A Moodle Plugin for Socratic Questioning

By
Manish Chouhan
Roll No. 113050082

Under the guidance of
Prof. Sridhar Iyer
Introduction

Motivation

- On-line learning and distance education systems are gaining attention now a days, which can be available at any place and at any time.

Goal

- Create ITS Framework for Socratic Questioning as a plugin for moodle through which teacher can teach their course in distance learning environment.
Socratic Questioning

- Socratic questioning is basically a dialogue conversation between two teacher and student.
- The main purpose of this strategy is to fill the students’ mind with questions;
- In this process they find answers on their own through a series of questions.
- This strategy based on the theory of Socrates “learning consists recollection of things that student already knows”.
Objective of Socratic Questioning

- This method helps students to develop critical thinking attitude and inquiring attitude.
- Socratic Questioning helps the students to debug their own theories.
- This method uses questioning about student’s existing beliefs. And lead their beliefs/ideas to a contradiction.
Types of Socratic Questioning

- Classic Socratic Method
- Modern Socratic Method
Classic Socratic Method [1]

- Classic Socratic Method uses questioning to lead on-going conversation towards contradiction of other person’s belief.
- In this method neither questioner nor respondent (other person) knows the correct answer.
- The result of classic Socratic Method is failure to find a satisfactory answer to the primary question.
- If satisfactory answer is found, this method becomes the modern Socratic Method.
- This method is difficult to implement in real conversation because of handling the wide range of possibility of answers is impossible.
Modern Socratic Method [1]

- questioning is used to lead a student’s mind to knowledge through small steps (series of questions).
- In this method teacher ask direct question that have predefines range of answers.
- Prompt student to answer correctly primary question before moving to the next question.
Principles used in Socratic questioning [3]

- Generate different examples to students
- Recall knowledge the student already has
- Determine students belief
- Ask for prediction
- Present counterexamples
Principles used in Socratic questioning [3]

- Testing of hypothesis formed by student.
- Entrap the student and lead him/her to contradiction. When he/she has not identified all the relevant factor.
- Question further to elaborate the concept.
- Help the student in establishing new rules.
- Ask the student to apply new rules.
Moodle Integration

Teacher
- Log-in to moodle
- Select a course
- Can
- Add an activity like:
  - Assignment
  - Surveys
  - wiki
  - Socratic

Student
- Log-in to moodle
- Select a course
- Can
- Do Activity set by Teacher
  - Also Can
  - See the grades, post to forum

Add a resource, see report etc.
Also Can
Moodle Integration
1. User will select Socratic activity on course home page.
2. System will prompt teacher to select one option either add question or edit question.
3. If user will select add question then if previous concept is completed.
Functional Requirement

4. If previous concept is not completed.
5. Teacher can link question with older question
If a student click on activity then
Functional Requirement

If there is an incomplete Concept in the quiz.
If there is no incomplete Concept in the quiz.
## Implementation

<table>
<thead>
<tr>
<th>File Name</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>view.php</td>
<td>Home page of Socratic activity</td>
</tr>
<tr>
<td>mod_firstQuestion.php</td>
<td>For teacher to enter First question of a new concept.</td>
</tr>
<tr>
<td>next_question.php</td>
<td>For teacher to enter a connecting question for previously added question.</td>
</tr>
<tr>
<td>show_list.php</td>
<td>For teacher to connect an option of current question to a previous question from the list of questions.</td>
</tr>
<tr>
<td>edit_list.php</td>
<td>For teacher to select a question for editing.</td>
</tr>
<tr>
<td>edit.php</td>
<td>For teacher to edit the selected question</td>
</tr>
<tr>
<td>attempt.php</td>
<td>For student to attempt the Socratic quiz</td>
</tr>
<tr>
<td>version.php</td>
<td>This file contains the current version number of the module.</td>
</tr>
<tr>
<td>lib.php</td>
<td>Contains functions which are needed to integrate the module with Moodle.</td>
</tr>
<tr>
<td>db/access.php</td>
<td>Use to Define capabilities to restrict access of the user</td>
</tr>
<tr>
<td>db/upgrade.php</td>
<td>Use to upgrade database in moodle</td>
</tr>
<tr>
<td>db/install.xml</td>
<td>Define Database tables in xml format</td>
</tr>
</tbody>
</table>
Component Diagram

view.php → attempt.php

mod_firstQuestion.php ← next_question.php

edit_list.php

edit.php → show_list.php
Process Flow

1. Add socratic activity
2. Fill up socratic form
3. User Type
   - teacher
   - student
4. Status of quiz
   - Incomplete
   - Complete
5. Select action to be performed
6. Selected action
7. Status of quiz
   - complete
   - Incomplete
8. Start a new concept
9. Attempt quiz
10. Finish
Database Structure
Functional logic

- Socratic quiz will be created concept wise.
- For each concept there will be a series of questions which forms a cycle.
- Every option of a question will be connected to some question.
- Correct option of first question will automatically connect to first question of next concept.
Cycle of questions
Status of concept in quiz?

- When add a new question - Add an entry in socratic_parsing table for that question.
- When all option are parsed – delete this entry from socratic_parsing table.
- If there are no entry for a course module {Socratic activity} then there is no incomplete in that Socratic quiz activity.
**Edit Question**

The image shows a screenshot of a webpage with a section titled "data structure." It appears to be a course page with a list of questions related to programming. The questions are numbered and include:

1. What would happen when this program is executed? using namespace std; int main() { int a = 0, cout<< a; }
2. In statement 'cout<< a;' what is cout?
3. Does our program contain this ofstream class?
4. Q4 Where is ofstream class defined?
5. So before using cout should we include this library header file?

These questions are likely meant to be edited or reviewed by the instructor.
Edit Question

In statement `cout<< a;` what is `cout`?

A* Object of `outputstream` class

B* Variable

C Function

D

Correct* A

Save changes | Cancel

There are required fields in this form marked *.
When user clicks on save changes button all entry in sequencing table and parsing Table for this question_id will be deleted.
New entry will be added to the parsing table with field parsed =0.
Question entry in question table will be updated.
Limitation & Future work

- At this time this plugin is limited to teach any subject with the help of multiple choice questions (MCQs) only.
- Teacher can't enter rich text content in question and option fields.
- Use of editor element instead of text area in moodle form. It allows user to enter rich text content.
- GUI may be improved.
References


2. Vikash Kumar, Development of Intelligent Tutoring System Framework: Using Socratic strategy


Thank You