were too hard, too easy, or covered content that was not taught as part of the game. To combat this, the quizzes were designed to cover material that players were supposed to learn in the game, but it's possible that the multiple-choice format wasn't the best way to do that.

It's also possible that the games weren't actually effective in teaching the players anything at all. Because of how the games were designed, players didn't retain any relevant information about what the game was trying to educate them on.

Another option is that the platform on which the surveys were administered has faults. There are spammers on Mechanical Turk, and while it is possible to

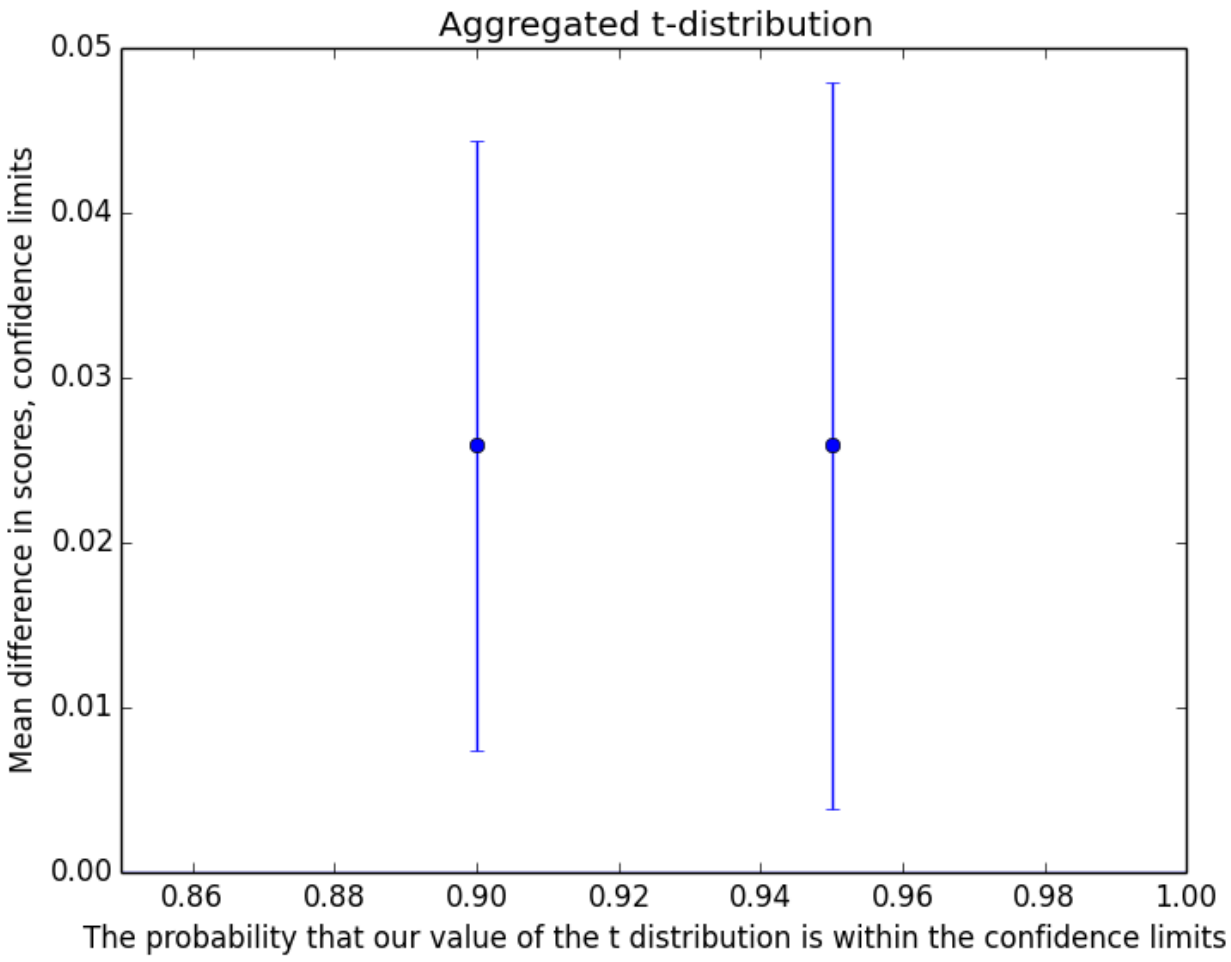

Figure 7.1: Aggregated tdist

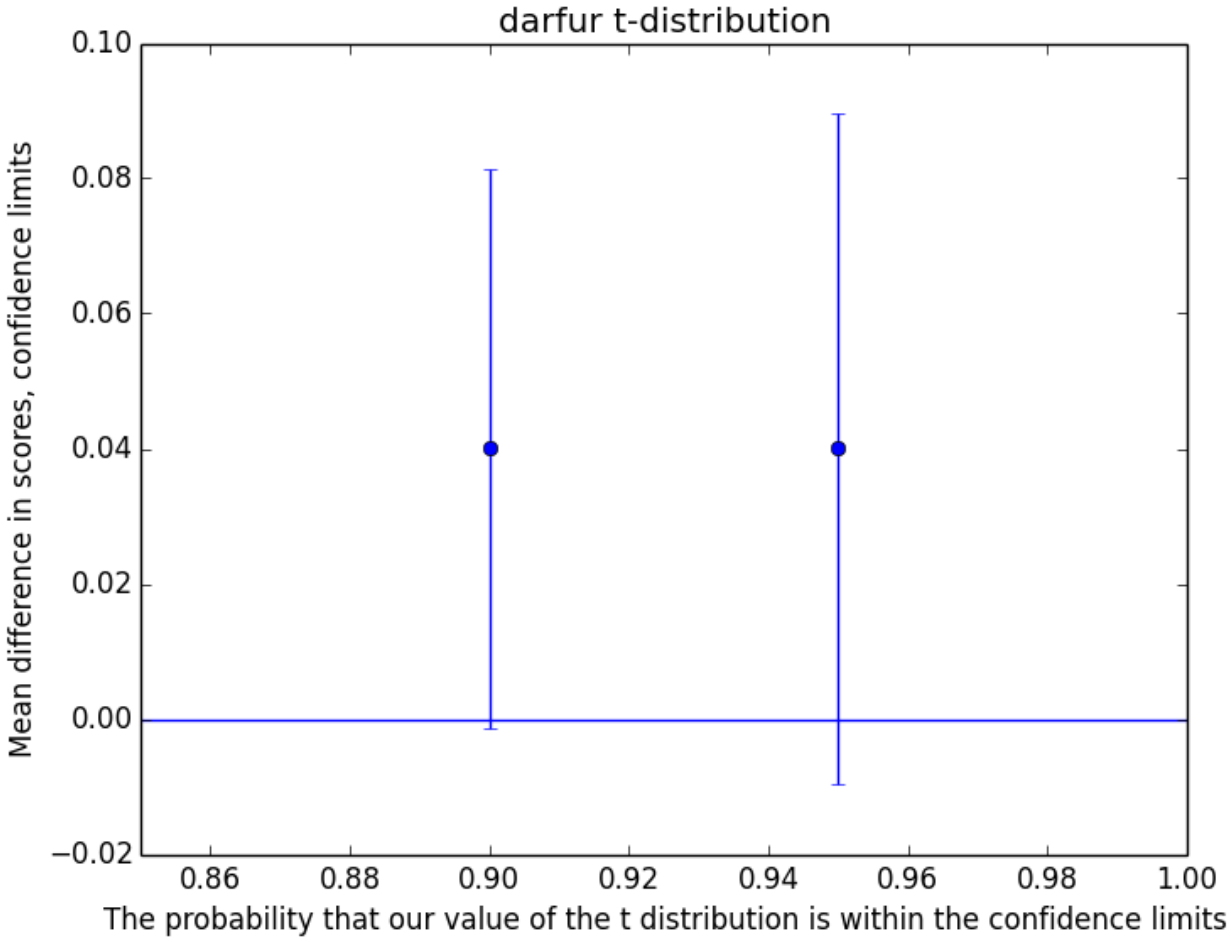

