Who Wants to Be an A+ Student?

Building a Review Game for Introduction to Programming Systems

By Griffin Telljohann

Advised by Robert Dondero

Introduction

Exam review is a process that can be stressful for students, even if they have felt reasonably confident in the materials when they were taught. Especially at a university like Princeton, many tests are designed to be hard enough to produce a specific distribution of scores, which fuels anxiety even for tests that are not like that. Luckily, some teachers actually want their students to be able to accurately display their knowledge of the course material, so they do their best to prepare review sessions for the course that will help the students to recall all of the topics covered on the exam.

However, despite the best efforts of the instructors from certain types of classes, review sessions can be quite difficult to plan effectively. For classes that regularly assign problem sets, the tests are likely comprised of similar types of problems as the weekly assignments. In such a class, the instructor can hold a review session that consists only of answering questions and doing example problems. Because the concepts in these classes are used directly on weekly problems, students know in advance what they need help with. On the other hand, a class such as Computer Science 217: Introduction to Programming Systems[1] is not conducive to doing example problems.

In the review sessions for COS 217, the students receive a sheet of paper with every topic that could appear on the exam. This list tends to be fairly long and can be overwhelming. What follows is a review of many of those topics. The fear with such a review session is that it is too dry and that it ends up being an hour of the preceptor talking through these topics without much active interaction with the students.

Because of the concerns that conventional review sessions create for a class like COS 217, it would be helpful to come up with a method of review that can be more interactive,

[1] Abbreviated COS 217

interesting and effective. To fulfill these three criteria, one solution is to create something with a format similar to a game show. However, rather than being a game show with a single contestant, such a game would need to include many players at once. Also, it would need to incorporate relevant information inside the wrapper of fun. These requirements lead to "Who Wants to Be an A+ Student", an interactive review quiz game that can be accessed by each student on his[2] laptop at the review session and can be administrated by the preceptor. The game can include everyone at the session interactively, but can also indicate to the preceptor which concepts need more in depth review. "Who Wants to Be an A+ Student", because it is interactive, engaging, and effective in identifying topics needing review, should be an effective tool for the review sessions of COS 217: Introduction to Programming Systems.

Previous Work

There is a fair amount of precedent in games that are knowledge based or that are designed to teach. Some, such as the JumpStart series of computer games by Vivendi Universal, are overtly centered on "jumpstarting" the player's skills in a certain field. A few examples of this series are JumpStart Typing, which gives children practice typing to improve their skills, and JumpStart 3rd Grade, which allows the players to develop the knowledge that they need for 3rd grade. Others of these types of games are knowledge-based game shows, which are not based on certain categories of facts, but instead reward contestants for their memory of things that can be quite trivial. Some game shows have become very popular in the past, which could be a result of a couple of things. First, people may enjoy going through an experience of risk and uncertain outcome with the contestants, especially without any risk to themselves. Another possibility is

[2] For the purposes of referring to any generic person, male pronouns will be used, not from any lack of respect for female computer scientists, but just from the fact that the author of this report is male.

that viewers just like to feel good about doing better and knowing more than contestants, at least in some categories of question. Both learning-centered games and game shows have a good precedent for engaging people, so using them as models for "Who Wants to Be an A+ Student" is a good choice.

One possible reason that learning games are successful is the idea that active learning is more successful than passive learning. One article on active learning says that it means that "students must do more than listen: They must read, write, discuss, or be engaged in solving problems" (Bonwell & Eison, 1991). Review sessions occur, in theory, after students have already learned the course material, so at that point it is too late to employ active learning to teach, but active learning can be used in the review process. Bonwell & Eison explain the rationale for using active learning:

> For example, several studies have shown that students prefer strategies promoting active learning to traditional lectures. Other research studies evaluating students' achievement have demonstrated that many strategies promoting active learning are comparable to lectures in promoting the mastery of content but superior to lectures in promoting the development of students' skills in thinking and writing. Further, some cognitive research has shown that a significant number of individuals have learning styles best served by pedagogical techniques other than lecturing (Bonwell & Eison, 1991).

Even though it is too late for the active learning that will benefit the majority of people in the class, it may still be possible to employ active review in the review session. As it is, the students who attend the review sessions are not active participants in it, but are mainly involved passively. Just as active learning is more effective in the classroom, so too should active involvement and engagement be effective in a review session. This makes an interactive game that includes the whole class a logical choice for a potentially successful review session.

The use of interactive quizzes and games to foster active engagement in classes is not without precedent. In particular, there are two articles that are both on this topic, one of which

talks about game shows in particular as learning tools and the other of which discusses using quizzes in lecture classes to make the lecture more interactive. The first of these begins by discussing a growing need for teachers who will focus more on facilitating active learning than simply allowing students to passively absorb what is presented in lecture-style sessions. Sarason & Banbury explain, "This philosophy toward more active learning suggests that the aim of teaching is not to transmit information but to transform students from passive recipients of other people's knowledge into active constructors of their own and others' knowledge" (Sarason & Banbury, 2004). This applies the concept of active learning to the classroom through the use of game shows. Sarason & Banbury alter the game shows so that they allow everyone to participate, either in groups or individually. They base these classroom games on *Who Wants to be a Millionaire?* and on *Jeopardy,* from which they intend to borrow format and the enthusiasm that students have about those two game shows in particular. They write, "Television represents a common experience for many of our students . . ., and there is evidence that they are cognitively engaged when watching television. . . . We propose that using a game-show format in the classroom leverages this cognitive engagement and facilitates translating the embedded lessons into an active learning experience" (Sarason & Banbury, 2004). Sarason & Banbury's reasoning for modeling their classroom games on *Who Wants to be a Millionaire?* and *Jeopardy*, are to harness the students' enthusiasm and to allow all students to be involved, reasons that are similar to those that motivated "Who Wants to Be an A+ Student"

The second article pertaining to interactive and engaging learning gives some perspective on the results of using certain interactive methods in a lecture setting. The reporter, Dan Carnevale, investigates the use of "clickers", which have several buttons for input from each individual in the class and transmit that input via receivers on the wall to Biology instructor

Mark Coykendall's screen (Carnevale, 2005). Instead of changing the entire format of the lecture to something discussion based or a session driven by the students, the featured professor uses these "clickers" to pose quiz questions to the audience that can be answered quickly and anonymously, which he uses to accomplish two goals. First, because these questions are graded, it forces students to make an effort to pay attention and learn the material as it is taught rather than catching up at the end of the semester. A related side effect is that, "Since he began using the clickers two years ago, Mr. Coykendall says he has noticed a drastic decrease in students' nodding off in the back of the room" (Carnevale, 2005). The second goal is to give feedback to the professor over the course of the lecture. This feature is interactive in a different way, facilitating a limited dialogue between the class and the professor and allowing him to tailor parts of the lecture on the fly to respond to a general lack of understanding among the students. The decision to make the lectures interactive "has received mostly positive feedback, although a few students gripe about the clickers" (Carnevale, 2005). From this example of interactive and engaging teaching, it seems that forcing the class to be more involved and allowing the instructor to base his or her presentation on class performance are beneficial to the learning experience as a whole. Each of these examples reveals part of what makes an effective learning and teaching tool, all of which can be incorporated into the specifications for the "Who Wants to Be an A+ Student" application.

