

Copy GMCQ editing UI

BACK TO STARTING POINT

ADAPT AUTHOR

SVEN LAUX

LOG OUT

Editing

Last modified: 1 Apr 2014

Author: Sven

Privacy: org

Save Changes

Preview

Cancel

Graphical Multiple Choice Question

Short component title: ?

Long component title: ?
(displayed on page)

Initial content text: ?

B I U

Number of attempts: ?

Random order: ? false

Question weight: ?

Number of items to select: ?

Buttons

Hide correct answer button label: ? Show correct answer button label: ?

Reset button label: ? Submit button label: ?

Feedback test

Feedback for incorrect answers

Incorrect first answer: ?

B I U

Incorrect non first answer: ?

B I U

Feedback when answer is partly correct

Partially correct first answer: ?

B I U

Partially correct non first answer: ?

B I U

Feedback for correct answer: ?

B I U

Question graphic

Media title: ?

Alternative text: ?

Large graphic asset: ?

Medium size graphic asset: ?

Small graphic asset: ?

Question answers

Item 1

Item 1 - Answer text: ?

B I U

Correct answer: ? false

Feedback

Next

_class: ? class1 class2 class3

We have defined some basic data input UI requirements:

Ability to display a language-independent label other than the variable name
Ability to display a language-adding context sensitive help per data item
Ability to display formatting fields only by default
Ability to display all data entry fields
Ability to group related data entry fields to the editor
Ability to apply basic formatting to relevant data types
Ability to call up a rich-text editor for events
Ability to limit arrays of items to sensible values / what the code can handle via the scheme
Ability to restrict a data input form field based on the data type

Use friendliness of input fields

The data input fields are read in from the component schemes and rendered by the authoring tool. This example shows that the authoring tool uses the variable names from the scheme rather than descriptive labels. Using scheme data alone, also creates an issue in terms of being able to translate the labels into other languages. Firstly, the data of the UI also might be used to add context sensitive help for each data field in order to make the authoring tool as easily usable as possible.

Design actions

- Add language string display labels (Ability to change authoring tool interface language)
- Add context sensitive help files
- Change input fields to better represent input data type (e.g. longer input boxes, buttons with formatting, etc. present)

Separation via accordion control

The separation of content, component properties and extensions is derived from the technical structure of the scheme files. I believe this is not suitable for separation from a usability point of view. I also believe that the accordion which is not covered and that some would be better served with a long scrolling page and maybe an index to the sidebar to help to them.

Design actions

- Add validation action buttons to sidebar
- Change accordion control to long scrollable page
- Initial view to show necessary fields only
- Add index to sidebar in scrolling page on sidebar