Using VCSU-MEP

An Introduction to the VCSU-MEP MOO Educational Platform

Introduction.

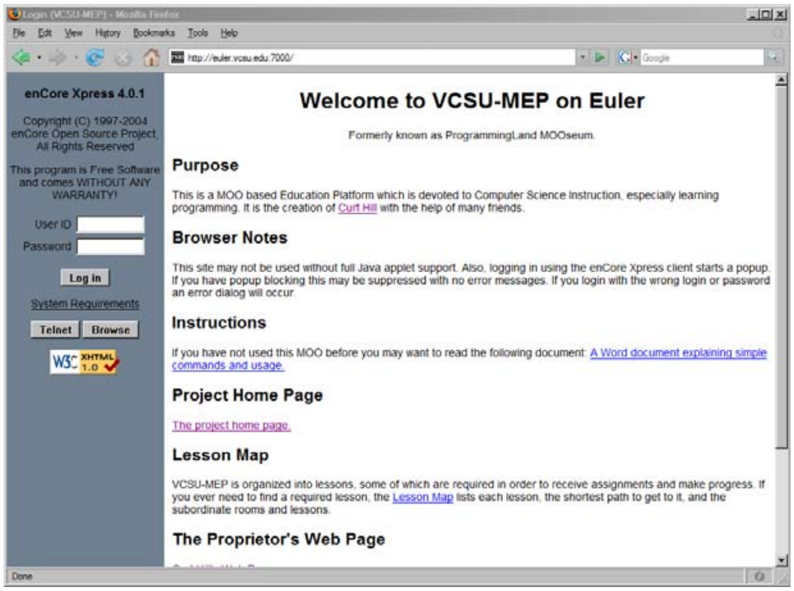
The VCSU-MEP system (formerly known as ProgrammingLand) is a hypermedia textbook and course in several topics of Computer Science available on the Internet. It implements an Exploratorium-style museum. It is a virtual museum with a text-based interface. Students participate in a self-paced exploration of the exhibit space as they browse through the museum. In their exploration they are introduced to the concepts of computer programming as well as the concrete details of programming languages. There are interactive demonstrations of the course content, which the students are encouraged to manipulate as part of the learning process.

The rest of this document is divided into the following sections so that a potential student need only browse what is lacking. The first section is for those who have never used a MOO or MUD before. The experienced MUD users may safely bypass this part. The next section is about the lesson structure of this MOO and how assignments are given to students. The next section considers basic commands common to most MOOs.

Use of the enCore Xpress client on VCSU-MEP.

Students may access VCSU-MEP using a Java enabled web browser or a variety of free or shareware client programs. Since the interface is text-based even a moderate speed modem is fast enough. See following URL for more information:
`http://community.vcsu.edu/facultypages/curt.hill/My_Webpage/clients.htm`

The overall appearance of the interface depends on the client that the user chooses. The preferred client is currently a Java-enabled browser using enCore XPress. The login screen is shown in Figure 1.


Login (VCSU-MEP) - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://euler.vcsu.edu:7000/
Google
User ID
Password
Log in
System Requirements
Telnet
Browse
W3C XHTML 1.0
Done

enCore Xpress 4.0.1

Copyright (C) 1997-2004 enCore Open Source Project, All Rights Reserved

This program is Free Software and comes WITHOUT ANY WARRANTY!

Welcome to VCSU-MEP on Euler

Formerly known as ProgrammingLand MOOseum.

Purpose

This is a MOO based Education Platform which is devoted to Computer Science instruction, especially learning programming. It is the creation of Curt Hill with the help of many friends.

Browser Notes

This site may not be used without full Java applet support. Also, logging in using the enCore Xpress client starts a popup. If you have popup blocking this may be suppressed with no error messages. If you login with the wrong login or password an error dialog will occur.

Instructions

If you have not used this MOO before you may want to read the following document: A Word document explaining simple commands and usage.

Project Home Page

The project home page.

Lesson Map

VCSU-MEP is organized into lessons, some of which are required in order to receive assignments and make progress. If you ever need to find a required lesson, the Lesson Map lists each lesson, the shortest path to get to it, and the subordinate rooms and lessons.