Requirements

There are two broad requirements of the "Who Wants to Be an A+ Student" game, but each has multiple aspects, some of which apply to both requirements. These broad requirements

of the game are that it engages all of the participants of the review session and that it effectively aids in the review of the topics that are on the course's final exam.

Engaging all of the participants of the review session means that each participant should be interested and attentive to the game and that each participant should feel like he is able to use and display the knowledge about each topic that he has in his memory. From this definition, several sub-requirements emerge. First, the review game must be fun and enjoyable to the participants. If the game is not fun, then not only is it not a game in the first place, but it is also more difficult for the game to act as a review tool. This is due to the fact that students can more easily remember times when they experience stronger emotions, namely the enjoyment of the game. The second sub-requirement involved in engaging all participants is that the game must allow enough time for all of the participants to remember the facts necessary for the current question. Some students remember facts more slowly than others, a fact that makes them less able to actively participate in typical review sessions. Similarly, naturally quiet students are also less likely to actively participate and benefit from typical review sessions, a fact that reveals the third, somewhat intuitive, sub-requirement, the necessity that the game has to be able to include everyone at the review session. The final sub-requirement is that the game should involve an aspect of competition, which will take advantage of the desire to be the best that most Princeton students possess.

Effectively aiding in the review of topics covered by the exam means that the game will give participants a feel for the types and categories of questions that will be asked on the exam and also reveal which of those topics the participants need most work in. The first sub requirement of this is that the game should force participants to evaluate their confidence in each category before seeing any questions in that category, while giving that evaluation some scoring

weight. Because evaluating confidence is important to the participant's final score, he will make an effort to be accurate. Also, knowing his confidence in each of the main categories that will appear will give the participant a good prediction of how well he could do on the exam. The second of these sub requirements is that the game should include questions that have a range of difficulty levels. This will help the participants to reevaluate their confidence based on how many questions they can answer without trouble. A third sub requirement is that the game should use some questions from old COS 217 exams so that the participants will be able to get accustomed to the style and type of question that could appear on the real exam. Finally, the game should give each participant an evaluation of their performance compared to their confidence so that he can figure out what categories need the most review.

There were a couple of potential requirements that were considered as well, but they ended up being thrown out. First, the requirement of having a game that makes it very difficult to cheat off of someone else's screen was discarded. This was not used because the review session is for the benefit of the participants, so if someone cheats in this game, it will be to their detriment rather than to that of the rest of the group. Second, the requirement of allowing students to enter the game once it has started was also ignored. It is possible to make the game short and fast moving enough that multiple rounds could be played in one review session, allowing late students to join at the beginning of a new round. For these reasons, these two requirements were not considered as part of the body of requirements.

Functionality

The functionality of "Who Wants to Be an A+ Student" is one that requires a simultaneous explanation of both the preceptor and student sides of the game. Each side of the

game is dependent on the other, so it makes most sense to describe the functionality of each piece by piece.

When the preceptor arrives at the classroom, he sets up his computer at the front, but without connecting it to the projector. He opens his browser and loads the first preceptor screen. This screen greets him and indicates the game ID number of the current game, as shown in Figure 1[3], so he can write it on the board for students to see.

As the students arrive, they open their browsers and go to the student start page, as depicted in Figure 2. This page requests a display name and the game ID number that the preceptor wrote on the board. The player should then read the provided instruction before clicking the "Ready!" button.

The player is then taken to the next page, shown in Figure 3, where he is asked to indicate his confidence in each of the categories that will appear in the game. The options given for each category are 25%, 50%, 75%, and 100%. The points the player will gain or lose on each question in that category depend on how confident they are in it. There is no 0% option to keep players invested in categories that they do not think they know. If there were nothing at stake, a player could skip those questions and miss the opportunity to find out how much he needs to study that category. Once the player submits his results, he is taken to a waiting screen.

As the players are going through these previous two pages, the preceptor will have advanced to his next page, which is displayed in Figure 4. This page displays an up-to-date list of players who have entered the game. It refreshes every few seconds and checks for players. Once a player has entered his name and the game ID number, he will show up on this list. Then, once that player has submitted his confidence level information, the entry will turn bold on this screen. Once that happens, the average confidence level for each category, which is displayed

[3] All screenshots appear in the appendix.

below the roster, will update with those new entries to reflect the average of all players in the game who have entered confidence levels. The preceptor will not be able to tell if everyone in the room has entered confidence levels, but he will be able to tell if everyone who has entered his name is ready. Once everyone is ready, the preceptor clicks on a button to start the first question.

Students, who have been waiting on a constantly refreshing page, are redirected to the page of the first question. As in Figure 5, students see a timer, the question text, and 4 answer choices. They have until the timer runs down to 0 to select their answer and submit it.

In the meantime, the preceptor sees a very similar screen, Figure 6, except that the correct answer is in bold and the timer begins with 5 extra seconds on it. To allow for slower browsers on the students' computers, the preceptor's timer has extra time so that every student's timer can reach 0 before their scores can no longer be updated.

When each player has either answered the current question or run out of time, he gets redirected to the answer page, on which the question and answers are displayed again. Now, if the player got the question correct, that answer will be in bold, green font and surrounded by text that tells the player that their answer was correct, as in Figure 7. If the player answered incorrectly or did not answer in time, the correct answer is in bold, blue font surrounded by text that indicates that it is the correct answer. If the player gave an answer, then that answer is in bold, red font and has text indicating that it was not correct, as in Figure 8. The players score is updated by taking his wager, which is 100 points multiplied by his confidence level for this category, and adding it to his score if he got the question correct and subtracting it otherwise.

Once everyone has answered the current question, the preceptors screen switches to display the question and answer text again, still with the correct answer in bold, but now with the

number of students who gave each answer immediately after the answer text. This screen is shown in Figure 9. This way, the preceptor can see how many people got the question right and if it is necessary to elaborate on the concepts tested by that question. Once he has finished that, he can click the button to take the students and himself to either the next question or the results page if the class has already completed each question from each category.

The student results page begins with a class high score board, where the player will see his own score as well as the display names and scores of the top 5 scoring players, as displayed in Figure 10. The motivation of showing a leader board is to tap into the competitive nature of which most Princeton students have an overabundance. From there, the player can go to a personal results page, an example of which is found in Figure 11. On the personal results page, the player will see, for each category, the number of questions answered correctly, the number of questions encountered, the percentage of questions answered correctly, and the confidence level that he gave at the beginning of the game. Here, the player can compare how he did in a category to how he thought he would do in that category, which will help him in deciding later on what topics to focus his studying.

