

インターネット用フラッシュ・クイズ・プログラムの概要とその開発

A Flash Quiz Program for the World Wide Web:
Its Features and Its Development

Charles I. Kelly *

チャールズ ケリー

Abstract: This article is about the Flash quiz machine used on the Activities for ESL Students <http://a4esl.org> website and the Interesting Things for ESL Students <http://www.manythings.org> website. It covers the development of the quiz machine and some of its features.

1. Introduction

In January 2001 I wrote a quiz program using Flash to be put on the *Interesting Things for ESL Students* website in a section called *Flash Quizzes for ESL Students* <www.manythings.org/fq>. This same program was adapted and also installed on the Internet TESL Journal's *Activities for ESL Students* website <a4esl.org/q/f>. It was further adapted to use with *Quizzes Based on VOA Special English Programs* <www.manythings.org/voa>. This article is about the development of the program and the program's features.

2. Background

Since the early 1980s I have written various quiz programs on various computer platforms. With the advent of the World Wide Web, I have focused on writing quiz programs that can be put on-line and used on any computer platform.

In 1996, I developed an HTML-only quiz template that allowed the answers to be hidden within the page, thus eliminating the need to use a CGI interface for quizzes designed for self study. Quizzes using this template can be found at www.aitech.ac.jp/~iteslj/quizzes/. This quiz format has become quite popular and

teachers are contributing quizzes of this type to the Internet TESL Journal's website as well as using this template on other websites. The advantages of this type of quiz include the fact that they are very fast-downloading due to the small file size and the fact that they are backward compatible to Netscape 2.0 which means they can be used by most, if not all users of the Internet.

In 1998, I developed several quiz templates using JavaScript. These can be found at www.aitech.ac.jp/~iteslj/quizzes/js/. With these quizzes:

- the score is calculated after each question is answered giving a more game-like feeling.
- immediate feedback is given when a student misses a question so students not only get a score, but can learn at the same time.
- questions that are answered incorrectly appear again giving the student another chance, thus a chance to learn the material.

The order in which questions appear and also the order in which the multiple-choices appear are chosen randomly. This allows students to get more benefit from doing the same quiz a second or third time, since each time a quiz is taken, it is different.

In 2000, I developed another JavaScript quiz template that generates quizzes based on bilingual word lists. These quizzes can be found at www.aitech.ac.jp/~iteslj/v/. The program randomly selects a word from one language to use as the question,

* 愛知工業大学 基礎教育センター (豊田市)

puts its equivalent in the other language into one of the five multiple-choice positions, then randomly selects other words as distracters. Since all questions are produced randomly, a student can study the same word list over and over again getting a new quiz each time.

With this as a background, in January 2001 I developed a quiz template using Macromedia's Shockwave Flash.

3. The Development Process

In this section, I will outline the steps I went through to develop the *Kelly's Flash Quiz Machine*.

3.1. I Surveyed the Web to See What Others Were Doing with Flash

Before doing any programming I spent a few weeks visiting as many websites as I could find that used Flash since I wanted to analyze what the capabilities of Flash were. I wanted to discover what worked well and what did not work well.

3.2. I Decided What I Wanted to Do

I wanted to make something that:

- had a small file size, thus was fast to download.
- was backward compatible, thus usable by as many people as possible.
- had an easy-to-use template so that producing future quizzes could be as fast and easy as possible.
- had as much of the program as possible cacheable on the visitor's computer, so when a visitor does a second quiz, he or she is only downloading the quiz data and not the whole program again.
- included the possibility of including specific feedback for missed questions without requiring that feedback for cases where it was not needed.
- included the possibility of different types of questions (i.e. multiple-choice, fill-in-the-blank and true or false) being used in the same quiz.

3.3. I Decided What I Wanted to Avoid Doing

By viewing many other websites that used Flash I was able to compile a list of things I wished to avoid doing.

- Avoid time-wasting cute introductions. Since Flash makes it easy to do animations, many sites were doing this.