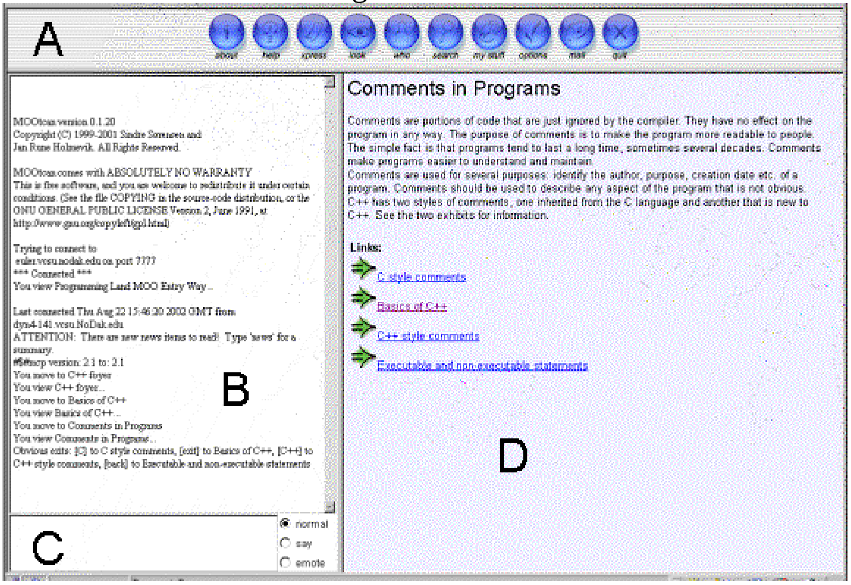
The Proprietor's Web Page

Figure 1. The login screen.

In Figure 1 there are two distinct parts, the area on the left side is where a login and password are entered. The user name is put into the first edit box and the password into the one just below it. The login is *not* case sensitive, but the password is. You should have received your login and password from the instructor. Pressing an Enter with the password or clicking on the Log in button will enter you into VCSU-MEP. This area has two alternate ways of using the MOO. The Telnet button starts an alternative client to the previously mentioned mechanism. This client is not graphical in nature, so is not recommended. The Browse button does not require a login but limits the interactivity of the MOO.

The area to the right of this contains several links to resources on MOOs and information on encore.

If the procedure for logging in was followed the regular view of the MOO should look something like the screen shot shown in Figure 2.



```
MOOtcan version 0.1.20
Copyright (C) 1999-2001 Sindre Sorensen and
Jan Rune Holmevik. All Rights Reserved.

MOOtcan comes with ABSOLUTELY NO WARRANTY
This is free software, and you are welcome to redistribute it under certain
conditions. (See the file COPYING in the source-code distribution, or the
GNU GENERAL PUBLIC LICENSE Version 2, June 1991, at
http://www.gnu.org/copyleft/gpl.html)

Trying to connect to
 euler.vcsu.nodak.edu on port 7777
*** Connected ***
You view Programming Land MOO Entry Way...

Last connected Thu Aug 22 15:46:20 2002 GMT from
dyn4-141.vcsu.NoDak.edu
ATTENTION:  There are new news items to read!  Type 'news' for a
summary.
#$#mcp version: 2.1 to: 2.1
You move to C++ foyer
You view C++ foyer...
You move to Basics of C++
You view Basics of C++...
You move to Comments in Programs
You view Comments in Programs...
Obvious exits: [C] to C style comments, [exit] to Basics of C++, [C++] to
C++ style comments, [back] to Executable and non-executable statements
```

Comments in Programs

Comments are portions of code that are just ignored by the compiler. They have no effect on the program in any way. The purpose of comments is to make the program more readable to people. The simple fact is that programs tend to last a long time, sometimes several decades. Comments make programs easier to understand and maintain.
Comments are used for several purposes: identify the author, purpose, creation date etc. of a program. Comments should be used to describe any aspect of the program that is not obvious. C++ has two styles of comments, one inherited from the C language and another that is new to C++. See the two exhibits for information.

Links:

- C style comments
- Basics of C++
- C++ style comments
- Executable and non-executable statements

Figure 2. A sample client screen.