While the students see those results pages, the preceptor sees the entire list of players, sorted by score. Each player's score and average confidence level is displayed, as in Figure 12, so the preceptor can see who did well and how confidence correlates with success. This is the functionality of "Who Wants to Be an A+ Student", which attempts to effectively engage all students through allowing them all to participate in the game simultaneously and to aid in the review of course materials by giving the preceptor a chance to elaborate on topics that people do not remember well.

Design & Implementation

The design and implementation of "Who Wants to Be an A+ Student" is broken up into four interconnected sections. These sections are the basic HTML, a layer of PHP immediately beneath the HTML, a MySQL database accessed from the PHP, and some JavaScript that enhances certain aspects of the HTML.

HTML

The HTML portion of the application mainly serves the purpose of providing the basic framework of each page. It provides the basic structure of the individual pages and the pieces of static text that appear, such as the paragraph of instructions on the start page for students. Pages like that first student page are almost entirely static HTML. Other pages would end up almost entirely blank if there were nothing but HTML. Although the HTML on its own would produce a very disappointing application, its role in acting as the framework for making each page viewable by browsers is important.

PHP

The PHP code acts as the workhorse of the application, passing some variables from page to page, interacting with the MySQL database, and filling in the HTML framework with the content for each instance of pages such as the individual question pages. The ability to pass variables from page to page, especially in a hidden form, is the first reason PHP is useful. This allows each preceptor page to store the ID number of the current instance of the game and each student page to secretly remember the ID number of the player to whom it corresponds. Also, because the MySQL database cannot be accessed once the page has been loaded, the answer that each player selects must be sent to the next page and retrieved by the PHP code so that it can subsequently update the players' scores and statistics.

A second useful feature of PHP is its ability to access MySQL databases. Various pieces of data about the game and players are stored in a MySQL database, so the capability of PHP to access that data is crucial. The previous example mentioned the fact that PHP had to retrieve the players' answer choices on the page following each question, a need that is driven by the fact that the information must be stored in the MySQL database. In addition to updating player information, this ability allows the PHP to retrieve game information, specifically things like the question text and the next question to be loaded onto the students' pages.

The third aspect of PHP is what allows data retrieved from MySQL to be displayed on the page. PHP can edit the HTML that is sent back from the server to the client, and it uses this to fill out the HTML framework with the information retrieved from the previous page and from the MySQL database. These three features, the abilities to retrieve variables sent from a previous page, to interact with the MySQL database, and to fill out the HTML framework for the displayable page makes PHP a very useful language to use in the development of this application.

MySQL

The MySQL database is the third portion of the application, and aids in two aspects of its design. First, it allows easy storage of things like individual questions and statistics of past games. Second, it connects the preceptor and students by storing data about the current state of the game in a place that both types of pages can access.

The permanence of the MySQL database is useful for storing information that remains the same from game to game. There are multiple ways to store category and question information on the server, but most of them are neither simple nor efficient. One of those options is to hard code unique pages for each question. With multiple categories that have 10 or

more questions each, this method would be tedious and not conducive to adding new categories or questions. Another option would be to store information about the questions in a comma-separated value file, which could be read in each time the application needed information for a new page. In this case, it would be easier to deal with new questions, but it would likely be fairly inefficient if the number of questions in the database grew too large. However, with a database like MySQL, none of these issues are relevant. MySQL allows easy insertion of new entries and simple access of existing entries.

The other useful feature of MySQL is its ability to link preceptor and student sides of the application. Because all of the pages can access the same database, it is simple to store information about the current state of each game. As long as the PHP retrieves the correct game or player ID number on each page, a MySQL call can be made to find out what question comes next for the students. Given the design of the rest of the application, using HTML, PHP, and JavaScript, MySQL is an appropriate choice for achieving the goals of storing permanent information and communicating between student and preceptor clients.

The MySQL database for "Who Wants to Be an A+ Student" is set up with several tables, many of which serve both of MySQL's aforementioned purposes. The first two tables, however, are included for only the first rationale of MySQL, the permanence of information. The first of these tables is one that represents the categories of questions. The only fields that it contains are one for the index of the category and a second for the name of the category. The second related table holds information about individual questions. This question table has fields for the text of the question, the text of the answer choices, and the number of seconds on the question timer, all of which will be displayed in some way to the students and preceptors. The fields that relate

more to the underlying mechanics of the system hold the index of the correct answer, the index of this question's category, and the ID number of this question.

The other three tables are mainly focused on the communication between preceptor and student pages, but also serve as an enduring record of each game. One of these tables holds the general information on the current game, namely its ID number, the number of categories used, and the number of questions in each category. This also has two more fields, storing the indices of the current category and question, which is updated by the preceptor's pages and retrieved by the students' pages. In order to give the preceptor a chance to explain questions that a majority of people got incorrect and to prevent a student from rushing ahead, these two values have to be stored in the database rather than in each individual student's page.

Another of these three tables holds information about each player. Each player is associated with a display name, an ID number, a score, and the ID number of the game in which the player is participating. This table also stores the index of the player's previous answer, both so the preceptor's page can tally the number of players giving each answer and so that the player cannot somehow answer the same question right multiple times. This is accomplished by having the preceptor's page reset that field to -1 at the end of each question and ensuring that a player's score will only change if the field is less than 0. The final table in the MySQL database stores statistics about each player's answers in each category. In order to make the number of categories easily changeable, there is an entry in the table for each category for each player. Each entry stores the ID of each player, the index of that category, and the ID of the current game. In addition, it stores the player's confidence level in the category, the number of questions answered, and the number of questions answered correctly. This set of tables is

effective in long-term storage of information and in communication of game information between student and preceptor pages.

JavaScript

The JavaScript in "Who Wants to Be an A+ Student" is very limited and is not essential to the general structure of the game, but it is useful for improving its usability. The main thing that the JavaScript does is to implement various timers on the pages. The first place this is important is when the preceptor needs to see real-time list of what players are in the game and whether they are at the waiting screen yet. Although it does not have an explicitly displayed timer, the preceptor's start page needs to refresh every few seconds to achieve this effect. This is achieved fairly easily by setting a timeout that refreshes the page. The second use for timers is the obvious application of the countdown timers on each question page. For this, a one second timeout updates the page's HTML and requests the next page once the timer has reached zero. While these features are not essential to the game, they do help it to run more smoothly, making the JavaScript a nice finishing touch to the design of "Who Want to Be an A+ Student."

Evaluation

The "Who Wants to Be an A+ Student" application, as a piece of software with which multiple users will be interacting and that is designed to improve review, has an ideal method of evaluation. Such a method, if done correctly, would be able to determine whether having a review session containing the game made any difference in the final exam grades for real COS 217 students. Unfortunately, such an evaluation was not possible, so alternative methods of evaluation were employed.