Figure 7.2: Darfur tdist

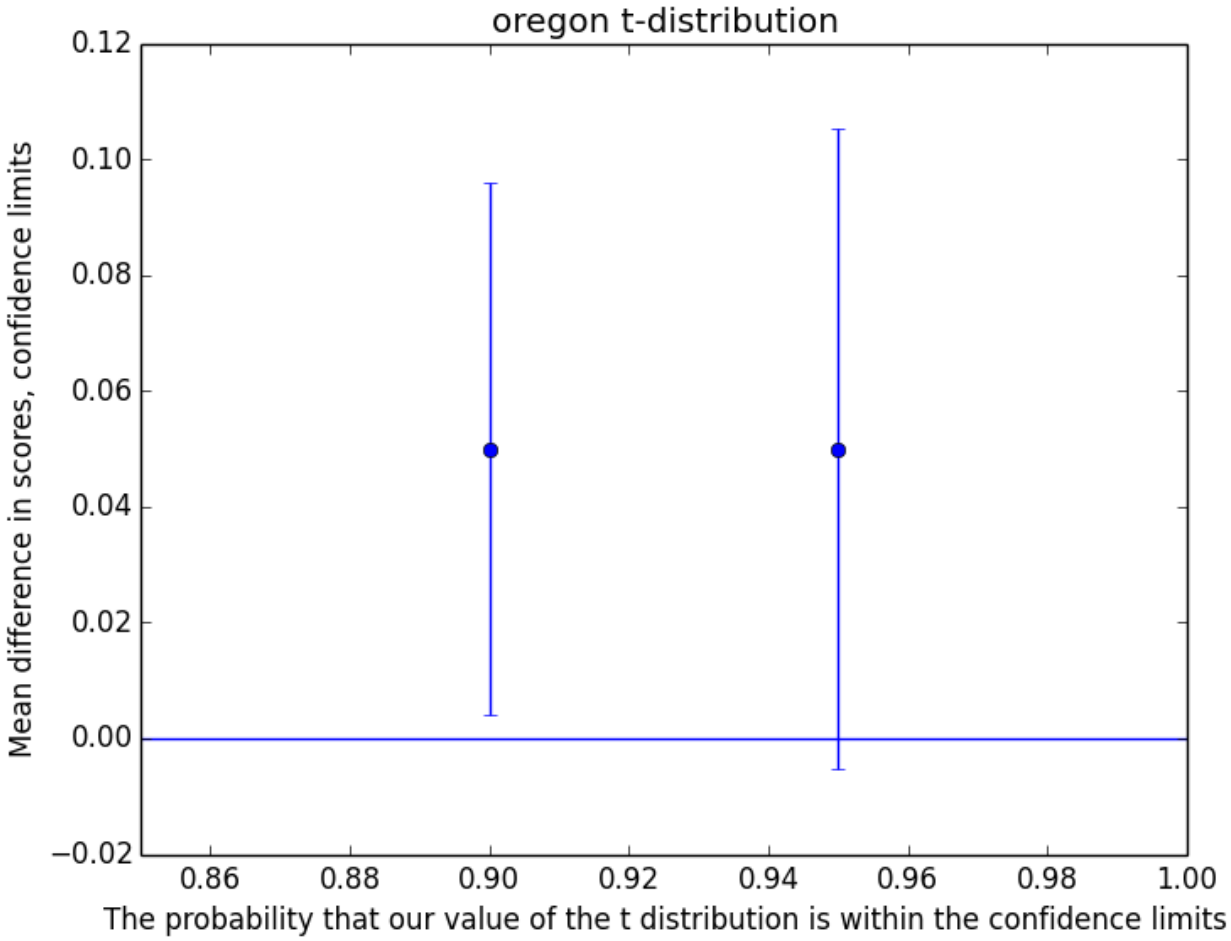

Figure 7.3: Oregon tdist

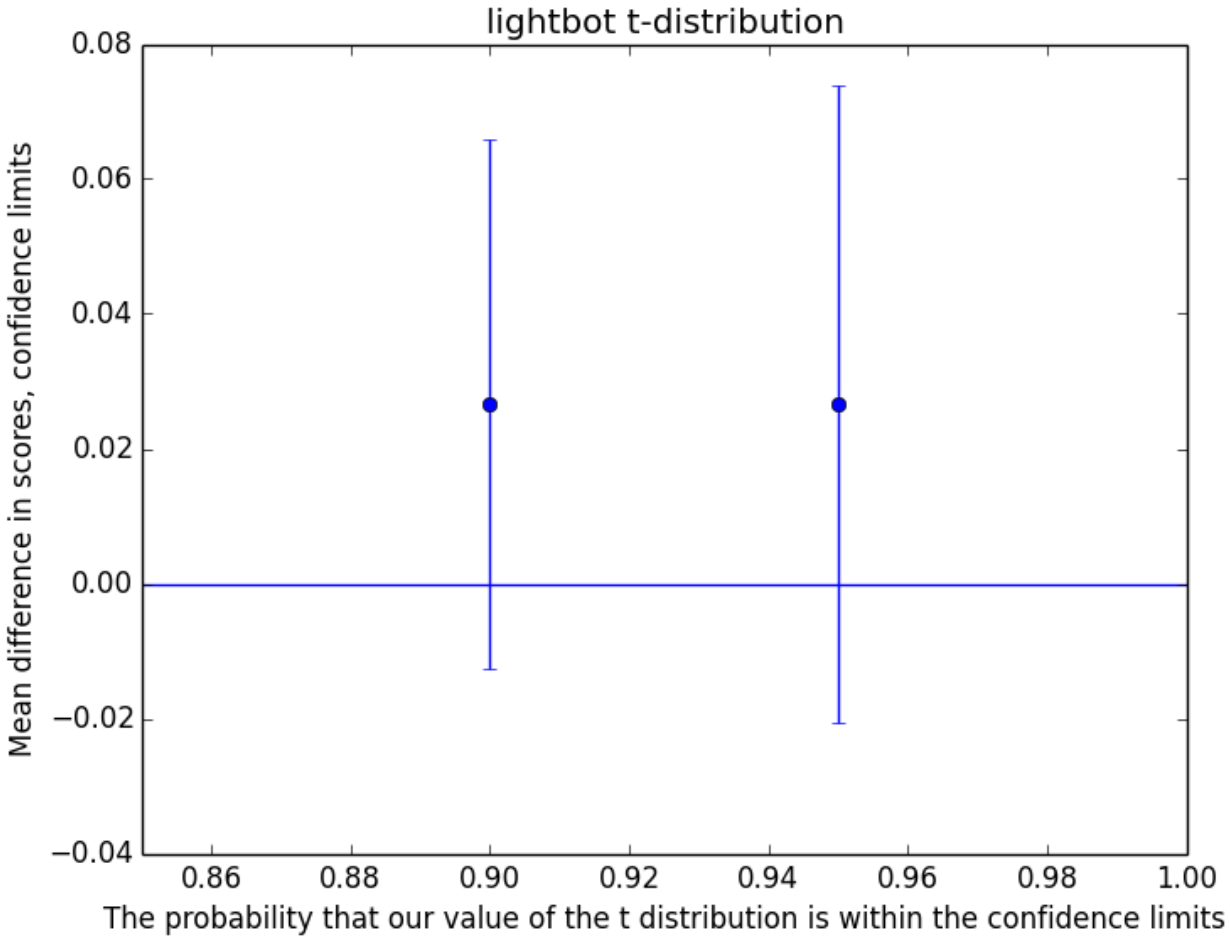

Figure 7.4: Lightbot tdist

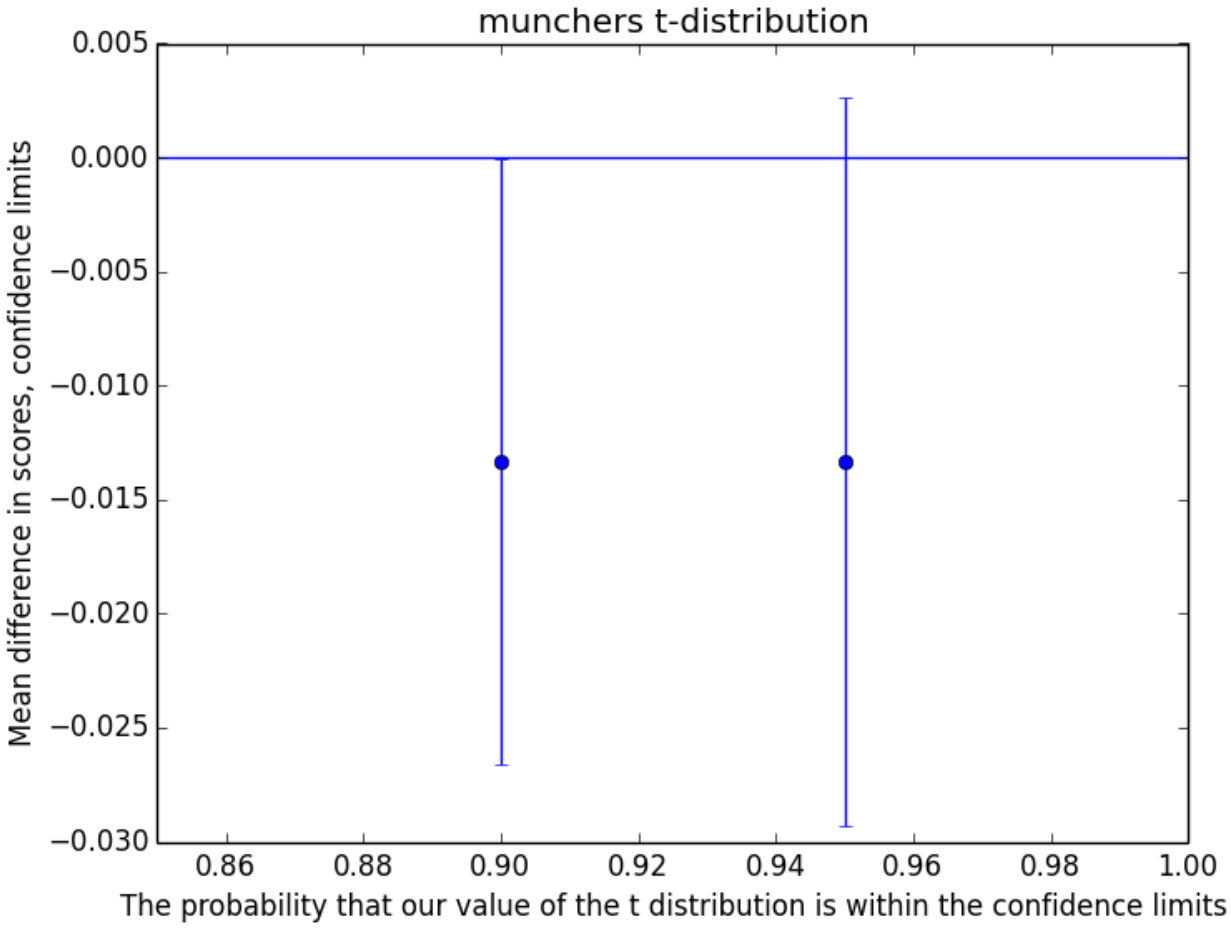

Figure 7.5: Munchers tdist