- Avoid a large file size.
- Avoid hard-to-read color combinations.
- Avoid hard-to-read fonts.
- Avoid requiring excessive mouse-clicking.
- Avoid a cluttered-looking screen.
- Avoid using pop-up windows to display the Flash quiz.
- Avoid delays by cute animation between questions.
- Avoid using irritating sounds.

3.4. I Wrote the Quiz Program

Since I had had experience in programming other quiz programs using other computer languages, writing the basic program was more a matter of becoming familiar with Flash, learning the ActionScript language and developing a concise method to use to include the quiz data.

3.5. I Fine-tuned the Quiz Program

After writing the basic program, the process of fine-tuning began. This primarily involved experimenting with ways to decrease the file size to promote faster downloading and seeking feedback from those who helped me test the program.

3.6. I Made Two Adaptations of the Full Program

Within a few days of posting the first set of quizzes at <http://www.manythings.org/fq/>, various teachers from around the world had sent me email messages wondering about the possibility of contributing quizzes. For this reason, I produced a quiz machine for the Internet TESL Journal's website, so other teachers could also use this quiz machine to produce quizzes to be used by those studying English as a second language. At the same time, I further adapted the program for the *Quizzes Based on VOA's Special English Programs*.

4. The Various Versions on the Quiz Machine

4.1. Flash Quizzes for ESL Students

<http://www.manythings.org/fq/>

This is the full version of the quiz machine and it has the following features.

- The quiz machine is backward compatible to Flash 4.
- The file size for the quiz machine is only 40 Kb.
- The file size for the quiz data is small.
 - Most quizzes require between 1 Kb and 2 Kb. Even the longest quiz requires only 5 Kb.
- The quiz machine may include any or all of the following types of questions.
 - Multiple-choice.
 - Fill-in-the-blank.
 - True or false.
 - Fill-in-the-blank followed by a multiple-choice question if answered incorrectly.
 - Multiple-choice questions with all choices being the same for every question.
 - Editing.
 - Questions using sound. The question is a spoken instead of written.
 - Multiple-choice.
 - Fill-in-the-blanks.
 - Dictation.
- For multiple-choice questions, the quiz taker has the option to either click the choice or to use the number keys on the computer keyboard.
- The quiz taker has the option to end the quiz at any time.
- The quiz taker has the option to restart the quiz at any point.
- At the end of the quiz, the quiz taker has the option to retry all questions that were answered incorrectly.
- The quiz taker has the option to keep a record by printing the final results screen.
- The quiz author has the option to randomize the question order and the order that the choices appear in a multiple-choice question.
- The quiz author may include a hint for any question. If included, a "hint" button appears on the screen. To give the quiz a game-like feeling, students lose points if they choose to look at the hint, thus students can get a higher score if they do not need to look at the hint.
- The quiz author has the choice to include one feedback statement to appear explaining why the correct choice is correct and/or may include feedback statements for each wrong choice explaining why the wrong choices are wrong.
- Two types of scoring are shown.
 - Standard Quiz Scoring
 - Game-like Scoring
- The quiz machine offers a fun, game-like feeling.

- There are sounds. These are short and not too loud.
- There is a game-like scoring with some questions having more value than others.
- There is a timer.
 - Not only is there a time limit for answering the question, but the potential points that can be earned by correctly answering the question drops as time goes by.
 - More time and points are given for more difficult questions. Fill-in questions are given more time than multiple-choice questions, multiple-choice questions with three choices are given more time than questions with two choices, and so on.

4.2. Flash Quizzes for English Study

<http://a4esl.org/q/f/>

This is the adaptation made for the *Activities for ESL Students* website. This website allows other teachers to contribute quizzes.

These are the reasons for the adaptation.

- I wanted something with an even smaller file size (26 Kb) since there are a large number of visitors to this site and we wanted to keep the load on our server to a minimum.
- I wanted something that would make it easy for other teachers to contribute quiz data.
- I wanted to eliminate some of the options so that proofreading the quiz data before uploading would be easier and less time consuming.

