QuizzesDev

Please see the attached file for more info!
Please when editing this page remember set to allow HTML. Thanks.

Status/RoadMap

The result bug is the priority

TikiTeam

UserPageDennisDaniels maintains this page.
George Geller is actively developing quizzes at present.

Quiz Links

QuizRFEs - New features requests

QuizQuestionImport - We need to have acurate help pages for the new functionality.

Trackers

Bugs

Major : Quiz Result Not Shown

"must haves"

- option: all quiz questions answered
- quiz question viewing option... when students are taking the test there must be an option to limit then umber of questions per page otherwise Fast Finger Freddy will see a thirty question test on one html page, ctrl +a, alt + tab, then ctrl +v into a waiting email app and send it to his/her friends for fun or profit... the question viewing must be controlled to 1 to all questions one page
    - perhaps go to a flash output to further hinder copying?
- answer data cannot be loaded into the dom or js... there are some sharp students who know to look at the source of a page to get the answers.

DB could have 100s of questions in a category... the teacher will want, for a summary test, to pull from entire category to generate test... so, option to pull 100 out questions ieither manually or randomly or combination of both out of large db.

- table for quiz categories (column titles sortable)

date created	author	last edited	category X, Y, Z	last used

Q: Why multiple categories?
A:There are a number of questions that can be used for two sometimes three different types of categories... for example looking at a reading standard I might also want to categorize the question by the unit of instruction, topic, book or other.

category	number of questions

Q: Why do you need the number of questions in the category?
A: If I give a test about a certain category and find half my students missed it I need to refocus my energies and come up with more questions to fully assess the needed depth of re-teaching.

- nice to have
- miss a question? create one as a make up
- test data loaded from a URL... this way I can load test questions from wiki pages or attachments from anywhere on the web.
    - Loading test questions from specially tagged wiki pages? I like the idea of loading test questions from wiki pages. This way I'd have a history of the evolution of the questions and have more help from other teachers in developing the questions.

RFEs

- The phpplayers offer exciting options in speeding the creation of test materials... adding data into a rolldown is very interesting....

Creating and importing questions easily is critical and is now in CVS thanks to George Geller...

- the QuickTags enable enormous power for quickly generating standardized tags for quick and easy dev of quizzes...

I've been thinking about how to get feedback connected to the questions...

- srand $positive_feedback
- srand $negative_feedback
- $specific

Each could be tagged like this
-- negative

++ positive
== specific

The negatives and positives would read a wiki page that would pull the comments

controlled testing?

- challenge response: student requests to take a test. An email is sent to the student with an md5sum URL in the message. The student must log into their email, and then click on the URL in the message to verify that they are are in fact the person taking the test...

Regex to the rescue

One of the key ideas in creating quizzes quickly and accurately is to use online texts like ProjectGutenberg for

creating my quizzes. Here's my approach for pulling words that end in ly, mostly adverbs.

1. Choose a text
2. Download the text
3. Run a unique word sort

```
tr ' ' '
' < sleepyhollow.txt | sort | uniq -c >words.txt
```

1. egrep *ly words.txt > newlywords.txt
2. wc newlywords.txt

```
 #sed -e 's/\(Mr\|Mrs\|Dr\)\. /MrDOTHERE /g' -e 's/\. /.\n\n/g'
sleepyhollow.txt |grep $(tr '\n' '|' < newlywords.txt |
sed 's/^/\\(/; s/.$/\\)/; s/|/\\|/g') | sed 's/DOTHERE/./g'
>finished.txt
```

    1. Now this last script should pull only sentences that have the list of words that have ly at the end.
1. Run a yet to be developed script that would take all of the words from the newlywords.txt list and wrap the word in a tag. For example, a sentence in finished.txt would look like this:

I occasionally walked my dog past the gas station.
Post processing would look like
I {=occasionally#$positive_feedback_random} walked my dog past the gas station.
The student would see
I <form> <input type="text" name="textfield"></form> walked my dog past the gas station.
Of course these answers work as well:

- quickly
- slowly
- haphazardly
- kindly
- cruelly

etc
And I would want to add each of those answers as correct! So it would look like this:

I {=quickly#$positive_feedback_random
=slowly#$positive_feedback_random
=haphazardly#$positive_feedback_random
=kindly#$positive_feedback_random
=cruelly#$positive_feedback_random
=occasionally#$positive_feedback_random}
walked my dog past the gas station.