The ideal method of evaluation for a review game such as "Who Wants to Be an A+ Student" would involve a large test group, to be divided into test and control group, all made up of students currently in COS 217. All of the students participating in the evaluation would be randomly sorted into one of three groups, the test group and two control groups. The test group would prepare for the final exam by attending a review session that used "Who Wants to Be an A+ Student". The first control group would prepare for the exam with the help of a traditional review session. Finally, the second control group would prepare for the exam without going to a review session. This would be an ideal setup for a couple of reasons. First, the random distribution of students into each group should even out the various levels of familiarity with the course material. Second, if the sample size is large enough, the results of the evaluation should be generally applicable to the population at large. Finally, this setup allows for comparison not only between having the game and not having it, but also between having review in general and not having review. For the sake of completeness, it would be useful to have a group with no organized review sessions to see if review sessions are helpful at all.

Unfortunately, there are about as many things wrong with the ideal solution for this situation as there are things that are right. First, to test the application on real COS 217 students, a group would have to use it as a review tool and then take the exam so that their scores could be compared to those who had not used it. However, the review session and exam are not events that happen during the course of this project under normal circumstances, so getting a large group of students who would be taking the exam before the end of this project would be nearly impossible. Also, in the interest of all students, it is better to have, at minimum, the amount of review that is already happening in the course. Because of this, the second control group, which would have received no formal review session, is not practical. The final problem is the fact that

the stakes for the test groups are important. Most students at Princeton are concerned with grades a fair amount, so it would be difficult to justify risking a reduction in those grades because they did not attend an official review session or get to use a review tool that is potentially more useful than the typical session. Overall, a proper test of "Who Wants to Be an A+ Student" is an unreasonable evaluation method.

Despite the lack of the correct type of testing for this application, it is possible for the programmer and others to evaluate by inspecting it closely. There are two types of methods for this, the heuristic evaluation, which evaluates the whole system's usability based on certain criteria, and the cognitive walkthrough, which goes step-by-step through the path that the user will take and determines how easy it is to learn (Preece, Rogers, & Sharp, 2002). Both of these methods are reasonable for evaluating the application, so the results in this report will be based on them, with evaluations of both the player pages and the preceptor pages.

Results

Heuristic Evaluation

The heuristic evaluation, aimed at an overall picture of the application's usability, is summed up in a few questions. First, is the information that the users would like to know accessible to them? Second, is that information presented in an understandable way? Finally, how easy is it to make a mistake or cause an error, and can the user get out of those mistakes?

For the student, most of the information that is important is always displayed for him in the corner of the screen[4]. For the entire game, the name and player's running score stay in the top left corner of the screen. The questions and answers appear in the center of each questions page, making it obvious that the current section involves them. After the student has answered

[4] This information was cropped out of the screenshots in the appendix to allow the text in them to be read.

each question, the answer is also displayed, making most of the information necessary available. The one piece of information that might be useful for the student is the number of points being wagered on the current question, so he knows before answering what is at stake. As for the understandability of the presentation, most of it should be fairly straightforward, although things like current score could be made to stand out more by being put in bold. When the correct answer is displayed, it is fairly well presented when the player gets it correct, but when the player gets it incorrect, the correct answer is not always intuitive to recognize. Unfortunately, there does not seem to be an easy and intuitive way to indicate the correct answer when the page also must display the player's incorrect answer. Finally, it should be nearly impossible to make a mistake as a student if the student does not try to use the back button on the browser, and if he does, then he will probably end up with fewer points than if he had followed the rules. This is an acceptable solution to discourage cheating. Overall, the student side of the application passes these three heuristics fairly well, with mainly presentation issues that should be dealt with.

The preceptor's side of the application passes these heuristics better than the student's. The information that the preceptor would want is displayed when it is logical to display it. A list of players is displayed at the beginning, along with an indicator of readiness. During the questions, a full list of players is not necessary, but the correct answer is indicated as well as how many people answered each question to display whether to elaborate on that particular question. Finally, the results page displays all of the players, so the preceptor can see how well people did compared to their confidence level from the beginning of the game. Also, for the presentation of information, significant numbers, like scores and percentages are bold to stand out, as is the correct answer for each question, which is a good way to indicate what is important. Finally, like the student side, the preceptor side of the application is designed to be linear, so as long as the

preceptor navigates only through the buttons given on the pages, then there should not be unexpected errors. For these reasons, the preceptor's side of "Who Wants to Be an A+ Student" does well on the heuristic evaluation.

Cognitive Walkthrough

The cognitive walkthrough of both sides of the application is fairly simple. Because "Who Wants to Be an A+ Student" is a game with a fairly linear structure, there is only one path that each person can take from one page to another. In the cases where there are several options, the PHP on the server or the JavaScript on the page will deal with it and load the correct choice of next page. All of the buttons on the student side are central to the page that they are related to, and they are labeled so that it is fairly obvious when the student should press them. Because the extent of what the player has to do is to answer the questions and press buttons that say ready, the player should be able to learn the interface almost instantly. Also, the instructions on the first page explain the rules, but also the fact that if the player is not ready in time for the next question to start, he will miss part of it. Overall, the walkthrough of the student side shows that it should be easy enough to learn that the students can get straight to focusing on the main point of the game, the review questions.

Similarly, the preceptor's walkthrough turns out to be fairly simple. The preceptor's role in the game is larger than the individual students' roles, so it is slightly harder to know when to start the first question or whether to explain a certain question. However, other than giving a threshold of correct to incorrect ratio under which the page suggests explaining the question, there is not much the application itself can do to help the preceptor to decide. A walkthrough of this side of the application reveals that the preceptor should have a fairly easy time learning how to use the preceptor interface, although certain cues that do not involve specific actions in the

browser might be more difficult to figure out because it requires gauging the knowledge of the participants of the review session.

Conclusion

"Who Wants to Be an A+ Student" is an application that attempts to engage all of those attending the review session and effectively allow them to review. After the evaluation of the game, it seems that it is not a perfect tool yet for review, but that it performs fairly well when faced with the heuristic evaluation and cognitive walkthrough. It is also able to engage all players by forcing them to have some risk for each category and by introducing competition for the top 5 spots into the game. The competition could become even fiercer if the top 5 players received a prize of some sort, even if it ends up being something small. The effective review requirement is fulfilled as well, by giving the preceptor the chance to elaborate on everything that was incorrect and by incorporating active learning theory, especially in fulfillment of the first requirement. In addition, despite being discarded as a non-important requirement, this application does, in fact, fulfill the goal of being able to include students who arrive late to the review session. Because of the design of the system, adding a new student would just start that student at 1000 points, where everyone else started, but without the opportunity of gaining points from the questions he missed. The ability to fulfill the original two requirements as well as a third that was discarded early on is a definite advantage of this application.