The four panes are labeled A-D and will be described next. The A pane is a series of clickable buttons, only a few of which will be described here. The fourth button is labeled Look and displays the room description in the D pane. The next one is the Who button which shows other people who are currently logged into the MOO. Since every room in a MOO is a chat room you may communicate with these using the Say or Page commands. The last one is the Quit button, which disconnects the player from the MOO.

The second pane, labeled B, is used to show your route and communicate with other players and some of the interactive objects. The messages there show your movements as you move to an exhibit and display it. The display of the exhibit occurs in the D pane. If another person speaks to you the message will be displayed in the B pane. If you interact with one of several interactive objects the display may also be in the B pane. The results of the @course and @requirements commands (which will be discussed later) will show in the B pane as well.

The C pane is where commands may be typed in. Most of the navigation of the MOO will be done in the D pane, but the C pane is where most other commands may be entered. In the last part of the C pane you can see the exits that may be taken. The C pane is required for the chatting to other characters and interacting with many other objects, such as goalies, lessons, code machines, and work benches.

The D pane usually shows the description of the room that you are currently in as well the exits that are available. Usually a student navigates in this pane by clicking on the links. If another person is present in the room or another type of object is present in the room they will be shown in this pane as well.

A typical session for a student starts with the execution of a client program that connects to the museum. VCSU-MEP itself is structured into rooms, with exits being the path from one room to another. The motif of VCSU-MEP is that of an Exploratorium style museum. Therefore the term exhibit is often used instead of room. Whenever an exhibit or room is entered, its description is displayed. Typically this is a paragraph or two of expository text on some instructional topic. The exhibit also indicates what exits exist and where they lead. In the spirit of user-centered control, the student is always free to choose which path to take and what to do next. Although an exit is a one-way path from one exhibit to another, they usually come in pairs so that a student may return conveniently.

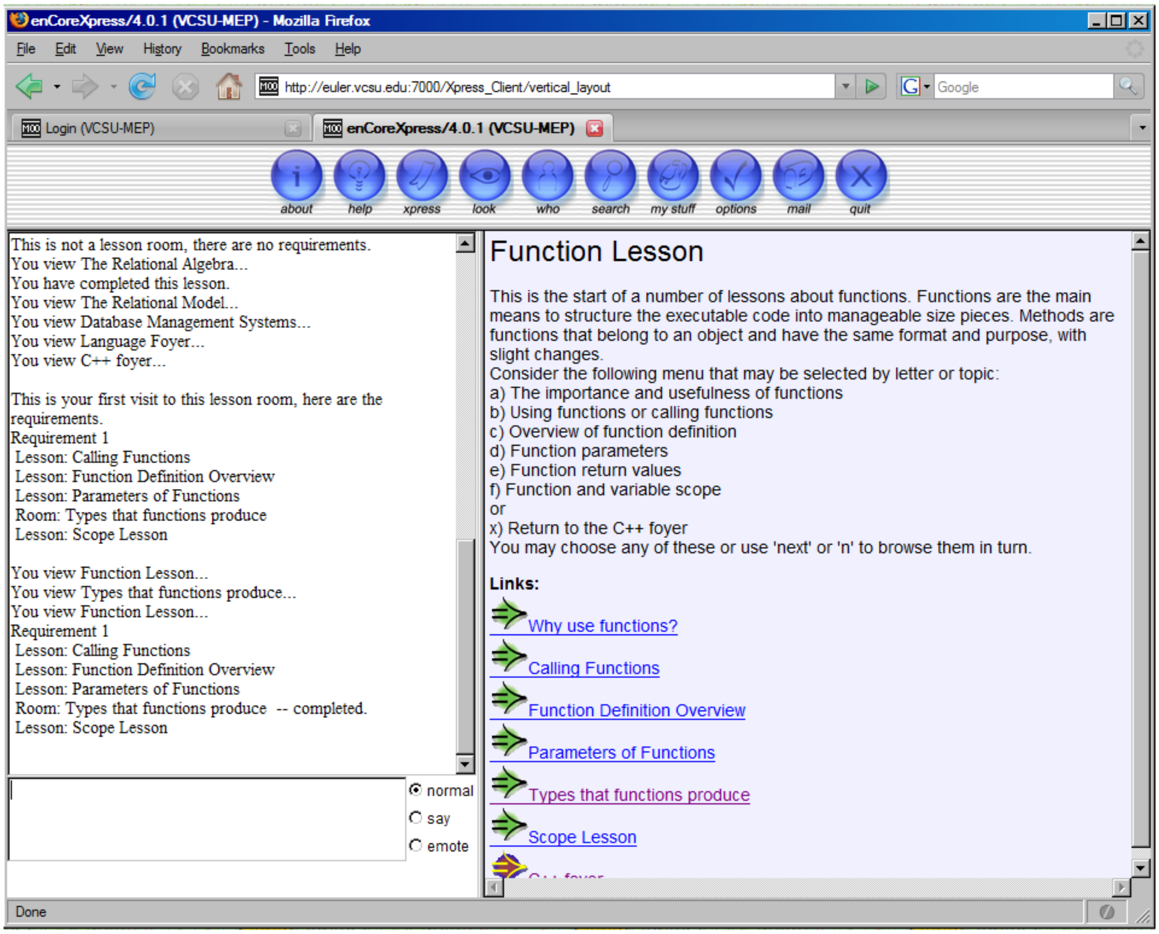
VCSU-MEP Museum is composed of "wings" dedicated to different programming languages: including C++, Lisp and Java, as well as a Database Management Systems. Most wings have several interactive objects, usually presentations or code machines. Some of these will be discussed later.