I want to use quizzes to see what students think should go in the answer area.
This question:
I was {/} asleep when the dog started barking.
Would look like
I was <form> <input type="text" name="textfield"></form> asleep when the dog started barking.

Again, there could be lots of answers and I want to measure and use the students right and wrong answers for building more quizzes and better questions quickly.

Some other scripts that might be useful but untested as yet...
egrep -n {*} somewikipage
I imagine parse_tag code is doing this for plugins already...for fun try this one:
egrep -n {*} /var/www/html/tikiwiki/templates/*

Auto check for text areas and text fields for correct capitalization and punctuation. Simple regex checks like

```
^[A-Z]
```

for caps at the beginning of a line.

```
 . [.?!"]  .
```

for the end of a line.

Step by step test creation:
creating the test info
Name: give the quiz a name
Introduction: explain the quiz
Open the quiz: date fields (start date defaults to current time)
Close the quiz: (end (start date defaults to currenct time plus seven days)
Shuffle questions: binary
Shuffle answers: binary
Attempts allowed:one to infinity
Each attempt builds on the last: If a student doesn't finish a test they can continue from where they left off.
Grading method:

- Highest grade
- Average grade
- First grade
- Last grade

After answering, show feedback?: binary
In feedback, show correct answers?: binary
Allow review: allow students to see questions they missed
Maximum grade: point or percentage? This grading aspect of this is tough because it directly begs the question of having a gradebook application connected to the interface. I'm inclined to go with point system and maybe a per centage option if /when there is ever a gradebook in tiki
Question feedback: students provide justification for answers
Question creation: students must provide new questions for each question missed
Force pass: student must pass test before going to next test
Force pass percentage: what per centage of questions must a student get correct before being able to take the next test

importing questions goes here

Selecting questions to be on the test:
Select the category of questions

Select checkbox	Question name description	Type multiple choice, single answer, MC multiple answer	short answer	long answer	binary	matching	Edit	author if author is registered tiki user then this should be a field linked to user page	date created

column head sort
option for

chck box to select all	move to new category	delete

checkbox	delete	move from one category to another	view wiki source page

load questions from category into quiz holding area
select next category
select questions
load questions from category into quiz holding area
repeat

After selecting the questions from each category to go into the quiz we need to determine the order of the questions... randomizing question order should be limited to question category but that's complicated... how do you determine which question has priorioty? Anyway, once you've selected your questions from the list of questions you want to make sure the questions you chose are utlimately the ones you want.

again all tables should be column head sort

chck box to select all	move to new category

Order up and down	Question name	category	author	Type	Grade rolldown, this is the value of the question	Edit	checkbox	remove

All of the questions selected should be tallied so that the test creator knows how many total points are at stake on the test...

Save the test and then view the test in test mode to make sure there are no mistakes...

Suggested tagging system:

Multiple choice: one correct answer radio button

```
Text of question {[category1, category2,image]=answer(feedback)
~option(feedback) ~option(feedback)}
```

Mulitiple choice: more than one answer correct

```
Text of question {[category1, category2, image]=answer(feedback)
~option(feedback) ~option(feedback)}
```

The questions should be wiki based so that histories of questions are easily tracked.

everything in square or curly brackets would be an optional variable.

