

The Geotechnical Virtual Laboratory

**Srboljub Masala, Kevin Biggar, Colin Geissler
University of Alberta**

Abstract

The Geotechnical Virtual Laboratory is a computer based instructional package intended to complement, and potentially replace, some physical testing in a real geotechnical laboratory. The focus of this software is toward active learning and development of critical thinking skills, including data interpretation, understanding of the processes and influential factors, and problem solving. This paper is a progress report for a software development project, which has been underway at the University of Alberta since December 1998. A module on permeability testing has been completed and beta-tested, and another module on consolidation testing is in the final stage of development. The paper covers the initial design stages of the project (task analysis, initial concept, the teamwork management, etc.) and describes the outcomes from the permeability module and the results of its evaluation. It is clearly indicated by the students' feedback that a winning combination should include both a test simulation and experimental data sets, together with the data analysis tools. As well, it should address both theoretical and experimental aspects of a geotechnical problem, and enable their comparison.

I. Introduction

The geotechnical curriculum of the Civil Engineering Department at the University of Alberta has a significant laboratory component, in which students must conduct a variety of soil tests to fulfil course requirements. The intention is to make them familiar with the most often used experimental tools in geotechnical engineering, and to establish an understanding of the role and benefits of experimental work for their future professional practice.

Problems that are recently faced at the Department are probably typical for many colleges and universities offering geotechnical programs: with increasing number of undergraduate students (60 – 90 per class) there is less laboratory space, equipment and time to allow each student to participate actively in all laboratory exercises. For example, the consolidation test procedure requires one to two weeks to test a single specimen. Few laboratories have enough space to accommodate the equipment and enough time to permit such a long test. This experiment is, thus, usually limited to a single load increment, with readings taken only for the duration of a regular laboratory class – about 3 hours. Another example is the triaxial test, a crucial experiment for the understanding of soil stress states and strength. The apparatus is expensive for teaching purposes and, also quite complicated to operate and comprehend in a limited time available for laboratory classes. Therefore, this test has been omitted from the undergraduate laboratory program.

Consequently, many students do not obtain full benefits from hands-on exercises. Incomplete testing, poor quality of the results obtained, multiple sections and large testing groups, with limited individual participation, etc. may all result in a low level of understanding of the subject. A possible conservative solution to this problem, purchasing more equipment and hiring additional laboratory personnel, is likely not the one sought in these times of fiscal restraint. The Geotechnical group at the University of Alberta has opted for a more radical solution, the development of a computer based instructional package.

The project is called the “Geotechnical Virtual Laboratory”, and is intended to complement, and potentially replace, some physical testing in a real laboratory with computer-aided instruction in the form of self-contained interactive software. It is anticipated that this will enable improved and enriched individualised learning, partially free of time limitations, which will be more productive for the instructors and more efficient and appealing to the students. The focus of the computer simulation is toward active learning and development of critical thinking skills, including data interpretation, understanding of the processes and influential factors, and problem solving.

This paper is a progress report on this project, which has been underway since December 1998. A module on permeability testing has been completed and beta-tested, and another module on consolidation testing is in the final stage of development. This paper mainly discusses the initial design stages of the project (task analysis, initial concept, the teamwork management, etc.) and describes the outcomes from the permeability module and the results of its evaluation.

II. Review of existing experience and available similar products

The search for existing geotechnical courseware and previous experience with computer-aided teaching in this discipline found two examples:

- “Geocal”, a joint project of several universities from United Kingdom, and
- a consolidation testing module from the University of Arizona in Tucson.

“Geocal”¹ is an unfinished, though almost complete software package, obviously intended to cover many areas of soil mechanics and geotechnical engineering. It consists of a few modules in various stages of completion, produced by different institutions and authors, with relatively heterogeneous design concepts and user interfaces. Particular values of this package are the comprehensive theoretical section with many quizzes, and a variety of laboratory tests and procedures.

The consolidation test module² from the University of Arizona is courseware available on the World Wide Web. It is limited to the oedometer testing, but the topic is exhaustively elaborated: it contains theoretical background, a detailed description of the equipment and working procedure, a test simulation and guidelines for user activities. The software contains tools for ‘test’ data processing and nice features like ‘setting-up the apparatus’ knowledge test.

What these programs were lacking from our perspective is the following.

- They are simply input – output machines, the outputs are either numbers or diagrams; there was little attempt to visualize the testing process.
- The equipment is described conceptually; the pictures shown are schemes without details. This is good for a supplementary teaching tool, for parametric analyses, etc. but has limited value in teaching the conduct of particular experiment, especially if the procedure will not be performed in a real laboratory.
- Finally, the soil models used are elementary idealizations intended only to illustrate theoretical concepts. These artificial materials exhibit very simplified test responses so that complex aspects of the behaviour of real materials cannot be fully addressed. The absence of real data (the results of tests on natural materials) in such teaching programs may be dangerous. It may develop exaggerated trust in the strength of numerical methods, without understanding of their limitations, and not prepare students to deal with the diversity of natural conditions and the peculiarities of real materials.