This conversion included the following changes.

- The game-like timer was eliminated.
- Instead of including a font in the quiz machine, this quiz machine uses the default font of the visitor's computer.
- The option to include a hint was eliminated.
- The spacing was changed to allow room for longer questions and longer choices.
- The cosmetic framing of the quiz was eliminated.

Teachers who would like to contribute a quiz to this project may find more information at http://a4esl.org/q/f/data_writers/.

4.3. Flash Quizzes Based on VOA Special English Programs

<http://www.manythings.org/voa/>

Since this group of quizzes only uses multiple-choice questions, everything other than the multiple-choice question option was cut out, so visitors would not need

to download more than necessary. The file size for this version is 18 Kb. Also, since these quizzes were based on pre-written material that would include the possibility of having longer questions, the layout was changed slightly to allow for longer questions.

5. How the Quiz Data is Formatted

In order to allow easy conversion from other quizzes, I programmed the quiz machine to accept question data in various formats. *Data format #1* is the default since it is the most concise method of formatting the data. To make it easier to convert from the Internet TESL Journal's self-study quiz format, the quiz machine can also handle *Data format #2*. I programmed the quiz machine to also accept *Data format #3*, since this is another popular way to format quiz data and would allow conversion from data submitted by teachers that used this format.

A Few Notes

- Variables begin and end with & and should each be on one line.
- Variables that are not used can just be left out, thus resulting in a smaller file. (Variables do not need to be set to NULL as in some programs.)
- The number following the letters in the variable is the question number.
 - For example,
 - Question 2: q2, ca2, w2,
 - Question 3: q3, ca3, w3,

5.1 Data Format #1

This is the default data format. It is the most concise, thus results in the smallest file size. If the quiz writer does not want the choices to be shuffled, one of the other data formats is required. With this format, the choices must be shuffled.

Example: Question 1 (Multiple-choice)

This is an example of how question one's data might look. Notice that there is no space on either side of the equal sign (=).

- &q1=This is ___ question.&
- &ca1=a&
- &b1=an&
- &c1=[nothing]&
- &w1=The word "question" is a singular countable noun and it begins with a consonant.& (optional)

- &h1=Look at the first letter of "question."& (optional)

What follows is a detailed explanation of the variables.

- The Question
 - q1 = question
- The Choices
 - ca1 = correctAnswer (if it's a Multiple-Choice)
 - b1 = distractor 1
 - c1 = distractor 2 (Optional)
 - d1 = distractor 3 (Optional)
 - e1 = distractor 4 (Optional)
 - f1 = distractor 5 (Optional)
- Help / Hint
 - h1 = Hint (Optional)
- The feedback statement for all choices
 - w1=Why the correct answer is correct (Optional)
- The feedback statement for the correct answer
 - wa1=Not really needed since w1 does the job. However, if included then the output will be w1 + wa1
- The feedback statements for all of the distractors
 - wb1=Why b is wrong (Optional) Output is w1 + wb1.
 - wc1=Why c is wrong (Optional) Output is w1 + wb2.
 - wd1=Why d is wrong (Optional) Output is w1 + wb3.
 - etc....
 - **Note:** Often w1, the reason why the correct answer is correct, is enough feedback for all choices, so it cuts down on the file size, since the "why something is wrong" does not have to be defined. In other words, to keep the file size down, do not include feedback for distractors that do not need further explanation.
 - If distractor "b1" is chosen,
 - w1 is shown if it exists, w1 + bw1 is shown, if bw1 also exists.
 - If distractor "c1" is chosen,
 - w1 is shown if it exists, w1 + cw1 is shown, if cw1 also exists.
 - This means that the reason the correct answer is correct doesn't need to be duplicated in each of "feedback" variable.

Example: Question 2 (Multiple-choice)

This is an example of how another multiple-choice question's data might look. This would be Question 2.