- support for csv upload to question database
    - rational: enormous number of csv databases with questions already built
        - people know how to use spreadsheets and can build questions quickly
- SCORM or IMS compatiblity
- BAD: the questions are all single option radio buttons
- question categories
- support for images
- matching questions to answers... display a list and match the correct answer to the question
- sharing of question database across multiple installations of Tiki
- short answer responses...
    - voting for more correct and less correct answers in the quiz results
- notification that a quiz question has been submitted
- notification that a quiz has been completed
- check box support to allow for many possible answers
- long test questions
- Plugin SF failed. One or more of the following parameters are missing: groupid, trackerid or itemid.
- instant feedback linked to wrong and right answers
    - comments
- submit new questions like article submission
    - user count of who has submitted most questions to which category
- easy way to edit the questions (if you're the owner of the question)
- reuse options
    - get rid of that annoying rolldown prompting user to reuse a question...test creators need to reuse the options! The options need to be in categories and displayed as tables so that the user can ckbox select the possible answers from a large list AND have the option of adding a new option if they don't find the one they like
- flashcard viewing
    - flashcard viewing in slideshow format for going over questions in the class on a projected screen in the classroom
- Finding misspelled or poorly structured sentences are easy... I'd like to be able to submit a list of sentences. Each sentence would then be numbered and then loaded into a form. The form would would

then have option fields:
- rewrite this sentence correctly
    - text field

This would be reviewed by the teacher or 'farmed out' for AnonymousPeerReview

    - identify and correct the misspelled word
    - text field with the correct answer(s) stored in the db

Competition and standards

List of other products with similar/interesting/related features.
A freshmeat search has a bunch of ideas.

- moodle

Here I would like to see some "editorial" content. How do our features compare to others?

- Since you ask and this is a wiki, I again would suggest that you consider incorporating test validity and item analysis. I found this free source.

CVS Doc section

This is where new features being developed and only in CVS are documented. When the CVS becomes RC/official release, the info in the CVS docs is transferred to update the official docs (FeatureXDoc).

Sample student output that needs to be reused

The target words are: captivated, audacious, complicated, truant, and eminance.

The people at my school are captivated.

- This sentence needs a statement that tells us what the students are captivated by...

When I wrote my essay the title was audacious.

- How is the title audacious?

Even know the words were bold it was a complicated essay.

- This sentence doesn't make sense.

I was trunt ten times in one year to class.

- The target word is misspelled.

When I was in the mountains the eminence was eleven thousand.

- incorrect or uncommon usage of term.

Discussion/participation

A very nice site for testing is found here http://www.visl.hum.sdu.dk/visl/en/edutainment/quizzes/wordformquiz.htm it offers instant feedback... I've got hundreds poorly structured sentences that students have turned in that I would like to convert to questions. Where ideas can be exchanged, debated, etc. Interested people can subscribe to the wiki page and/or to these forums as they would a mailing list.

Rational

quiz control

- teacher control of quizzes and questions is not useful
- students demonstrate their knowledge by asking questions and posing answers

Give students write and read access to quizzes

- reward students for asking questions
- reward students for practicing posing and answering questions
- encourage peer review of materials
- encourage creation of materials

Use students errors to create tests

Finding misspelled or poorly structured sentences are easy... I'd like to be able to submit a list of sentences. Each sentence would then be numbered and then loaded into a form. The form would would then have option fields:

- rewrite this sentence correctly
    - text field
- identify and correct the misspelled word
- text field with the correct answer(s) stored in the db

Teacher control spells misery for students

Imagine if at your job all you did all day long was repeat exactly what your boss told you, copied things out a book, like a photocopier, and worked for little or no recompensation?

Treat students like content creators not content copiers

If you had a job as boring and as hectic as the one described above you'd be angry, depressed and bored all at the same time. What's worse you wouldn't be able to escape...that's what school is like for most students because they are so removed from the learning process.

Open the materials

Let the students create the materials, the quizzes and you'll be surprised at how quickly the students will rise to the challenge. The challenge is between each other not the the ALL mighty sage on the stage.

Instant Feedback

The test taker should be prompted immediately, while in practice mode, to see the rationale for their answer as

well as the number of other students who answered similarly.