III. Instructional design – starting concept

What was particularly beneficial in the planning stage of this project were discussions with our colleagues from the Faculty of Engineering who had been already involved in computer aided instruction. They turned our attention to significant points like careful planning of time and budget, the necessity for standardization, careful development of user interaction, the need for feedback during development, etc.

Following our colleagues' advice, we established a co-operative relationship with the Academic Technologies for Learning (ATL) Production Studio. This unit was formed to support the design and development of educational technologies at the University of Alberta. The cooperation with ATL helped bridge significant instructional and technological obstacles encountered during the work.

The design of the “Geotechnical Virtual Laboratory” followed established processes in instructional design (Figure 1). The comments that follow discuss certain specific aspects that are usually not apparent, but may be helpful for those interested in engaging in a project of this sort.

1. Analysis

The *learning environment* was relatively difficult to define, particularly because of the continuously changing technological environment. It was concluded that the program must be available both in the Department computer labs and at home, to relax the time constraints on students. The survey among our students showed that the home computers, although variable in performances, are generally more advanced than those currently in university computer labs. The latter posed grave restrictions for our planning. As a result of policies that inhibit user installed software; that there were no audio cards installed in the labs; and that the processor speed and RAM memory were relatively low, etc. we had to seriously consider the delivery format, programming language and media content of the final product. As a result, the configuration of the computers in the labs became the minimum configuration required to use the software.

Stages	Topics	Tasks
Needs Analysis	Learning context	Learning environment Access to the learning system Available time
	Learners	Prior learning / experience Attitudes and goals
	Learning task	Learning goals Assessment of achievements
Development	Organization	Types of content Content structure Instructional strategies Interface
	Delivery	Types of media Multimedia elements Programming environment
	Management	Working team
Evaluation	Formative	Testers Questionnaire Revisions
	Summative	Testers Questionnaire Content update

Figure 1 Instructional design process

We also discussed the instructional context and determined early on in the process that an individual test should not exceed in duration the length of regular laboratory class – 3 hours. However, additional material may be included for students who choose to explore the content further. The total time estimated for students to spend on the virtual experiment was limited to about 5 to 6 hours. This is what is normally needed to conduct the physical test and to write the lab report.

The *learners analysis* showed a very homogeneous group in terms of age (third year of undergraduate studies), gender structure (male / female about 60 / 40 %) and attitudes and goals (the course is a compulsory one).

The *learning task* was the most difficult to deal with. We postulated there were two possible ways to develop this courseware. In most general terms: either as a coherent ‘story’, with strong connecting thread in the form of a (design) problem to be solved, or as a loose assembly of individual modules performing separate tasks. The latter is a more flexible design where every instructor may find his own preferences and determine a student’s path through the program using a supporting printed laboratory handout.

During the planning, the teaching of the Soil Mechanics course was in a transitional stage: a professor, having taught the course for many years, handed it over to two new staff members (one each term). Each one of them had own view of the course, including laboratory work and the testing program.

Opinions on how to present the information ranged from, on one hand, teaching only procedures and methods to, on the other, developing a comprehensive theory section to patiently explain the basic concepts, including a number of knowledge tests. Similarly, opinions about the experiment simulation strategy ranged from a “step-by-step” guideline and feedback to every user action, to a “neutral” interface that provides all necessary tools and allows the user complete freedom in their application.

These diverse opinions precluded a simple story telling option, requiring an alternative of a module menu with selected experiments as separate units. To compensate for that, and to enable the interested student to obtain more information, it was decided to equip each teaching unit with the following:

- a solid theoretical section,
- detailed explanations of testing apparatus,
- standardized procedures,
- discussions of influential factors,
- advantages and limitations of the tests, etc. and
- supporting content by way of numerous pictures, animations and digital video.

2. Strategy

The *software organization* was another major challenge, this time because of heterogeneity of the content. The three tests that were chosen for the virtual laboratory were permeability, consolidation and triaxial test. They differ significantly from one another in terms of: the theoretical aspects of soil behaviour they are intended to examine, the apparatus used in experiments, the procedures, data analyses prescribed, etc.; very simply, they differ in the entire visual aspect of their description, simulation and presentation.

The permeability test is the simplest from a theoretical point of view, but the most attractive regarding screen presentation (water flow, level changes, etc.) and the high level of user interaction that can be elicited. On the other hand, the consolidation test in an oedometer is visually very simple and practically deprived of any user interaction, although the background theory is much more complex. Therefore, the type and nature of the content dictated the presentation style required to keep the user engaged and interested.

The *delivery of a final product* was a subtle point. As explained earlier, it was decided that the program would be self-contained, capable of running on various hardware platforms, including those that do not permit any installation procedure (i.e. the University computer labs). Therefore, a recordable compact disc (CD) was practically the only choice for delivery.

