

PROJECT IDEAS

-Anup Gupta
Undergraduate Student
Computer Engineering
NSIT

Examination Simulator

Project Description:

An Examination Simulator creates a simulation of a real time examination. The simulator provides practice in actual test conditions and gives test results instantaneously. The simulator will also provide a detailed analysis of the test.

Supposedly, the test contains 50 questions, out of which 10 are knowledge based, 10 understanding based, 10 skill based, 10 conceptual and 10 application based. The student will also be able to get a detailed analysis based on the type of questions attempted along with other details like correct and wrong answers, percentage obtained, subject wise analysis, efficiency, efficiency depending on the time taken.

Maximum learning takes place by practicing questions. The test can prove very useful both for classroom and home environment. It has utility for both teachers and students. We can have animated, colorful backgrounds for the test screens. The questions can also be displayed in an attractive manner. Thus, it will enable learning with competitiveness and fun.

"This Simulator can also serve the purpose of the Worksheet Generator as proposed on the OLPC Wiki"

Development Stage:

The Examination Simulator can be developed in PHP with MySQL as the database. I feel that the most important part in the entire process would be to create a database with appropriate fields. I have thought over the fields that will constitute the database. We will need to divide the backend in two tables. The first table focuses on the details of the test while the second on the details of the question. The first column in both the tables constitutes the field in the database; the second is an example entry while the third gives the description of the field.

TABLE I

	Example Entry	Description
Username	jessica	Specifies the username taking the test, as entered by the student
Date	Thursday, 23/08/2007	Extracts the system date
TestID		The TestID can be generated in two ways: a. A randomly generated ID by a simple code. b. An ID depending on the test conditions like level, no. of questions and the other fields.
level	Basic	We can keep three levels: Basic, Intermediate, Advanced

Time	10800	The total time of the test, which will be set by the administrator/teacher																																																																																																
Time_remaining	10800	(Target time) - (Current time)																																																																																																
total_testmarks	100	Total test marks as specified by the teacher. The marks for each question can also be entered but that will be in a separate table																																																																																																
total_questions	50	Total number of questions entered by the teacher																																																																																																
Is_negativemarks	N	Whether the test has any negative marks or not. This is a Boolean variable with two logical values																																																																																																
Is_finished	0	The value can be set 0 or 1. The value is 1 in the following cases: a. When the test time gets over b. When the student enters 'Submit' at any point during the exam. We can also keep an option for 'Take the test later'. In this case, is_finished=0																																																																																																
category	Math	The category of the test. In this case, category="Science". We can keep broader categories too, say Category="GRE"																																																																																																
presentquestion	1	Presentquestion contains the value on the window. This value is incremented whenever "Next is clicked" and decremented in case of "Previous". Min(presentquestion)=1 Max(presentquestion)=50																																																																																																
topic	addition	The subject under the category. e.g., Category 'Math' can include sub topics like 'Addition', 'Subtraction', 'Division' etc.																																																																																																
QuestionID		The questionID will be a 16-digit string, which will be generated depending on the values given in the question table. The 16-digits can be represented as follows: <table border="1" style="margin-left: 20px;"> <tr> <td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td> </tr> <tr> <td colspan="3" style="text-align: center;">←→</td> <td colspan="3" style="text-align: center;">←→</td> <td colspan="2" style="text-align: center;">←→</td> <td colspan="2" style="text-align: center;">←→</td> <td colspan="2" style="text-align: center;">←→</td> <td colspan="2" style="text-align: center;">←→</td> <td colspan="2" style="text-align: center;">←→</td> </tr> <tr> <td colspan="10">1-3: Category</td> <td colspan="6">10: Sub Question No.</td> </tr> <tr> <td colspan="6">4-6: Topic</td> <td colspan="6">11: Type</td> </tr> <tr> <td colspan="6">7: Test level</td> <td colspan="6">12: Skill</td> </tr> <tr> <td colspan="6">8: Question level</td> <td colspan="6">13-16: Question No.</td> </tr> <tr> <td colspan="6">9: Is_main_question</td> <td colspan="6"></td> </tr> </table>	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	←→			←→			←→		←→		←→		←→		←→		1-3: Category										10: Sub Question No.						4-6: Topic						11: Type						7: Test level						12: Skill						8: Question level						13-16: Question No.						9: Is_main_question											
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16																																																																																			
←→			←→			←→		←→		←→		←→		←→																																																																																				
1-3: Category										10: Sub Question No.																																																																																								
4-6: Topic						11: Type																																																																																												
7: Test level						12: Skill																																																																																												
8: Question level						13-16: Question No.																																																																																												
9: Is_main_question																																																																																																		
Description	Test on Kinematics	The teacher can enter the Description of the test																																																																																																
Instructions	The test contains 50 questions, each of 2 marks.	The teacher can give instructions for the test																																																																																																