- &q2=What number comes immediately after seven?&

- `&ca2=eight&`
- `&b2=nine&`
- `&wb2=That's too high.&` (Optional)
- `&c2=five&`
- `&wc2=Five comes before seven.&` (Optional)
- `&d2=six&`
- `&wd2=Six comes before seven.&` (Optional)

Example: Question 3 (Fill-in)

This is an example of how a "type-in" question's data might look. This would be question 3. Note that instead of the correct answer being indicated by `ca3`, it is indicated by `fa3`. This tells the program that it will be a fill-in question instead of a multiple-choice question.

- `&q3=The meeting will be held ___ the morning of the fourth.&`
- `&fa3=on&`
- `&w3=When the date is included we say "on the morning of..."&` (Optional)

What follows is a detailed explanation of the variables.

- `fa3` = the correct answer if it's a fill-in-the-blank
- `fb3` is used for an for fill-in (optional)
- `fc3`=alternative correctAnswer for fill-in (optional)
- `fd, fe, ff, ...etc...`
- `w3` is the feedback given if the wrong answer is given.

Example: Question 4 (True or False)

This is an example of how a true or false question's data might look. This would be question 4. Note that instead of the correct answer being indicated by `ca4`, it is indicated by `ta4`. This tells the program that it will be a true or false question instead of a multiple-choice question or a fill-in question. This question type is done differently than multiple-choice since the choices should not be shuffled on the screen.

- `q4=Most dogs are green.`
- `ta4=False`

Example: Question 5 (Type-In, Followed by a Multiple-Choice, if Missed)

This is an example of how this kind of question's data might look. This would be question 5. For this kind of question, include at least one possible answer for a type-in question (`fa5`). Also include at least one distractor for a multiple-choice question (`b5`). If both of

these variables exist, then the program "knows" that this is the kind of question that you want.

- `q5=The meeting will be held ___ eight o'clock.`
- `fa5=at`
- `fb5=before`
- `fc5=after`
- `fd5=around`
- `b5=in`
- `c5=on`

Here is a detailed explanation. The variable "q5" is the question. Since "fa5" exists, the program gives the student a fill-in question. The student may fill-in the words "at," "before," "after" or "around" and get the answer counted as correct. If any other word is typed, then the student will be given a multiple-choice question with the choices being "at," "in" or "on." If the student chooses "at," then it will be counted as correct. Filling-in the correct answer gives a higher game score than getting the correct answer as a multiple-choice question.

5.2 Data Format #2

This is the easiest way to convert data from the self-study quizzes. Everything is the same as *data format #1* except the following.

- The Choices
 - `a1` = choice 1
 - `b1` =choice 2
 - `c1` = choice 3 (Optional)
 - `d1` = choice 4 (Optional)
 - `c1` = choice 5 (Optional)
 - `f1` = choice 6 (Optional)
- The Answer
 - `an1`= The letter for the answer's position (`an1=a` means a is the correct choice.)
- Preventing the Shuffling of the Choices
 - Note: You may prevent shuffling of the choices for all questions with one variable.
 - `sc=no`
 - If the "Shuffle Choices" (`sc`) is "yes" (the default), this can be temporarily ignored for a given question like this:
 - `sc1=no` (Question 1's choices will not be shuffled.)

5.3 Data Format #3

This is used to easily convert data from quizzes that have been written in which the correct choice is

indicated by a special mark. Everything is the same as *data format #1* except the following.

- The Choices
 - a1 = choice 1
 - b1 =choice 2
 - c1 = choice 3 (Optional)
 - d1 = choice 4 (Optional)
 - c1 = choice 5 (Optional)
 - f1 = choice 6 (Optional)
- The Answer
 - The correct answer is indicated by attaching an "at sign" (@) at the end. For example, in the following, the word "swimming" would be the correct answer.
 - swim
 - swims
 - swimming@
 - swam
 - swum
- Preventing the Shuffling of the Choices
 - If the "Shuffle Choices" (sc) is "yes" (the default), this can be temporarily ignored like this:
 - sc1=no (Question 1's choices will not be shuffled.)