Along the same lines, printing the results of the student's virtual laboratory session (graphs, data tables, etc.) seemed very difficult to achieve directly from the courseware. It was decided, thus, to provide means for data transfer to common specialized programs (spreadsheets, text editors) which would provide the necessary environments for finalizing laboratory reports. Data transfer from the courseware to other programs was limited to text files, which can be saved to or read from any floppy disc, again because of limitations imposed by the computer labs.

The choice of *programming environment* (software tools) was directed by the following considerations:

- hardware platforms (although MS Windows has become almost a standard),
- multimedia capabilities (to easily accommodate various types of content: text, pictures, animations, video, audio, etc.),
- software compatibility (for data exchange with various auxiliary programs which may be used for preparation of the multimedia material for the courseware),
- the development platform's longevity (because of the future maintenance of the product), and
- the programming skill set needed (again because of the maintenance).

The selection of Macromedia Authorware³, as the most versatile and simple platform, and due to its long and successful history, seemed the logical choice in a relatively narrow competition. It was also widely used and supported by our partner, ATL.

The *working team* was composed of:

- from the engineering side: a director and content expert, a manager and writer, and 3 programmers; and
- from the educational technology side, as an advising group: an instructional designer, an authoring specialist, a graphical design specialist and an evaluation specialist.

The programmers were three 2nd year co-op students from the computing engineering undergraduate program, with noticeable inclinations toward certain specializations in programming (computer graphics, algorithmic programming, etc.). The decision to hire them instead of professional programmers was dictated mainly by available resources at that time.

3. Evaluation

The *evaluation plan* was intended to consist of two stages:

- formative evaluation with a focus group, consisting of small representative sample of students who have already taken the appropriate course, with various level of course grades, to provide the feedback to the project team that would enable improvements during the software development; and
- summative evaluation, on a number of enrolled students in the course, proposed as a regular class in a computer laboratory, with questionnaires and automatic tracking of user's activities by the software.

This plan was greatly affected by the circumstances and development of the work. The formative evaluation took place as a testing of an initial version of the permeability module, with a group of graduate students. The main task was to assess the proposed organization (menu structure) and navigation (user interface). Although consisting of only 5 people, the testing group did its job very well, as it will be seen later in the text, in the section describing the results of beta testing of the software. Their preferences of proposed options, although sometimes divergent from the wishes of members of working team, appeared to be very close to the choices of the students which were involved in the summative evaluation (beta testing) of the permeability module six months later.

IV. The permeability module

This is the first module brought to the beta testing stage, after about 2 months of planning and 4 months of programming. The module is embedded into an introductory menu structure – a ‘shell’ (shown in Figure 2). From this menu the user may select a test or obtain general information about the program (the purpose, the participants, the description of the screen layout and graphical navigational tools, instruction for the use, etc.).

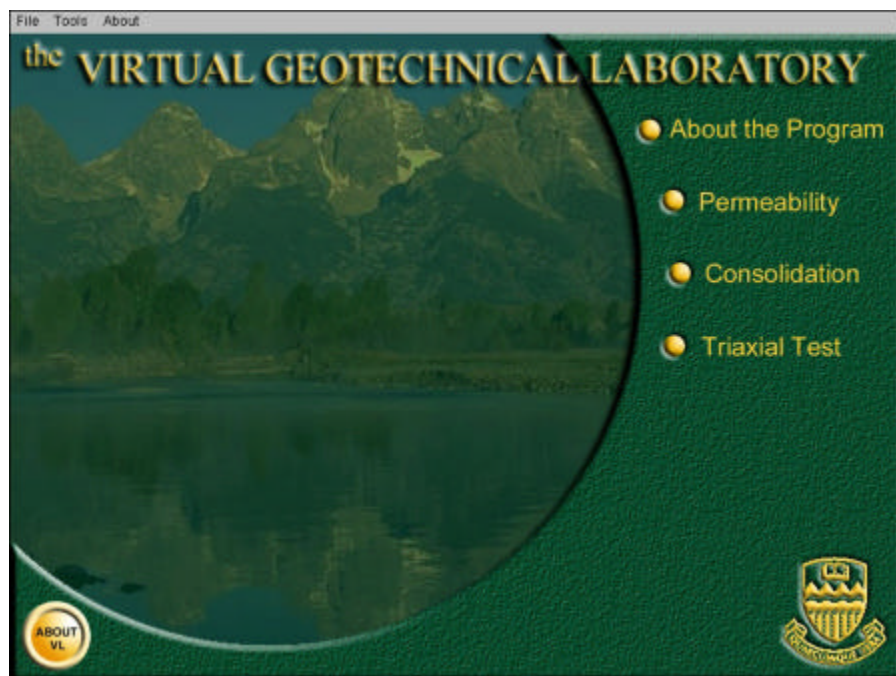


Figure 2 The introductory menu

The *design* was adopted as a flexible scheme of nested menus - a hierarchical menu tree. Each teaching unit may be accessed directly from these menus. The intended initial program structure, including the shell, is presented in Figure 3 (solid lines outline currently developed topics). This flow chart shows the main structural units (the main menu topics in this case). The columns in this scheme are actually individual menus. It should be mentioned that the depth of menu nesting

is not limited in Authorware, but the speed of execution may be affected by the complexity of menus.