TABLE II

	Question No. 1	Question No. 2	Description
questionID	MATADDBB00SA0001 MAT=Category ADD=Topic B=Test Level B=Question Level 0=is_main_question 0=Subquestion_no S=Single choice A=Application Skill 00001=Question No.	MATADDBB11FA002	Generated in the same manner as described in the previous table.
Level	Basic	Basic	Level of the question. Just like the level of the test, each question can have three levels
Is_mainquestion	False	True	I gave this option a lot of thought before finally deciding to keep it. This option can will be useful in the following cases: <ul style="list-style-type: none"> a. Fill in the blank type of questions. b. When the teacher wants
Main_questionID			The ID of the question generated in the same pattern as the questionID
Negative_marks	0	0	The negative marks allotted to the 'presentquestion'
No_of_subquestions	0	1	No. of sub questions in a

			particular main question. There can be 10 fill in the blanks in the main question.
Skill	Application	Application	Specifies the skill required to solve the 'presentquestion'
Topic	Addition	Addition	This is the same as the Topic of the test.
Total_marks	3	3	The total marks the 'presentquestion' carries
Type	Single choice	Fill in the blanks	This specifies the type of the 'presentquestion'. The types I could think of: <ul style="list-style-type: none"> a. Single choice b. Fill in the blanks c. True/False
Question_statement	What is $180 + 79$?	$100 + 273 =$ _____	The statement of the 'presentquestion'
Statements	a. 1879 b. 259 c. 189 d. 799	None	The answer statements
response	a	373	The response entered by the student
Right_answer	b	373	The correct answer to the 'presentquestion' entered by the teacher

Table I is connected at the runtime and values of the variables are changed according to the test conditions.

Table II is used for following tasks:

1. To generate a unique QuestionID.
2. To check the response of the student with the correct answer
3. To analyze the test given by the student, e.g. to judge the type of questions attempted wrong.

Educational games

Project Description:

Games can serve as interactive, fun based tools for learning. Games provide a forum in which learning arises as a result of tasks stimulated by the content of the games, knowledge is developed through the content included in the game, and skills are developed as a result of playing the game.

Games can be simple or complex. I feel that most of the knowledge-based learning takes place through simple rather than complex games because then the child is able to focus more on the learning part in the game rather than on the playing part of it. Big Math Attack is one example of a very simple Educational game. (A trial version of the game can be downloaded from <http://www.pc-shareware.com/bigmath.htm>). It tests spelling, math, typing and metric skills of the player in a fun arcade environment. There are 10 skill levels available in each category. The objective of the game is to prevent a city from falling meteorites. The meteorites are disguised as math problems/word with a missing alphabet (to test spelling). The frequency of the fall of meteorites as the game proceeds. The only way to prevent a meteorite hit the city is to give the correct answer to the falling problem. The game thus teaches a certain subject to student by creating interest, motivation and concentration in the student.

The user can also edit the database of each category.

For instance, the database of 'Spelling' contains over 1000 words. The user can change these words in the database or can create his/her own spelling lists. Such a system will also prove very useful for parents and teachers for testing the students on various sets of questions.

The game displays the result with the number of current and wrong answers and also the percentage obtained. The correct answers are also shown side by side.

Such a game can be used to teach the students a certain subject, expand concepts and assist in learning a particular topic as they play in a fun, engaging and highly interactive medium.
languages

Development Stage:

The game can be developed using C, C Graphics and by including library OpenGL. The basic task in the game is to introduce an interrupt as soon an input from the keyboard is given. The keyboard input is compared with the correct answer and if the answer is correct, the number of correct answers is incremented while in case the input doesn't match, the correct answer is displayed and the counter of wrong answers is incremented.