Although "Who Wants to Be an A+ Student" is a working review game, there are still many features that could be added to enhance it. First, it would be useful to have many categories from which the preceptor could select a different set each game. This way, the game could test more categories without increasing the length of a single game. Another option would

be to add a leader board that displayed after the end of each category. This would make the competitive nature of the game greater as well as allowing players to see how they are doing in relation to others in the midst of the game so that they could use their competitive spirit to focus themselves more for the future categories. A third new feature could be the use of Ajax, which would consolidate the number of pages needed by removing waiting pages and by being able to change the question results screen to attach itself to the question page once the question is answered or the timer has run out. One final future idea would be to allow students to alter their wager slightly before seeing the question. This way, if a player is doing better or worse at a certain category than he originally expected, he could alter the number of points he would lose for the rest of that category. All of these future features have the potential to improve "Who Wants to Be an A+ Student", but as it stands now, this game should work as an engaging and effective review tool for Princeton University's Computer Science 217: Introduction to Programming Systems.

Works Cited

Bonwell, C. C., & Eison, J. A. (1991). *Active Learning: Creating Excitement in the Classroom. ERIC Digest.* Washington, DC: ERIC Clearinghouse on Higher Education.

Carnevale, D. (2005, June 24). Run a Class Like a Game Show: 'Clickers' Keep Students Involved. *The Chronicle of Higher Education* , p. B3.

Preece, J., Rogers, Y., & Sharp, H. (2002). *Interaction design: Beyond human-computer interaction.* New York, NY: John Wiley & Sons, Inc.

Sarason, Y., & Banbury, C. (2004). Active Learning Facilitated by Using a Game-Show Format or Who Doesn't Want to be a Millionaire? *Journal of Management Education* , 509-518.

This paper is my own work in accordance with University Regulations.