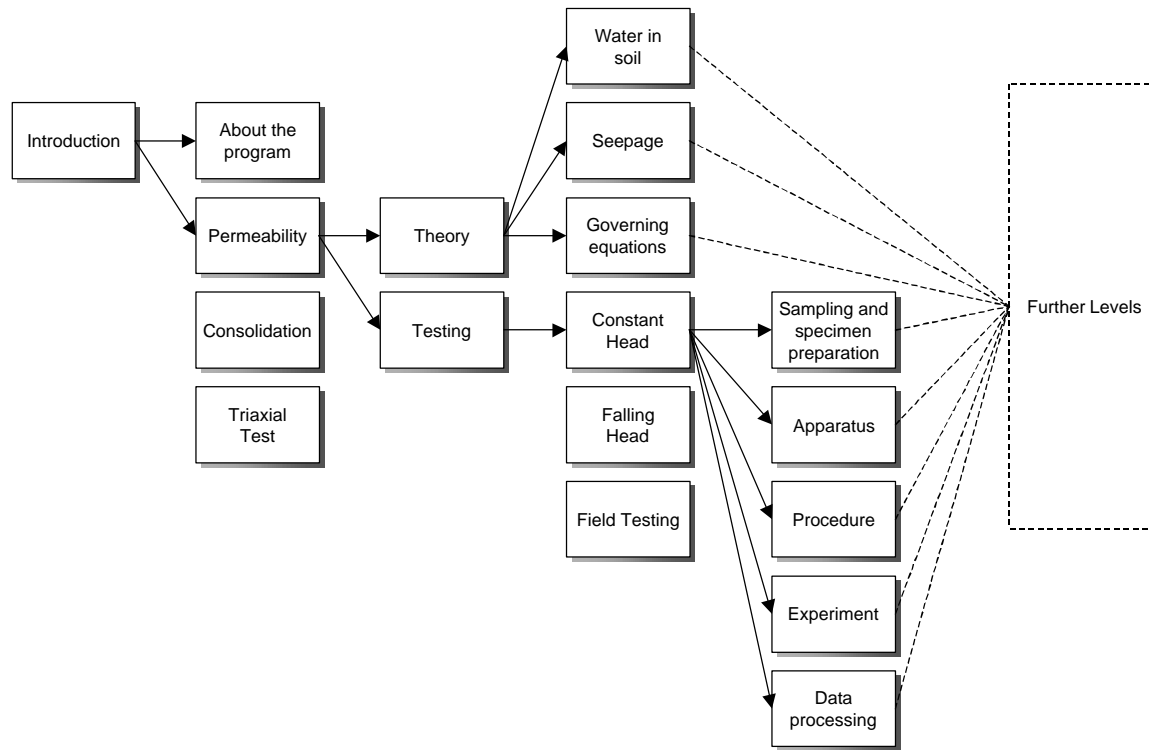


Figure 3 The menu structure

Screen design is an issue of great importance for multimedia projects. It has a crucial role in the delivery of the information to the learner. In our case, regardless of the diversity of the content, a uniform and consistent screen layout had to be designed to provide effective instruction, appropriate navigational tools and visual aesthetics. Our solution, without pretensions to be the unique or the best one, appears to conform well to the suggestions elaborated in the pertinent literature⁴.

The screen is organized into *functional areas*, as seen in Figure 4. These areas appear in the same locations throughout the program for consistency. The three bars at the top contain the navigational tools (the menu bar and the button - icon bar) and the information about student's location in the program (the title bar). The menu panel on the right is the main navigational area. It can accommodate up to three levels of nested menus.

The body area on the left serves for content display. It is possible to mix text with still graphics (diagrams, graphs, photos) and animations / movies. The text is not scrollable, but is displayed in chunks the size of which depends upon the amount of graphics simultaneously appearing in the area.