- use case

1. test takers opts for instant feedback
2. test taker answers question
3. prompt to see rationale
4. new window pops up with rationale
5. test taker views rationale, number of students who chose same and closes

some prototype text based input

```
When tags are in the middle they indicate test taker must choose /n
or fill in the blank the word /n
that best fills the in the area. /n

This is a fill in the blank {=this is the answer (#feedback [url])
~this is an option # feedback [url]
~this is another option #feedback [url]
} question.
------------------------
The tags are at the end indicating it is a basic multiple choice question.

Which word in this list has a negative connotation?
{~cravat #that's wrong
=lemon
~key
~tape}
---------------------------
Which term has the most negative connotation about people?
{~elderly
~plebian
~pedestrian
=geezer}

Which of these terms has a negative connotation?
{~details
~items
~counts
=minutiae}
```

some prototype interfaces

<form>
When tags are in the middle they indicate test taker must choose or fill in
the blank the word
that best fills the in the area.

<p>This is a fill in the blank
<input type="text" name="textfield">
question.

The tags are at the end indicating it is a basic multiple choice question. Which word in this list has a negative connotation?
<input type="checkbox" name="checkbox" value="checkbox">
cravat #that's wrong
<input type="checkbox" name="checkbox2" value="checkbox">
lemon
<input type="checkbox" name="checkbox3" value="checkbox">
key
<input type="checkbox" name="checkbox4" value="checkbox">
tape

Which term has the most negative connotation about people? Radio button answers
<input type="radio" name="radiobutton" value="radiobutton">
elderly
<input type="radio" name="radiobutton" value="radiobutton">
plebian
<input type="radio" name="radiobutton" value="radiobutton">
pedestrian
<input type="radio" name="radiobutton" value="radiobutton">

geezer Describe a similarity found between the power structures found in Animal Farm and Lord of the Flies
<textarea name="textarea" cols="100" rows="15"></textarea>

</form>

Reviewing and reusing the data from students errors and answers is critical! The idea here is to provide some basis for pattern matching in the data base... there can be a few answers that are correct but worded differently.. This is essentially the problem with tools for testing and teaching a language.. it's incredibly complex...The goal is to use the student's errors as a way of building the database for future questions..

This interface is assuming a short answer interface. Filters for searching are also important..
<form name="form1" method="post" action=""> <table width="93%" border="1"> <tr> <td>sortable</td> <td>sortable</td> <td>sortable</td> <td>sortable</td> <td>sortable</td> <td>sortable</td> <td>sortable</td> <td>sortable</td> <td>sortable</td> </tr> <tr> <td>question categories</td> <td>question text</td> <td>question answer</td> <td>student </td> <td>answer</td> <td>use this answer as an incorrect option?</td> <td>use this answer as a correct option?</td> <td>level of correct</td> <td>feedback</td> </tr> <tr> <td>Shakespeare, CA RS 1.3</td> <td>What were Caesar's last words?</td> <td>Et tu, Brute! Then fall, Caesar.</td> <td>student name</td> <td>Et tu, Brute!</td> <td><input type="radio" name="radiobutton" value="radiobutton"> yes</td> <td><input type="radio" name="radiobutton" value="radiobutton"> yes</td> <td><select name="select2">
<option>correct</option> <option>100</option> <option>75</option> <option>50</option> <option>25</option> <option>0</option> </select></td> <td><input type="text" name="textfield2"></td> </tr> <tr> <td>?</td> <td>?</td> <td>?</td> <td>?</td> <td>?</td> <td>?</td> <td>?</td> <td>?</td> <td>?</td> </tr>

</table> ?

</form>
So at the end of the grading the text file will change from this

```
What were Caesar's last words? {=Et tu, Brute! Then fall, Caesar./ #feedback [URL]}
 to something like
What were Caesar's last words? {=Et tu, Brute! Then fall, Caesar./Et tu, Brute! /n
 #feedback #feedback 2 #feedback 3 [URL]}
```

Again the ultimate goal is to use the students' errors to build the database not only with quizzes but with all homework... there is nothing new under the sun but in education we're constantly hashing the same problems over and over because we have no way of effectively capturing answers and problems and turning them back on to the student. Students who are held closely accountable to their errors and to their questions become better thinkers!