Griffin Telljohann

Appendix

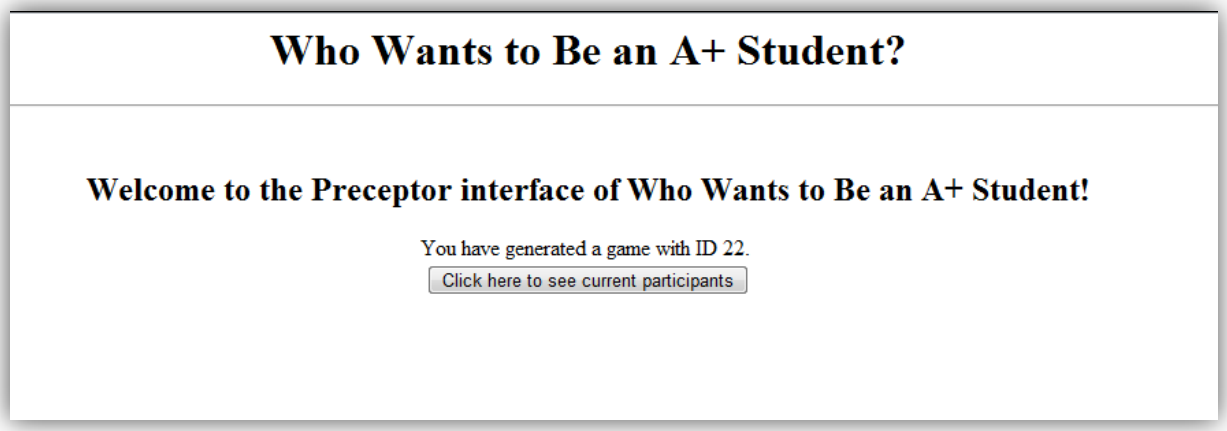

Figure 1: Preceptor - Starting Screen

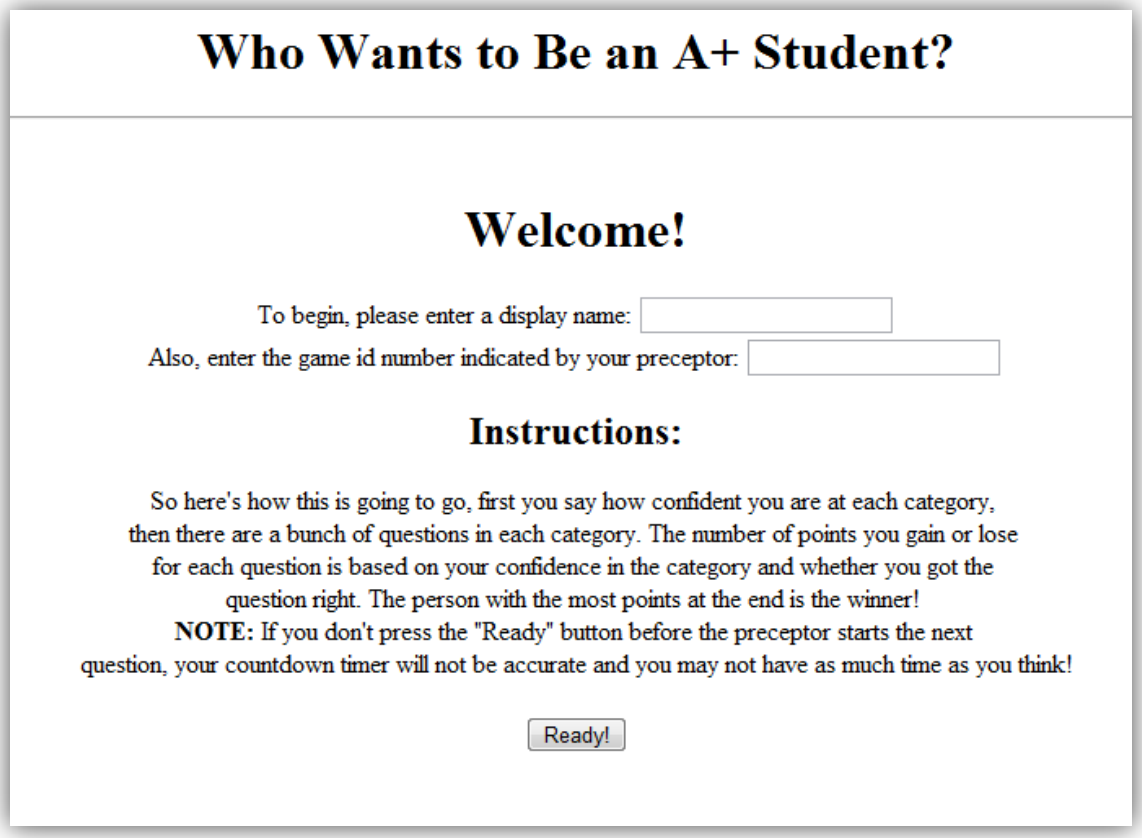

Figure 2: Student - Welcome Screen

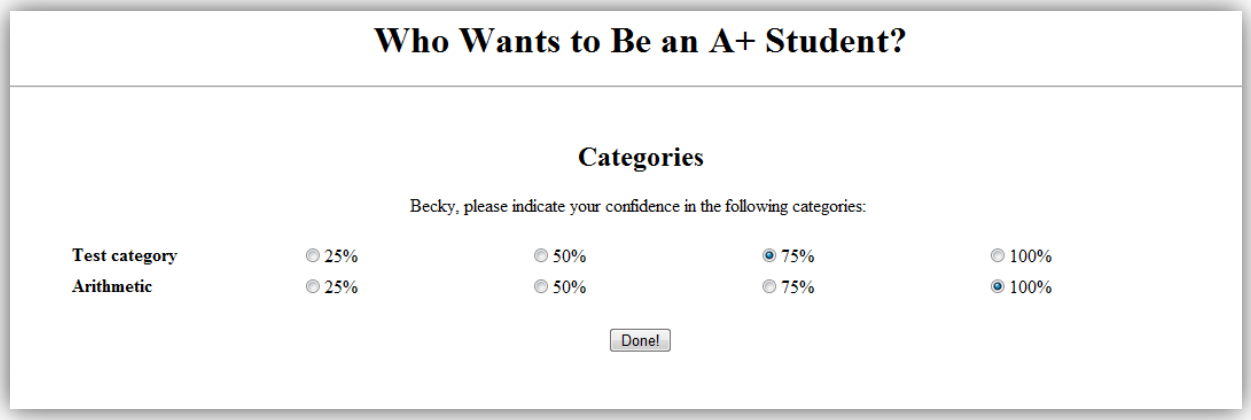

Figure 3: Student - Confidence Screen

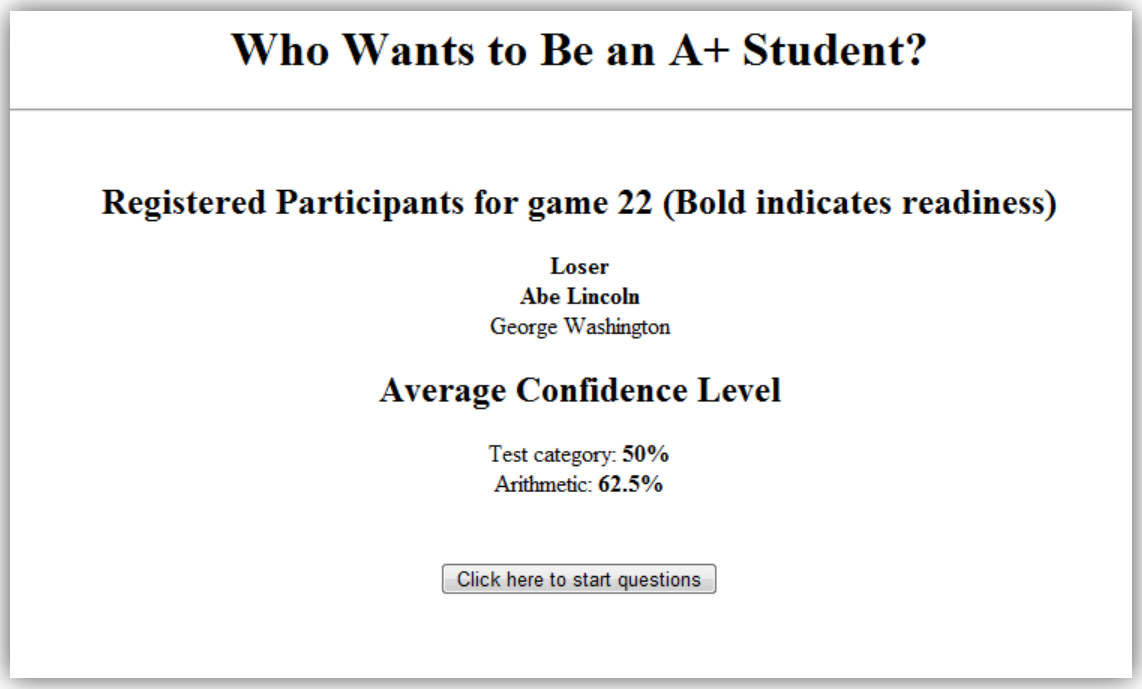