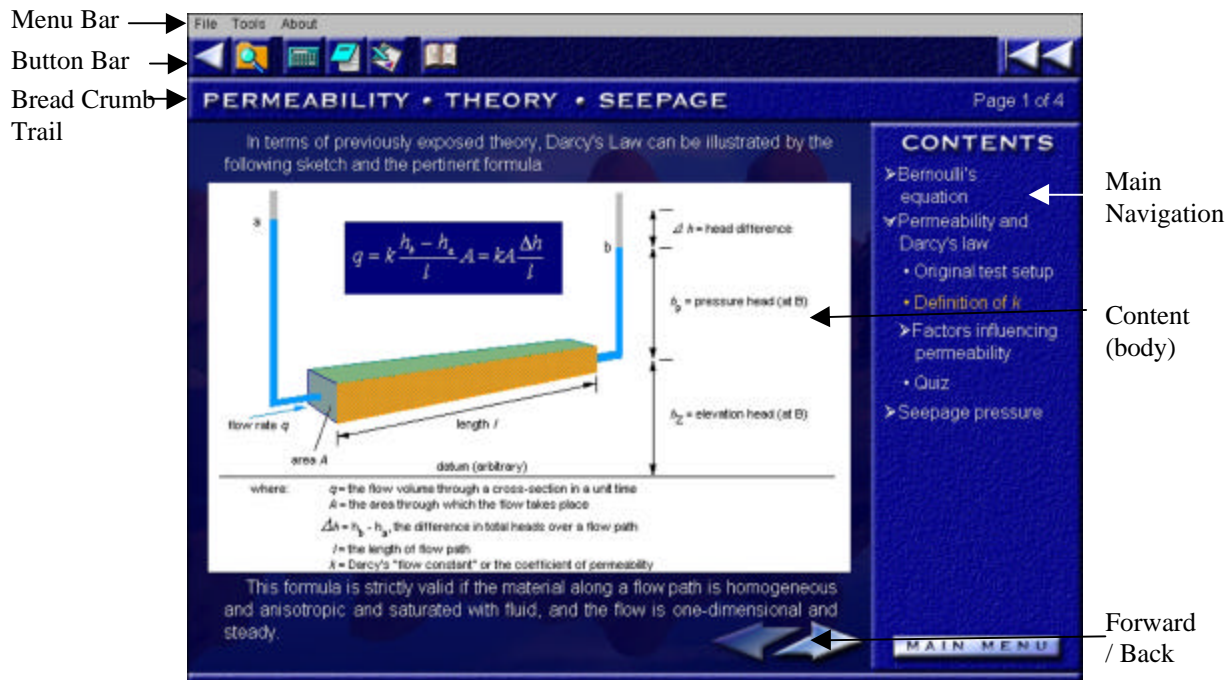


Figure 4 Screen design

Simple linear navigation from one content page to the next or previous is accomplished by using the "forward" and "back" buttons in the bottom corner of the body.

The title bar also serves as a table of content for the entire module: the words displayed are “hot spots” – by clicking them the user goes to the chosen menu or section. This redundancy provides more depth to the menu tree. Taking into account the menu panel on the right, it is possible to directly access up to six levels of menus in this way.

As can be seen, the user is provided with great freedom or ‘learner control’. It should be mentioned that a significant portion of instructional design literature⁵ expresses doubts about merits of full user control in educational software, being more inclined toward a guided use of such a software or, in technical terms, an adaptive computer control. In this sense, our solution may be considered as a compromise. Six levels of menus is almost enough to completely cover the exposition of theory, where immediate jumps are allowed to a selected topic, but the experiment simulation does not require so many menu levels. Even more, since an experiment is generally a sequential activity with a previously specified order, a simulation is subjected to relatively rigid control guidelines; occasionally, the user is not permitted to proceed unless he has finished all required work. In this way, the software allows for a user's freedom to select content in areas where there need not be a strictly linear or hierarchical progression.

User interaction is mostly reserved for the simulation and the quizzes. It was assumed that the theory section is primarily for review and that the user will have significant prior knowledge of the subject (being taught that in the classroom). Thus, the main goal of this section is not to teach

new content, but to provide an opportunity for review and remediation. Consequently, it was decided that there is no need for user interaction beyond simple navigation in that section.

The experimental portion developed so far is limited to the simulation of the constant head permeability test, which is applied to the testing of sandy materials. Other sections (referred to in Figures 3 and 4) are provided as menu topics without content (but they will be added in later versions of the software). The principle reason they were not provided in this version is that they are not in the experimental program in our current undergraduate geotechnical curriculum. However, they will be relatively easy to develop, being visually more ‘static’ than the constant head test.

The *test simulation* in the permeability section consists of two parts, the downward and the upward flow. The former serves to determine the value of the coefficient of permeability, and the latter for the evaluation of critical hydraulic gradient, when piping, or so-called ‘soil boiling’, starts. The simulation graphics screen for downward flow is shown in Figure 5. The screen for upward flow is similar, with the appropriate number of control buttons on the right, and with additional graphics animation that visually denotes the beginning of piping.