Therefore, the VCSU-MEP experience is one of selecting a room to enter, reading the information that is presented. This information may be course content, such as in the first figure, or merely directions as to where to go next. The student then decides what to do next: interact with an object or person or move on to one of several other rooms.

The Lesson Structure of VCSU-MEP.

There are at least four types of exhibits that are worth noting here: regular rooms, menu rooms, lesson rooms, and work rooms. The regular room just shows some instructional text and the bulk of the MOO is made up of this type of rooms. A menu room usually introduces a topic but gives little information on it. Instead it directs the student to a series of other rooms that cover the aspects of the topic. The lesson room is often a menu room but heads up a lesson and will be discussed next. The work room is a room that only allows one student at a time. It usually houses an interactive object that cannot be shared.

Groups of exhibits are clustered into lessons, where a lesson contains a single topic. Usually a lesson starts and ends with a lesson room. Each lesson has sets of requirements for completing the lesson. These requirements may include the viewing of one or more exhibits, the execution of interactive exercises and completion of previous lessons. The first time that a student enters a lesson room, the system displays the requirements in the B pane. Any requirement that has been accomplished by the student will have the word “completed” follow it in the display. The student may enter the @requirements command to have the requirements displayed to them at any later time they are in the first room of the lesson. (@requirements may be abbreviated to just @req) Figure 3 shows such a display. When a student leaves a lesson the system checks their progress towards completion. If the lesson requirements are satisfied then the MOO announces to the student that they have completed the lesson. (These messages also appear in the B pane.)


enCoreXpress/4.0.1 (VCSU-MEP) - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://euler.vcsu.edu:7000/Xpress_Client/vertical_layout
Google
Login (VCSU-MEP)
enCoreXpress/4.0.1 (VCSU-MEP)
about
help
xpress
look
who
search
my stuff
options
mail
quit
○ normal
○ say
○ emote
Done

This is not a lesson room, there are no requirements.
You view The Relational Algebra...
You have completed this lesson.
You view The Relational Model...
You view Database Management Systems...
You view Language Foyer...
You view C++ foyer...