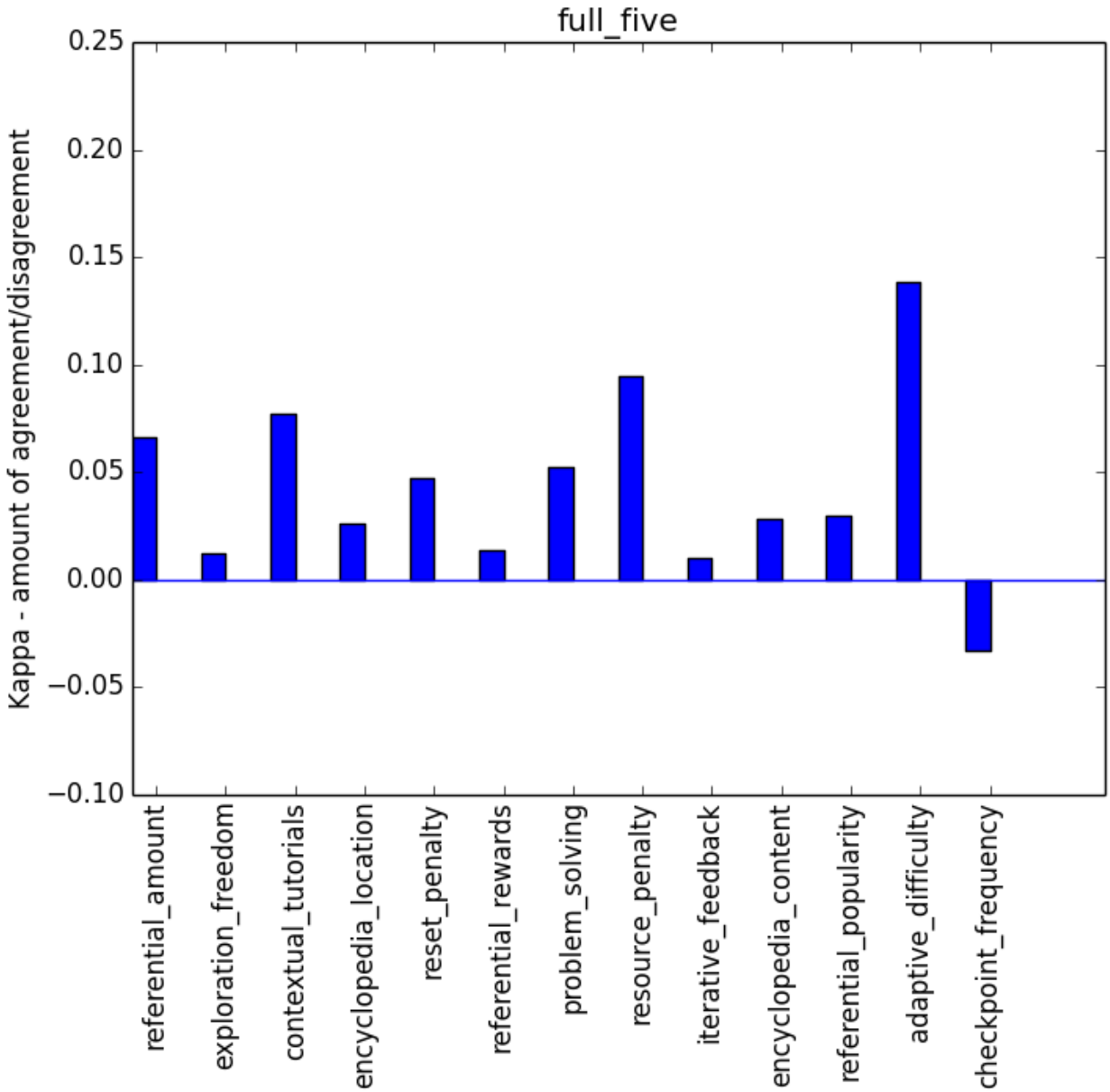

Figure 7.6: Inter-rater reliability

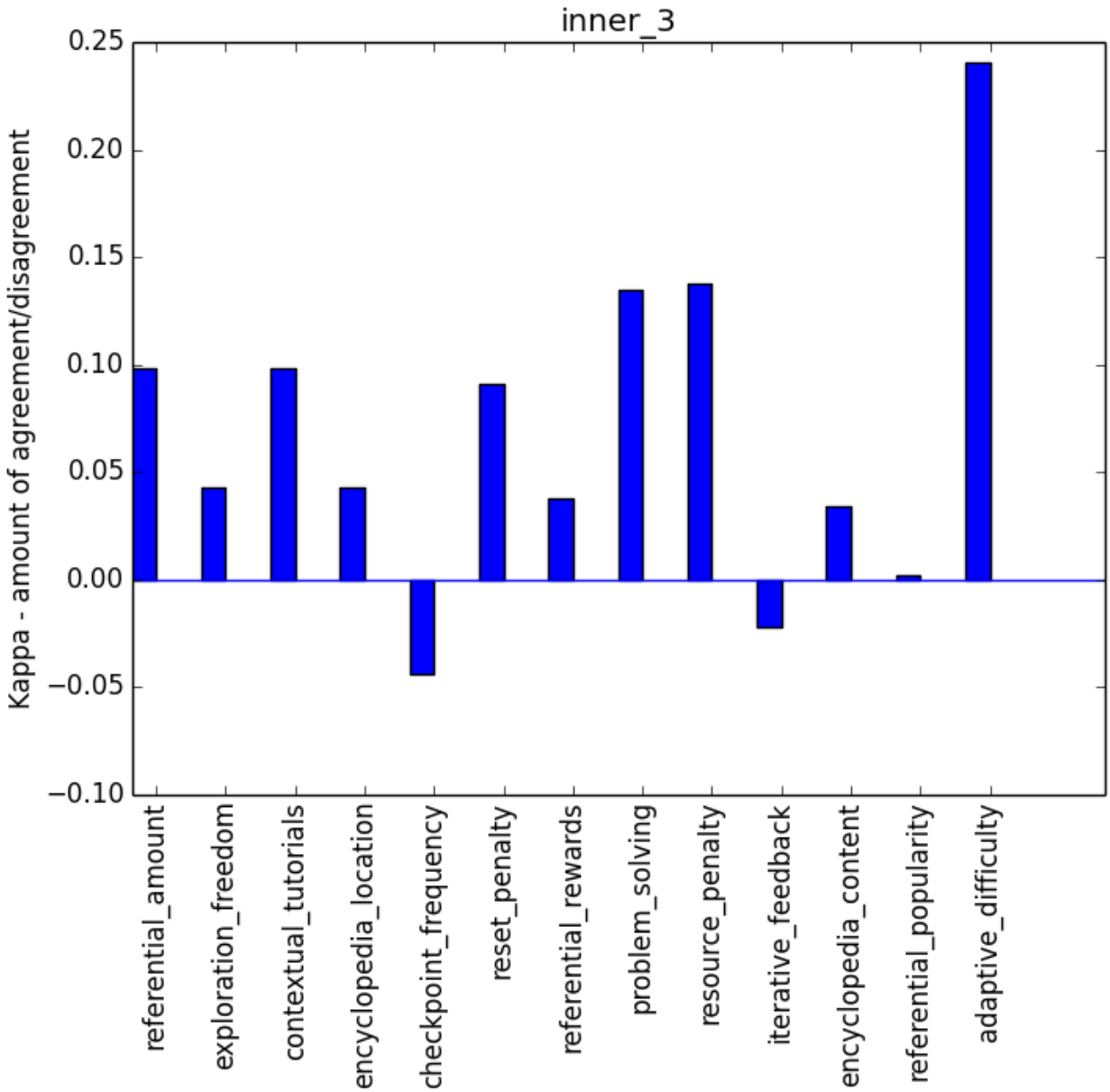

Figure 7.7: Inter-rater reliability with middle 3 options combined

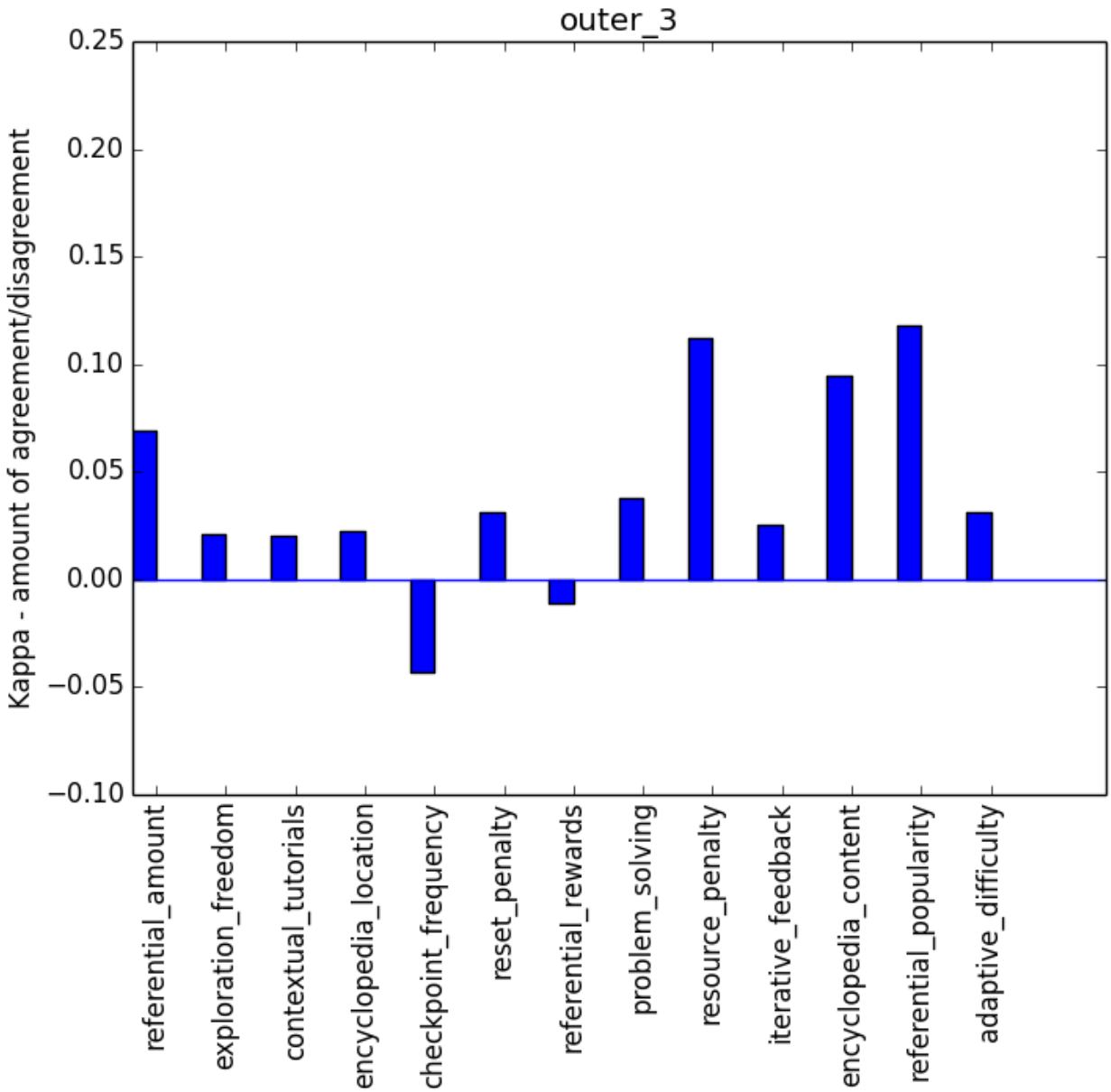

Figure 7.8: Inter-rater reliability with outer pairs combined

review and reject responses deemed unusable, it's impractical to filter out every single one. However, if a spammer is selecting random responses, it's highly unlikely that they'll end up with a higher or lower score than when they started.

It's also possible to filter out poorly rated workers with an "acceptance rate" criteria, where Workers that accept the task are required to have at least a 90% acceptance rate across all of their HITs, but that increases response times and payout amounts.