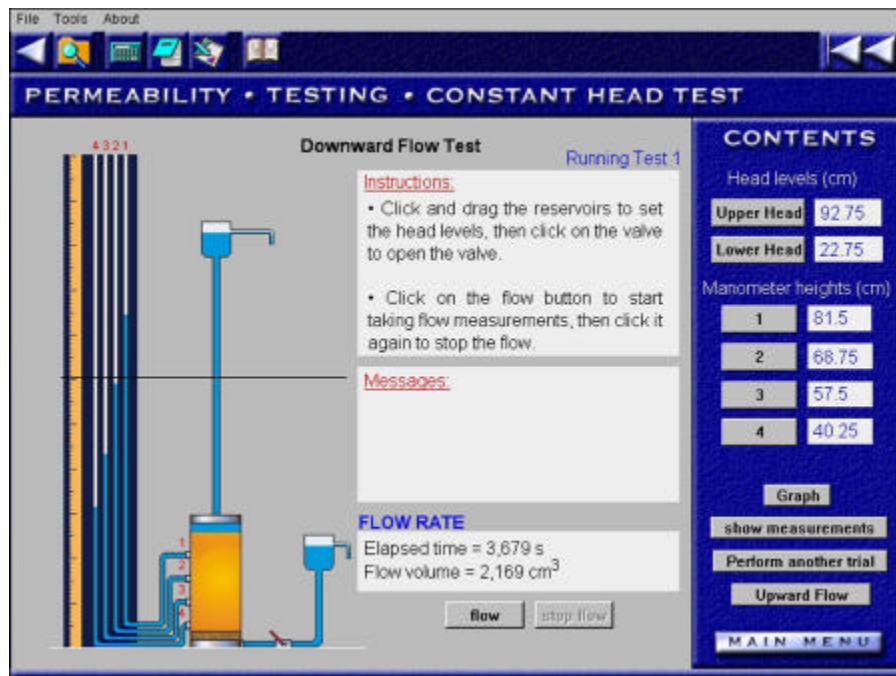


Figure 5 The simulation screen

The use of simulation is completely graphical. Certain parts of the picture of the apparatus are movable. They are either program controlled or user controlled (movable by mouse). For example, to start the test, one must first determine the upper (incoming) and lower (outgoing) water head. This is accomplished by dragging the reservoirs along a vertical axis. The valve is then opened to allow water flow, by clicking on the valve in the picture, which is a *hot spot* that

activates the appropriate procedure. The program then calculates and displays the positions of the water levels in the piezometric pipes; this is a time development process, out of user control. To help the user determine whether the process has stabilized, there is the *Graph* button that shows the diagram of piezometric levels versus time.

The collection of data is again a purely graphical process. With the aid of a horizontal line from the figure (dragging it over the apparatus), and clicking on one of the buttons on the right panel, the user can determine needed geometric values (the soil sample top and bottom, the levels of piezometric outlets, etc.) or hydraulic parameters (the upper and lower head, the levels in piezometers, the flow rate, etc.). The help is provided in the two boxes: one showing what user actions are expected next and the other giving messages about possible omissions and errors.

The use of *feedback* is an important issue that should not be ignored in multimedia software. Feedback should provide information about correctness and appropriateness of user response. In addition to all of the feedback provided in the simulation in terms of appropriate user input and actions, there needed to also be feedback in regards to the application of this learning.

Feedback found its place in the quizzes that were provided at the end of each main theory unit (lesson). The quizzes were intended to strengthen the main theoretical concepts, to test the user's understanding of basic definitions and terminology, and to examine their applicability through simple problems, the solution of which required little or no mathematical computation. The challenge here is to try to predict the usual mistakes a student might make and suit the feedback so that it resembles what a real teacher would respond in each case. An example of such a question is shown in Figure 6.

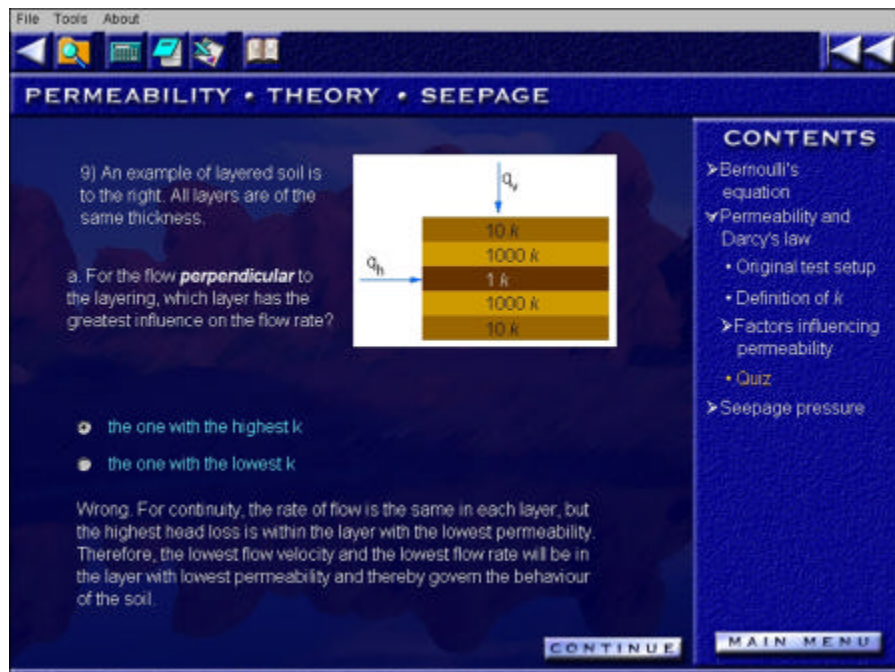


Figure 6 Quizzes: a question and the feedback

As can be seen, feedback is always provided immediately following a student's response and it is kept on the same screen with the question and student's response. For incorrect responses, the student is often given a hint and asked to try again (this is not the case with the question in the figure since only two answers are possible). Feedback was always encouraging and would include a basic theoretical explanation that described the phenomenon mentioned in the question.