Figure 4: Preceptor - Roster

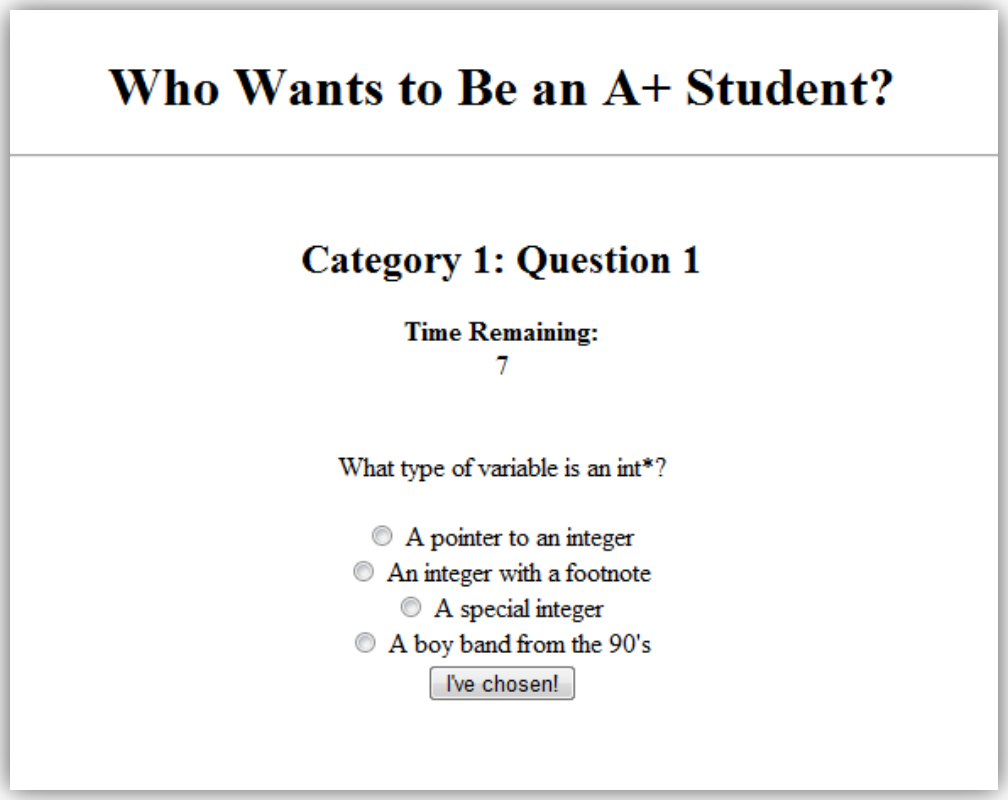

Figure 5: Student – Question

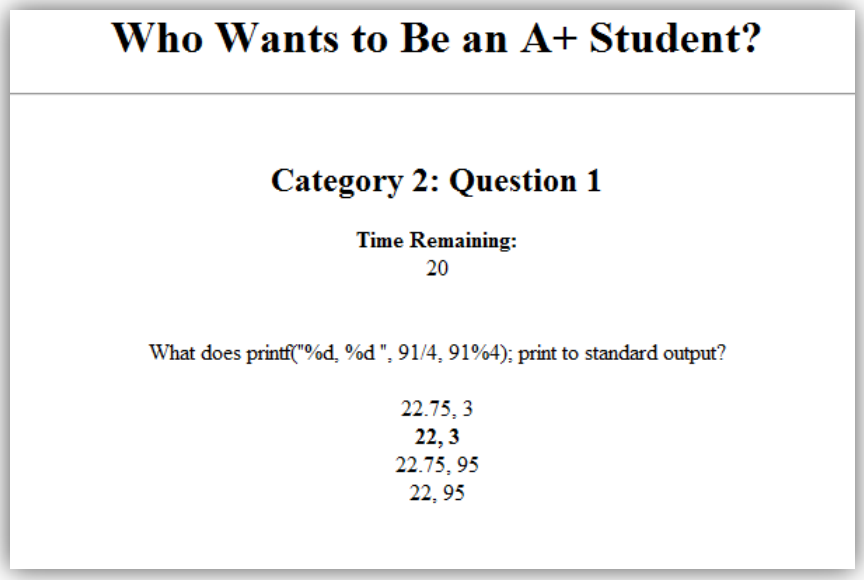

Figure 6: Preceptor - Question

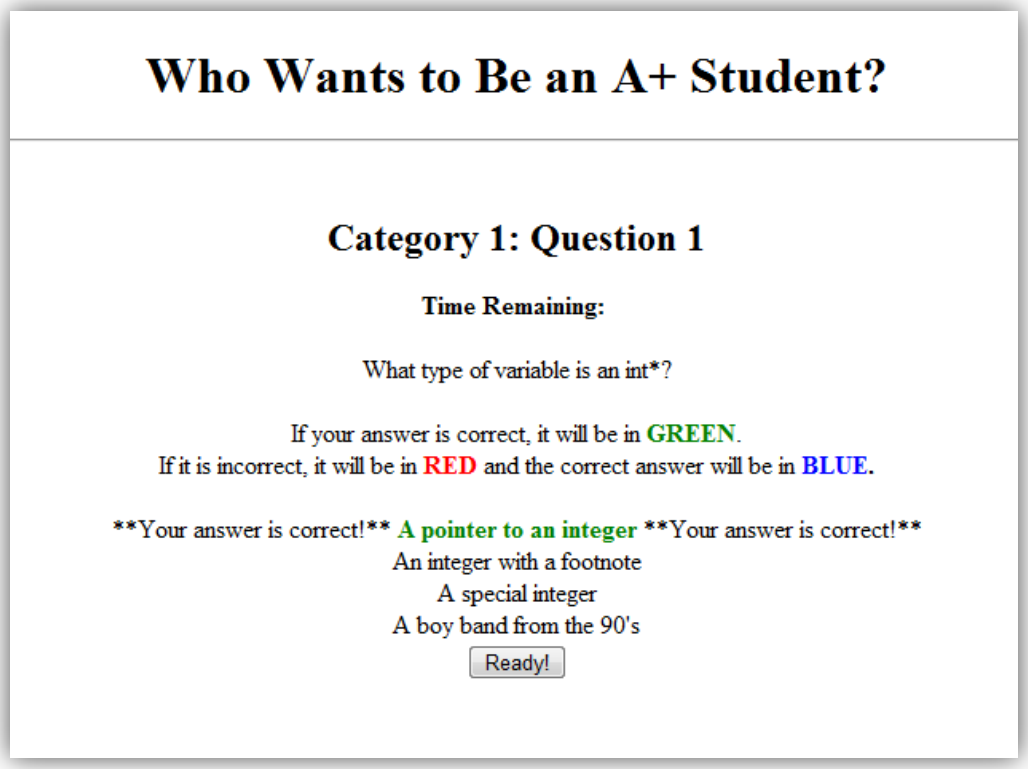

Figure 7: Student - Correct Answer

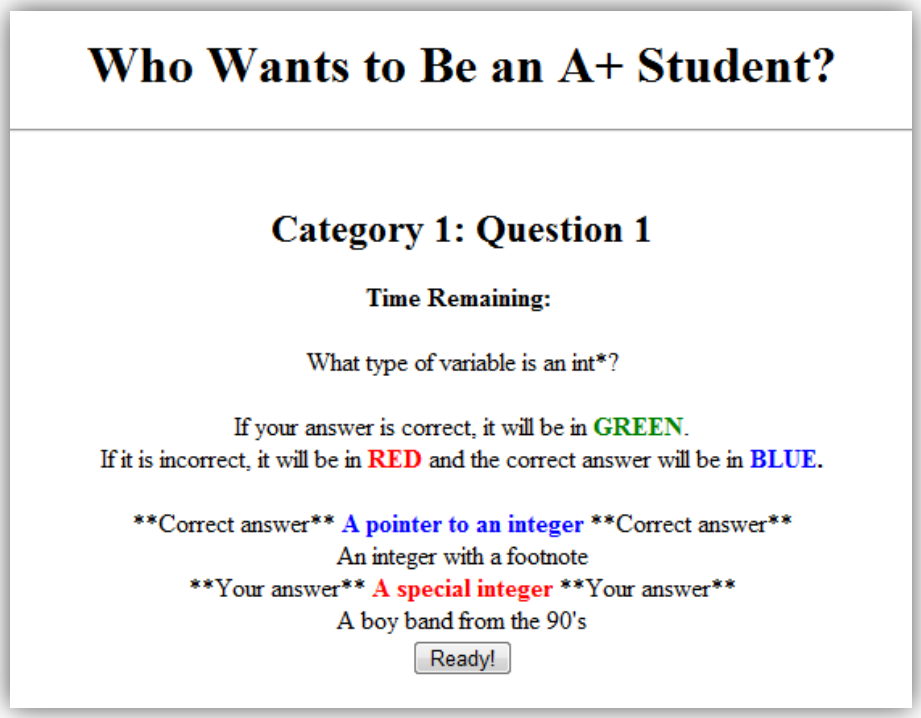

Figure 8: Student - Incorrect Answer