Finally, it's possible that some workers lost interest in the HIT once they had finished the first sections, resulting in shoddy work on the second. This would mean that some workers would guess at the questions in the post-quiz, resulting in worse performance across the board.

For a future quiz/survey on the Mechanical Turk platform, I'd recommend both an "acceptance rate" criteria, as well as some form of automated validation that the worker is actually completing the survey. This may include a "dummy question," where the reviewer only has to click the right button, or some other kind of verification where they submit a screenshot of their highest score within the game. This will increase response times and require higher payout amounts, but if it's focused only on one game it won't be much.

Another route worth exploring is paying out "bonuses" to workers who show exemplary work in the quiz and survey. However, you'd have to make sure to incentivize honesty and not greed; my hunch is that asking workers to not use the internet to find solutions may not work when there's extra payouts involved.

Chapter 8

Conclusion

At present, educational games do not actually increase the content knowledge of their players. Across 80 surveys, the average score of a pre- and post-quiz increased 3%; not a statistically significant amount. Further measures can be taken to increase the quality of the survey and quiz, the effectiveness of using Mechanical Turk as a platform for research, and focusing the research on one specific game with more responses.

8.1 What worked well

8.2 What didn't work well

8.3 What I learned

8.4 What to do next

Bibliography

[1] Breuer, Johannes, and Gary Bente. "Why so Serious? On the Relation of Serious Games and Learning." Eludamos. Journal for Computer Game Culture 4.1 (2010): 7-24. Eludamos. Journal for Computer Game Culture. Web. 02 July 2013.

[2] Buhrmester, Michael. "Amazon Mechanical Turk Guide for Social Scientists." Web. <http://homepage.psy.utexas.edu/homepage/faculty/gosling/reprints/MTurkhowto.pdf>.

[3] Coe, Robert. "It's the Effect Size, Stupid: What Effect Size Is and Why It Is Important." It's the Effect Size, Stupid: What Effect Size Is and Why It Is Important. British Education Index, 12 Sept. 2002. Web. 21 Nov. 2013.

[4] Csikszentmihalyi, Mihaly. "Finding Flow." Psychology Today. 14 June 2012. Web. 02 July 2013.

[5] Dondlinger, Mary Jo. "Educational Video Game Design: A Review of the Literature." Journal of Applied Educational Technology 4.1 (2007). Print.

[6] Floyd, Daniel, and James Portnow. "Tangential Learning." Season 2, Ep. 9 - Tangential Learning. Web. 02 July 2013.

[7] "Genocide in Darfur." Game. Fun Trivia. Web. 21 Nov. 2013. <http://www.funtrivia.com/playquiz/quiz2560151d4fc10.html>.

[8] Hallgren, Kevin A. "Computing Inter-Rater Reliability for Observational Data: An Overview and Tutorial." NCBI. U.S. National Library of Medicine, 23 July 2012. Web. 21 Nov. 2013.

[9] Hyman, Paul. "Communications of the ACM." Software Aims to Ensure Fairness in Crowdsourcing Projects. Communications of the ACM, Aug. 2013. Web. 21 Nov. 2013.

[10] Kjell, Bradley. "Counting Loop Quiz." Counting Loop Quiz. Web. 21 Nov. 2013. <http://www.cs.iastate.edu/~honavar/JavaNotes/Notes/chap16/chap16quiz.html>.

[11] "The Measurement of Inter-rater Agreement." Web. <http://hpm.fk.ugm.ac.id/hpmlama/images/Biostatistik/Tutorial_4_AS/2-chapter18.pdf>.

[12] Paras, Brad, and Jim Bizzocchi. "Game, Motivation, and Effective Learning: An Integrated Model for Educational Game Design." Proc. of DiGRA 2005 Conference: Changing Views Worlds in Play. 2 June 2005. Web. 2 July 2013.

[13] "Pioneer Life." ThinkQuest. Oracle Foundation. Web. 21 Nov. 2013. <http://library.thinkquest.org/J001587/>.

[14] "Quiz: The Oregon Trail." Quia. Quia. Web. 21 Nov. 2013. <http://www.quia.com/quiz/462983.html?AP_rand=227719871>.

[15] "QuizMoz - Oregon Trail Quiz." QuizMoz - Oregon Trail Quiz. Web. 21 Nov. 2013. <http://www.quizmoz.com/quizzes/Interesting-Facts-Quizzes/o/Oregon-Trail-Quiz.asp>.

[16] "What Do You Know about the Darfur Genocide?" What Do You Know about the Darfur Genocide? ProProfs. Web. 21 Nov. 2013. <http://www.proprofs.com/quiz-school/story.php?title=what-do-you-know-about-darfur-genocide>.