In any case, to prevent potential cheating when the program is used for marking purposes, the questions are programmed mostly as simple procedures with variable, randomly chosen input. For example, in multiple choice questions the order of proposed answers is constantly changing with each new run of the quiz. In the case of numeric calculations, the answer is computed during the run-time, depending on the random input within a priori determined limits.

Animations were used sparingly, for particularly effective presentations and explanations of dynamic processes. The voice narration was excluded from the first version of the program because the computing laboratories did not support audio.

Motion video was prepared in the form of a 5-minute movie, to be run in a separate window. This was considered especially important if the program were to be used as a replacement for a real experiment. In such a case, the developers felt video material showing the most important stages of sampling and specimen preparation, together with the apparatus assembly and its handling during the test was required.

In addition, certain visually attractive phenomena, which could not be easily simulated on a computer (like piping in upward flow) were filmed as well. Unfortunately, due to some technical difficulties (the incompatibility of the format of video file and the one that our software platform was capable of reproducing) the video clip was not incorporated in the current version of software (but it was provided separately on distributed CD's). The video was filmed and processed by the working team, and it turned out to be a tedious job for a group of amateurs. A professional organization will be hired to make the final version of this movie clip.

V. Formative survey (beta testing)

The beta testing of the permeability module was organized with a group of students enrolled in the Soil Mechanics course, who had already performed this test in a physical geotechnical laboratory.

The group consisted of 10 people, ranging in age from 20 to 30 years. A bit surprisingly for engineering students, only seven of them had their own computers at home. Nevertheless, the properties of this group reflected well the conditions of a student population, surveyed earlier by the university administration for other purposes. Therefore, our sample group was representative for the anticipated (target) audience.

A questionnaire was prepared with the questions grouped according to the general issues of:

- the overall impression,

- the civil engineering content, and
- the presentation and aesthetics.

The answers are presented in Figure 7.

Areas of suggested improvement include:

- separating the theory and reorganizing it into two sections: the brief review of main concepts and an additional or extra information (this was common for all comments),
- increasing visual items, especially including video material,
- adding more interactivity when going through the theory,
- improving the glossary and search function,
- bookmarking ability, etc.

It seems necessary to explain an apparent contradiction between a warm acceptance of this software and noticeable rejection to use it as a replacement for the real laboratory test. It is perhaps the best to quote a few opinions.

“Although the computer test was faster to complete, I found I learnt more from performing the test myself.”

“The simulation is good because it is ‘purged’ from redundant details that you have in a real test. It is better as an introduction to the laboratory work, but it cannot substitute a physical lab.”

“I don’t prefer the lab over the computer simulation or vice versa. The lab should not be excluded from the curriculum as it provides useful hands-on learning. The computer simulation could also be added as a supplementary addition to the course.

When in the actual laboratory, we worked in groups of 6 people. Each person had the responsibility of only one task (i.e. measuring height, time for flow rate measurement, etc.). It was hard to keep track of everything going on at that time. Doing the simulation on the computer allowed the user to take all the measurements and be involved in all aspects of the experiment. It was easier to understand in the simulation.”

This apparent scepticism can find an acceptable explanation in students’ suspicion in the powers of the computational models used. Such a critical attitude is, eventually, a characteristic of a scientific thinking. Theoretical models are perceived as (only) approximations to the complexities of natural phenomena. The students still believe that there may be something *surprising* in a physical laboratory, and they expect that there is more intellectual challenge and reward in an attempt to interpret their own data obtained in a real experiment, though often erratic and irregular.