The Quiz Plugin

I think I've got it... using plugins... a student makes a type of error... for example
They did not except my offer. <-----errror

- This is an error or word choice.
- The error was made on an assignment in category X
- The error is of category Y type, word choice.

Using the quicktags I can wrap the error in tags like this:
{QP()}They did not except my offer. {QP}
Then next using another quicktag:
{QP(word choice)}They did not except my offer. {QP}
Now, the QP would know the category of the error and could pick up the category of the page where the error was found. This is supposing that the students are doing their homework assignments in wiki which is not really feasible at this time ...) The word choice category is STILL valid though.
Now... how to turn the tags into a quiz...
Well, word choice questions are by default fill in the blank/ short answer. So, the question would for (word coice) type of errors would be:

> Type a word that corrects this sentence:
> They did not except my offer.
>
> <form> <input type="text" name="textfield"></form>

<form name="form1" method="post" action=""> <table width="93%" border="1"> <tr> <td>sortable</td> <td>sortable</td> <td>sortable</td> <td>sortable</td> <td>sortable</td> <td>sortable</td> <td>sortable</td> <td>sortable</td> <td>sortable</td> </tr> <tr> <td>question categories</td> <td>question text</td> <td>question answer</td> <td>student </td> <td>answer</td> <td>use this answer as an incorrect option?</td> <td>use this answer as a correct option?</td> <td>level of correct</td> <td>feedback</td> </tr> <tr> <td>Shakespeare, CA RS 1.3</td> <td>What were Caesar's last words?</td>

<td>Et tu, Brute! Then fall, Caesar.</td> <td>student name</td> <td>Et tu, Brute!</td> <td><input type="radio" name="radiobutton" value="radiobutton"> yes</td> <td><input type="radio" name="radiobutton" value="radiobutton"> yes</td> <td><select name="select2"> <option>correct</option> <option>100</option> <option>75</option> <option>50</option> <option>25</option> <option>0</option> </select></td> <td><input type="text" name="textfield2"></td> </tr> <tr> <td>Word Choice, CA RS 2.4</td> <td>They did not except my offer.</td> <td>none yet...I don't want to correct every little error!!!!</td> <td>student name (lifted from userID from page?!)</td> <td>accept</td> <td><input type="radio" name="radiobutton" value="radiobutton"> yes</td> <td><input type="radio" name="radiobutton" value="radiobutton"> yes</td> <td><select name="select2"> <option>correct</option> <option>100</option> <option>75</option> <option>50</option> <option>25</option> <option>0</option> </select></td> <td><input type="text" name="textfield2"></td> </tr> <tr> <td> Apostrophe, The Tiki Project </td> <td> Lets start the New Year with New Tools!</td> <td>none yet...I don't want to correct every little error!!!!</td> <td>student name (lifted from userID from page?!)</td> <td>Let's</td> <td><input type="radio" name="radiobutton" value="radiobutton"> yes</td> <td><input type="radio" name="radiobutton" value="radiobutton"> yes</td> <td><select name="select2"> <option>correct</option> <option>100</option> <option>75</option> <option>50</option> <option>25</option> <option>0</option> </select></td> <td><input type="text" name="textfield2"></td> </tr>

</table> ?
</form>

Again the ultimate goal is to use the students' errors to build the database not only with quizzes but with all homework... there is nothing new under the sun but in education we're constantly hashing the same problems over and over because we have no way of effectively capturing answers and problems and turning them back on to the student. Students who are held closely accountable to their errors and to their questions become better thinkers!

These docs become almost unmanageable at this size... the edit link is becoming a major nicety...

```
PHP text parser. The code will reference an array of regular expressions. For example :
/^T/
/!/
/,/
/:/
/;/
/hero/

And search for the string in a separate text document. When the string is found the ENTIRE
sentence where the string is found is 'pulled' from the text document.
The found search string is then 'blanked out'. For example:
The dog was a hero.

Would print:
_he dog was a ___.
```

The found search string is placed directly beneath with an asterisk like so:
_he ___ was a ___.
*T, hero

The searched text can be 400 to 600K in size of ascii.
The output needs to be simple text.