This is your first visit to this lesson room, here are the requirements.
Requirement 1
 Lesson: Calling Functions
 Lesson: Function Definition Overview
 Lesson: Parameters of Functions
 Room: Types that functions produce
 Lesson: Scope Lesson

You view Function Lesson...
You view Types that functions produce...
You view Function Lesson...
Requirement 1
 Lesson: Calling Functions
 Lesson: Function Definition Overview
 Lesson: Parameters of Functions
 Room: Types that functions produce -- completed.
 Lesson: Scope Lesson

Function Lesson

This is the start of a number of lessons about functions. Functions are the main means to structure the executable code into manageable size pieces. Methods are functions that belong to an object and have the same format and purpose, with slight changes.
Consider the following menu that may be selected by letter or topic:
a) The importance and usefulness of functions
b) Using functions or calling functions
c) Overview of function definition
d) Function parameters
e) Function return values
f) Function and variable scope
or
x) Return to the C++ foyer
You may choose any of these or use 'next' or 'n' to browse them in turn.

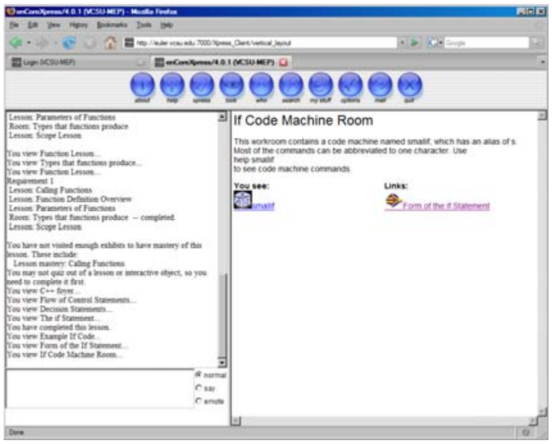
Links:

Why use functions?
Calling Functions
Function Definition Overview
Parameters of Functions
Types that functions produce
Scope Lesson
C++ foyer

Figure 3. A lesson room on the first entry.

In Figure 3 there is one set of requirements. It requires four lessons to be completed and one room to be visited. The first time the room is visited a requirements are shown and this display is in the middle part of the left pane. Later the @requirements command was given and this is shown in the lower part of that pane as well. In this it is seen that the room was visited and so this requirement is complete.

The first word in each of the requirements line is the type of requirement. A lesson is collection of rooms that must be finished as whole. It may either be a prerequisite or subordinate lesson. A room just requires the student to visit the room and read the text in the right hand pane. There are several other types as well including MOOApplets to use, Code machines to describe or trace or Presentations to view. A MOOApplet is a Java applet that the student clicks on and it generates a new window or tab. A Code machine traces through a portion of executable code to show how the statements work. Figure 4 shows a code machine icon found in an exhibit. Figure 5 shows a trace of a code machine. A Presentation shows the slides of Powerpoint Presentation with narration. It will appear in a new browser window or new tab, depending on the web browser settings. Figure 6 shows a presentation and its player.


enCoreXpress/4.0.1 (VCSU-MEP) - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://euler.vcsu.edu:7000/Xpress_Client/vertical_layout
Login (VCSU-MEP)
enCoreXpress/4.0.1 (VCSU-MEP)
about
help
xpress
look
who
search
my stuff
options
mail
quit
○ normal
○ say
○ emote
Done

Lesson: Parameters of Functions
Room: Types that functions produce
Lesson: Scope Lesson

You view Function Lesson...
You view Types that functions produce...
You view Function Lesson...
Requirement 1
Lesson: Calling Functions
Lesson: Function Definition Overview
Lesson: Parameters of Functions
Room: Types that functions produce -- completed.
Lesson: Scope Lesson

You have not visited enough exhibits to have mastery of this lesson. These include:
Lesson mastery: Calling Functions
You may not quiz out of a lesson or interactive object, so you need to complete it first.
You view C++ foyer...
You view Flow of Control Statements...
You view Decision Statements...
You view The if Statement...
You have completed this lesson.
You view Example If Code...
You view Form of the If Statement...
You view If Code Machine Room...

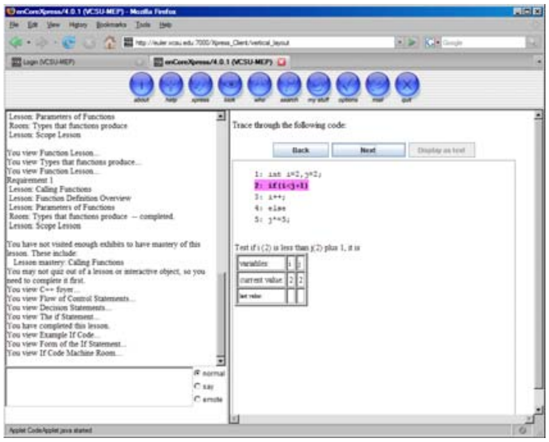
If Code Machine Room

This workroom contains a code machine named smallif, which has an alias of s. Most of the commands can be abbreviated to one character. Use
help smallif
to see code machine commands.