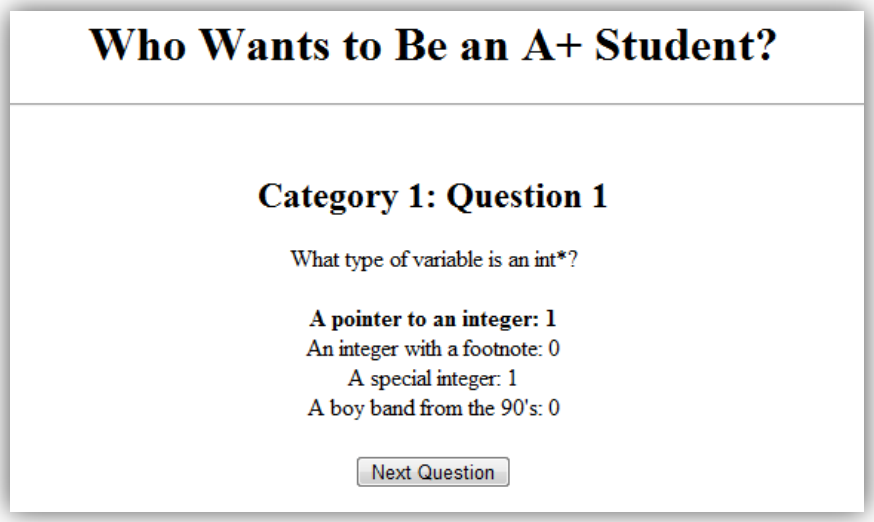

Figure 9: Preceptor - Answer Tally

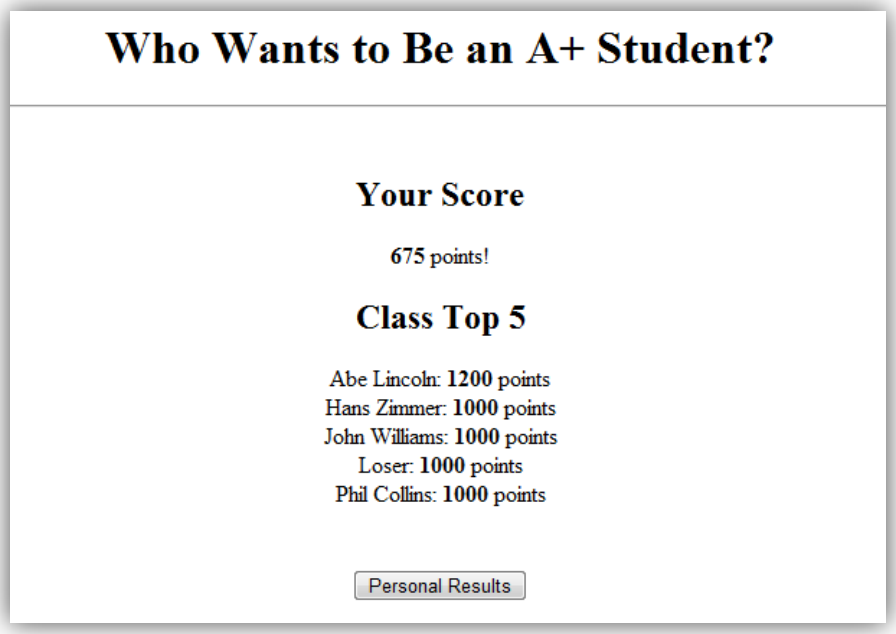

Figure 10: Student - Class Results

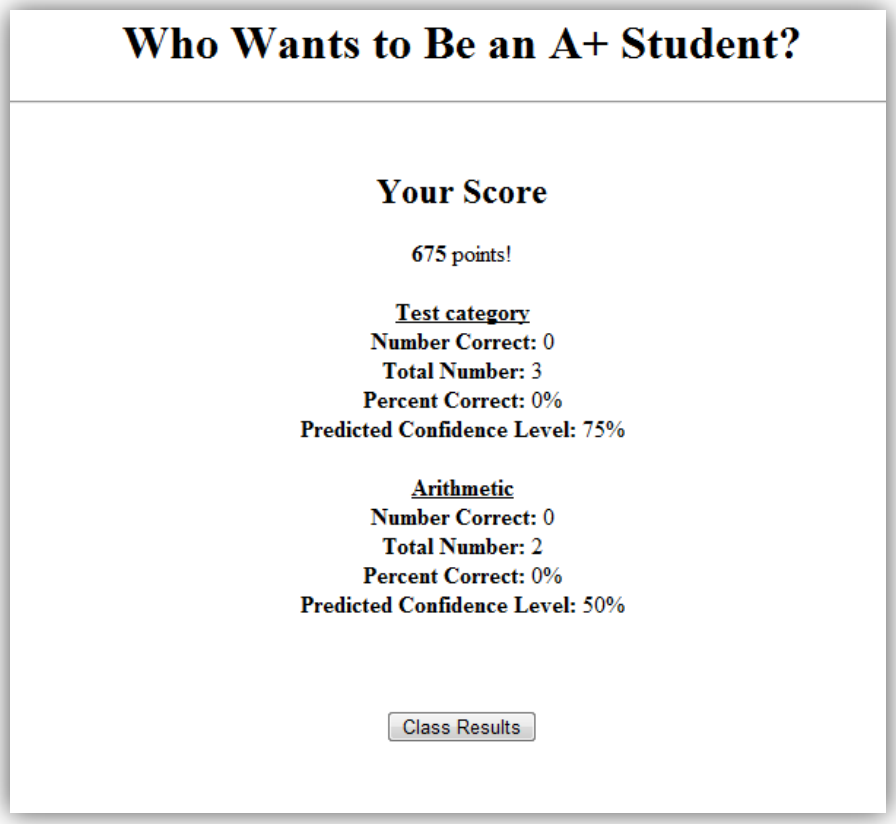

Figure 11: Student - Personal Results

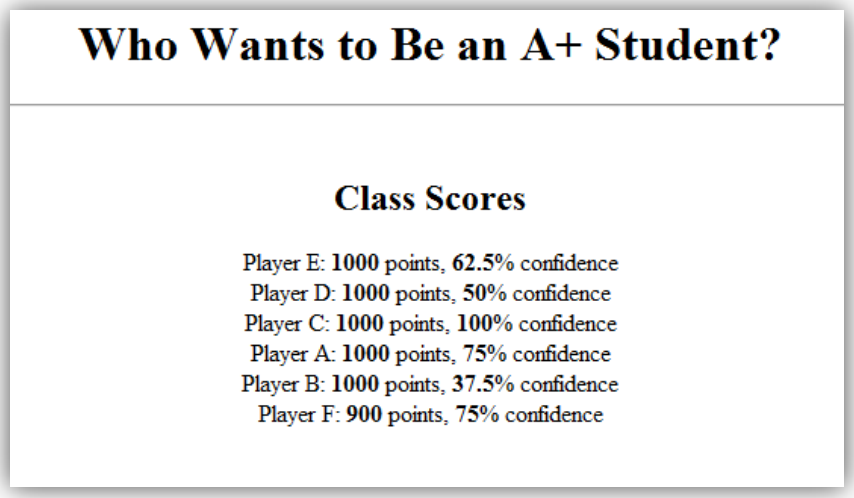

Figure 12: Preceptor - Class Results