Variables Used Only One Time Per Quiz

The following data is the same regardless of which of the above data formats is used.

- title=theTitleOfTheQuiz
 - Example: &title=Preposition Practice&
- writer=theWriter'sName
- date=theDateTheQuizWasWrittenOrSubmitted
- url=theQuizWriter'sWebSite'sURL
- directions=instructionsIfNeeded
- nq=urlOfTheNextQuiz
- sc=no (This means that choices will not be shuffled)
- (Note: Even if the default for the whole quiz is "yes", you can specify this for a specific questions, for example, you if you do not want the choices for question 2 to be shuffled use this:
 - sc2=no
- sq=no (means that the questions will not be shuffled. The user has the option to change this. However, you can set it to "no" so that the quiz is loaded with this setting.

6. Appendix: Screen Shots

From www.manythings.org/fq/

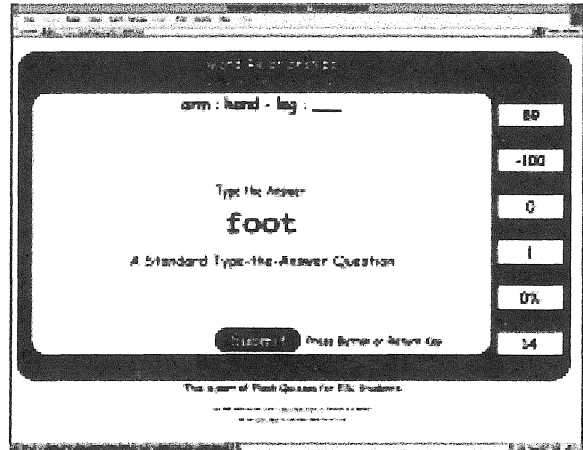


Figure 1: Fill-in Question

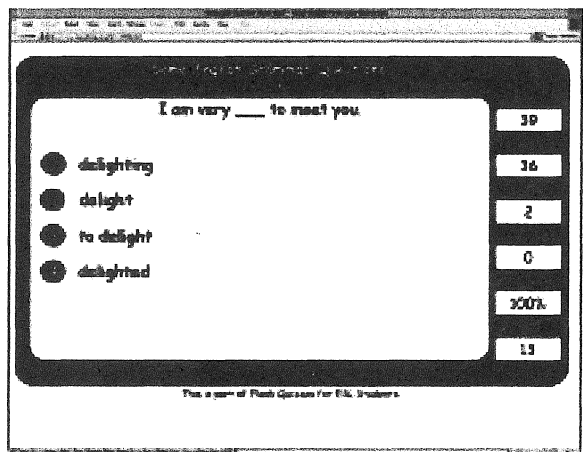


Figure 2: Multiple-choice Question

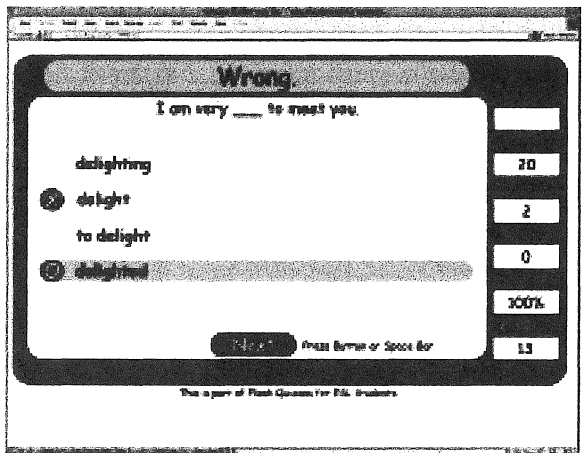


Figure 3: Example of the default feedback when multiple-choice question has been answered incorrectly.