Overall impression	
Overall impression of the software	Positive to very positive
Should the physical test be maintained?	Yes, though nearly 50 % were undecided.
Looking forward to the next virtual experiment?	Mostly yes
Is a demonstrator or help needed?	No
Hours spent with the entire lab	Most (70 %) spent less than 5 hours
Hours spent with the simulation	Most (70 %) spent less than 2 hours
Likes (the best)	Likes seemed to be towards the simulation and animations. A number liked the theory (certain parts). Other likes include the quizzes, and the freedom to do it on their own time.
Dislikes (the worst)	Focused on excessive theory, and an uncertainty as to what was important in the content
Content	
The organization (menu structure)	Appears logical and comprehensible
Able to realize what was important in the content?	Sometimes there was ambiguity in determining it
The amounts of various content:	
Theory	Too much
Experiment	Enough
Quizzes	Enough
The theory as a separate unit?	Most agreed
Terminology	Familiar and consistent
Textual information	Legible
Pictures and animation	Useful to very useful
Confused with more than one animation / picture on the screen at a time?	Little confusion
Quizzes	Useful to very useful
Knowledge level required for the quizzes	Average
Feedback in the quizzes	Useful to very useful
Presentation	
Navigating through the program	Easy to very easy
Buttons, icons, and hotspots	No problems with understanding what they meant
Screen setup	Deemed quite appropriate
Text (fonts, sizes, etc.)	Legible
Change colour scheme (use dark letters on a bright background)?	No consensus, though there was no dissent on the existing scheme
Scrolling in the text?	No one wanted it - new screens preferred
Quality of pictures and animations	Good to very good
Use of voice (narration)	No consensus, most were undecided
Simulation tools - easy to use?	Yes, with one exception

Figure 7 Beta testing results

Having postulated this as a positive tendency, we decided to follow the direction that this survey had pointed to in the modification and further development of our courseware. A virtual test, however sophisticated it may be, is taken merely as a 'purified theory' and a training tool. It is absolutely necessary to complement it with the sets of real data, obtained in its physical counterpart – a real experiment, and the analysis tools to process these data. Real data should be selected in such a way to illustrate the extent of variation of behaviour of real materials and show the limits of relevant theories. After analyzing a data set, the user should be allowed to simulate the same problem in a virtual test and then compare the model behaviour with a real one. Such a comparison closes a learning cycle and provides enough material for further thinking on the subject and, what is more, develops in the user a desired general attitude and approach to the problems in their professional field.

VI. Conclusions

Based on our experience, the following items are deemed of utmost importance in computer aided instruction.

Project Planning and Management

1. Plan carefully. Analyze both the audience and the instructors. Define the goals precisely – be modest. Do not underestimate the time needed.
2. Design the program structure with as much flexibility as possible, for modification and maintenance purposes. Standardize everything: general look, notation, colour schemes, fonts, figures, etc.
3. Compose carefully the working team to cover all important aspects of the project. Hire professionals to cover the critical areas of expertise. Develop strong management structure, with permanent team links and regular meetings.
4. Organize early testing of the pilot version of your program and verify user interface, navigation and interactivity. Provide systematic periodic evaluation with feedback during the software development.

Permeability - Virtual Experiment

5. The software developed was well received, easy to use, and augmented understanding of the experiment and theory very well.
6. Given the choice, most students would prefer to do the physical experiment as well.

Acknowledgements

A generous donation from the Canadian Pacific Charitable Foundation was the primary financial source for supporting the work on the Geotechnical Virtual Laboratory. The authors also acknowledge in-kind contributions to this project of their colleagues from the Geotechnical

Group Dr. Robertson, Dr. Chalaturnyk and Dr. Brachman, and helpful discussions with the members of the Faculty of Engineering Dr. T. Hrudehy and Dr. R. Toogood.

Bibliography

1. Geotechnical Engineering Computer Aided Learning (GeoCAL), HEFC Teaching and Learning Technology Programme: Project 58 (1993-1997). Former URL: <http://cl24.uwe.ac.uk/GEOCAL/GEOCAL.htm>
2. URL: <http://www.u.arizona.edu/ic/CE340>
3. URL: <http://www.macromedia.com/software/authorware>
4. Stemler, L.K. Educational Characteristics of Multimedia: A Literature Review. *Journal of Educational Multimedia and Hypermedia*. Association for the Advancement of Computing in Education, Charlottesville, VA. Vol. 6, No. 3/4, 1997, pp. 339-358.

SRBOLJUB MASALA

Srboljub Masala is currently a research associate at the Civil and Environmental Engineering Department of the University of Alberta in Edmonton, Alberta, Canada. He received his B.Sc. in Civil / Structural Engineering from the University of Belgrade, Serbia, Yugoslavia, and an M.Sc. in Geotechnical Engineering from the University of Alberta. Mr. Masala is a registered Professional Civil Engineer in Yugoslavia where he was working for 17 years.

KEVIN BIGGAR

Dr. Biggar is currently an Associate Professor in the Department of Civil and Environmental Engineering at the University of Alberta. He received his B.Eng. in Civil Engineering from Royal Military College of Canada in 1979, and his Ph.D. in Geotechnical Engineering from the University of Alberta in 1991. He has taught courses in Civil Engineering since 1991, with an emphasis on geotechnical engineering and material behaviour.

COLIN GEISSLER

Colin Geissler is an Instructional Designer with Academic Technologies for Learning (ATL) at the University of Alberta in Edmonton, Canada. He is currently finishing a Master of Arts, Educational Technology degree from Concordia University in Montréal, Québec.