You see: smallif

Links: Form of the If Statement

Figure 4. A Code Machine icon.


enCoreXpress/4.0.1 (VCSU-MEP) - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://euler.vcsu.edu:7000/Xpress_Client/vertical_layout
Login (VCSU-MEP)
enCoreXpress/4.0.1 (VCSU-MEP)
about
help
xpress
look
who
search
my stuff
options
mail
quit
Back
Next
Display as text
○ normal
○ say
○ emote
Applet CodeApplet.java started

Lesson: Parameters of Functions
Room: Types that functions produce
Lesson: Scope Lesson

You view Function Lesson...
You view Types that functions produce...
You view Function Lesson...
Requirement 1
Lesson: Calling Functions
Lesson: Function Definition Overview
Lesson: Parameters of Functions
Room: Types that functions produce -- completed.
Lesson: Scope Lesson

You have not visited enough exhibits to have mastery of this lesson. These include:
Lesson mastery: Calling Functions
You may not quiz out of a lesson or interactive object, so you need to complete it first.
You view C++ foyer...
You view Flow of Control Statements...
You view Decision Statements...
You view The if Statement...
You have completed this lesson.
You view Example If Code...
You view Form of the If Statement...
You view If Code Machine Room...

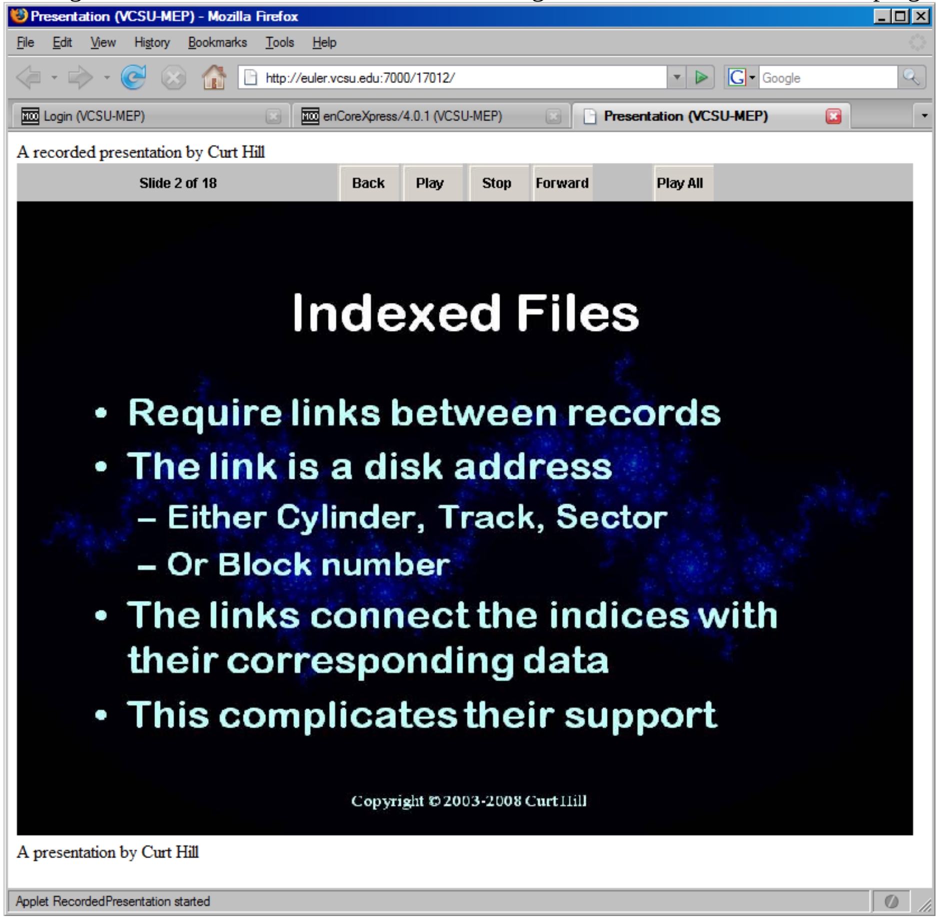
Trace through the following code:

```
int i=2,j=2;
if(i<j+1)
i++;
else
j+=5;
```