From a4esl.org/q/f/

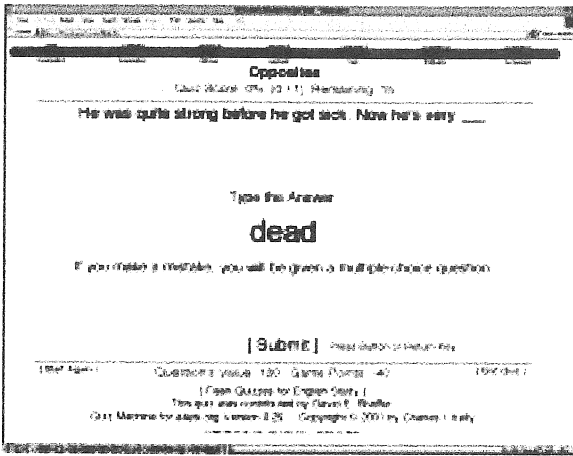


Figure 4: A type-in question.

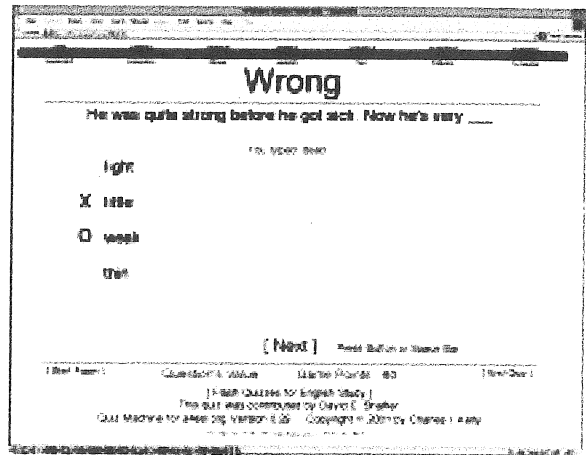


Figure 6: Example of the default feedback given when the multiple-choice question was answered incorrectly.

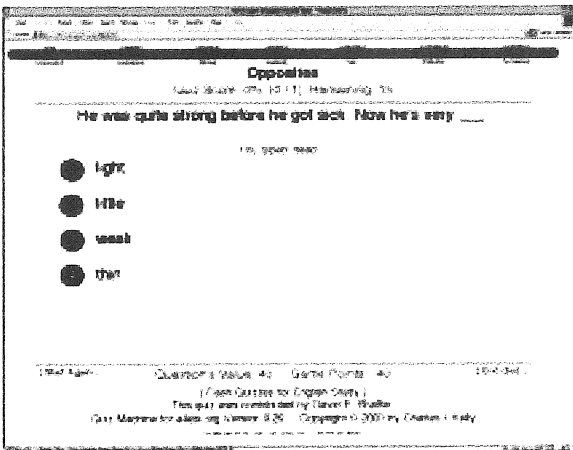


Figure 5: Example of the "second-chance" multiple-choice question given when the type-in answer was incorrect.

From www.manythings.org/voa/

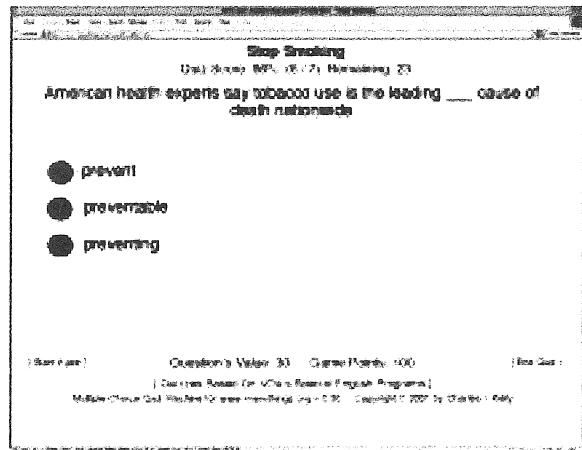


Figure 7: Multiple-choice question

(Received March 19, 2002)