Test if i (2) is less than j(2) plus 1, it is

variables	i	j
current value	2	2
last value		

Figure 5. A Code Machine trace in progress.


Presentation (VCSU-MEP) - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://euler.vcsu.edu:7000/17012/
Google
Login (VCSU-MEP)
enCoreXpress/4.0.1 (VCSU-MEP)
Presentation (VCSU-MEP)
Slide 2 of 18
Back
Play
Stop
Forward
Play All
Applet RecordedPresentation started

A recorded presentation by Curt Hill

Indexed Files

- Require links between records
- The link is a disk address
  - Either Cylinder, Track, Sector
  - Or Block number
- The links connect the indices with their corresponding data
- This complicates their support

Copyright © 2003-2008 Curt Hill

A presentation by Curt Hill

Figure 6. Screen 2 of a presentation.

VCSU-MEP keeps track of all the pertinent things that a student has done: rooms visited, lessons completed, presentations viewed, etc. These are compared with the lesson requirements only when the lesson room is re-entered. If the lesson is completed then the student is informed of this completion. If a student asks for the requirements of a lesson that has been completed, the system will only show that this lesson is complete.

Certain lessons come with an out-of-MOO assignment. When a student enters the lesson room after completing the requirements, the MOO states that the lesson has been completed. Approximately 5 seconds later an agent arrives and gives the student an assignment and then departs. This assignment may be described fully by the agent or it may refer the student to a web page where a full description is given. Each student will receive an equivalent but somewhat different assignment. The `@showgoal` command will show the last such assignment.

Agents in VCSU-MEP.

There are several agents that inhabit the system. An agent is a program that interacts with students but takes the form of another user. Agents typically give messages in the B pane and may wait for an answer to a question to be typed in the C pane. Although most agents can be safely ignored, they are only there to help the student navigate the museum.

The first agent is Roxy the Receptionist. She visits immediately after a person enters the system. Roxy asks no questions but usually has a suggestion as to where to go to continue work on the current lesson.

Lonnie is an agent that looks for and assists students who are lost. A lost student is one who is not in a room that is relevant to the lesson that they should be working towards. Lonnie will enter a room with a lost student and ask them if they wished to be move to a room that might further them towards their current lesson goal. This is a yes or no question, but if Lonnie is ignored for a while, he will leave.

Aimee, is like Lonnie, but looks for aimless students. An aimless student is wandering in a relevant lesson, but is not making progress towards the goal of completing the lesson. Like Lonnie, Aimee will ask if assistance is needed and move them to a better location.

There are a number of agents that cannot be ignored, even though they ask no questions. These are the roving goalies and have various names. When a student completes a lesson and that lesson has an assignment connected with it, one of these agents drops in to deliver the assignment. This assignment is now connected with the student. The @showgoal commands will display the assignment.

There is substantially more to know on using VCSU-MEP, but most of it should be learned by experiencing it. On the next page is a short discussion on the basic commands. These are placed in the lower left hand pane.

Commands concerning goals and lessons.

There are several commands that help students get some information on the course. These should be typed into the C pane of the browser.
`@course` – shows the lessons that need to be completed and their goalies
`@grade` – If your course has an attached grade book it will show your scores.
`@progess` – shows all event, goal and award information.
`@requirements` – May be abbreviated to `@req`. This tells the student what they have completed and what yet needs to done to satisfy this lesson.
`@return` – Return to the last visited room.
`@showgoal` – Shows the last assignment given upon the completion of a lesson with an out of MOO assignment.
`@showgoal# N` – Shows the Nth assignment given upon the completion of a lesson. The @progress command will show all previous goals. N must be an integer.