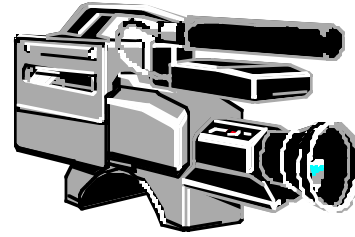


CS 2574 Fall 97

Project Specifications

Cinema Critic Comments



Cinema Critic Comments (**CCC**) is an interactive software tool that would be used by movie critics like Siskel and Ebert. The program is designed to allow reviews of specific movies to be created, deleted, browsed, and allows reviews to be grouped into categories. CCC is menu-driven single-mode system. The user will scroll a selection bar to a desired data line and then choose a pull-down menu command to operate upon the current selection.

The discussion that follows describes the CCC interface and operations in detail. Very minor deviations from these specifications may *not* result in grade penalties. However, developers would be well advised to receive prior approval before altering any of these requirements.

Discussion

CCC will begin execution with a startup screen giving the usual information about the system and developer, along with the current product version number. The startup screen may be presented in a clever manner in order to capture a viewer's attention. After the user hits the return key, the startup screen will clear and the CCC screen will be displayed, (see CCC screen layout, Figure 1). The command line parameters are then accessed to open the optional *category.ccc* file. The *category.ccc* file name, if present, follows the CCC program name on the operating system command line: "*ccc category*". A file existence check for any file opened is required and an error message is displayed if the file is not present in the current directory. The user may alternatively elect to start CCC without any command line parameters. In this case the user must first open an existing *category.ccc* or initial *vrs* files before most CCC menu commands become available. (For an explanation of the *vrs*, *mov* and *rev* file formats see: <http://ei.cs.vt.edu/~cs1704/reviewer.pdf/>). The remaining discussion will be made with regards to the areas of the CCC screen layouts and how the system execution affects them.

The CCC screen will be divided up into several areas: the menu area, display area, I/O area and the status area. The contents of these areas may change, but the four areas will be maintained on the screen at all times, after the startup screen has cleared and until the CCC quit operation executes. The menu area will hold the names of the pull-down menus available to the user, containing all CCC system commands. The display area will be used to show various movie review information dependent upon the user's request. The I/O area is employed by CCC to give error/ warning or prompt messages and to accept keyboard input. (All input from the I/O line, unless explicitly noted, will be buffered, (i.e. `cin >>`). All other input, [menu selection and scrolling actions], will be unbuffered.) The status line will contain 3 fields: 1. the name of the current open CCC category file, left justified; 2. the CCC system name, centered and **inversely displayed** ; and 3. the current movie category and review file name, (dependent upon the display area contents), almost right justified. Note: most terminals will automatically scroll one line whenever a character is displayed at the lower right corner, (row 25 column 80), so care must be taken to ensure that this auto-scrolling does not affect the CCC system display if line 25 is used.

When the display area contains category data, (i.e. the categories' review listing), a **inversely displayed selection bar** will be provided, that the user can position over to highlight the desired data line that they wish to perform some menu command upon. They may change the "current data line" by moving the display area inverse video selection bar, vertically using the $\uparrow\downarrow$ cursor keys, one line vertically up/down at a time. When the selection bar is located at the top or bottom of the display area and the user depresses the $\uparrow\downarrow$ cursor key respectively, the display area contents will scroll one line. The line scrolled into the display area becomes the current highlighted data line. If the inverse selection bar is positioned at the

top or bottom of the data list, (i.e. at the head or the tail), and the user attempts to scroll beyond the list bounds this should cause a bell to sound or the screen to flash and an appropriate error message to be displayed on the I/O line. In addition to the single line scrolling just described, CCC will provide two other display area 'browsing' methods. The user may also scroll up/down a 'page', [a page in this context is defined to be the number of data list elements shown in the display area, approx. 20], at a time by pressing 'U' or 'u' / 'D' or 'd' respectively, (the PgUp and PgDn keys may be used instead). CCC will also allow the top or bottom of the data list to be moved to directly, by the user depressing either the 'T' or 't' / 'B' or 'b' keys, respectively, (the Home and End keys may also be used). The display area should never contain blank lines for data lists with ≥ 20 elements. Consider a list with 100 elements, if currently elements 11 through 30 are displayed and page up is hit, the first 20 elements in the list should now be on the screen. Similarly, if elements 71 through 90 are displayed and page down is hit, the last 20 elements (81..100) are then displayed. For files with < 20 elements, page up/down would receive an error message and line up/down should not cause any line/element to scroll off the display. Display speed should not be an over-riding factor for implementation of these browsing actions.

Areas		CCC Screen Layout					Line	
Menu		Program	File	Edit	CCC	View	Search	1
		Title		Rating	Filename			2
		Alien			3.5	alien.rev		3
Display Area		Aliens II			4.0	aliensII.rev		
		Back to the Future			3.5	backtoth.rev		
		Beneath Planet of the Apes			2.0	beneatht.rev		
		Blade Runner			4.5	bladerun.rev		
		Clan of the Cave Bear			2.5	clanofth.rev		
		Cocoon			2.5	cocoon.rev		
		The Empire Strikes Back			3.0	empirest.rev		
		Enemy Mine			2.5	enemymn.rev		
		Forbidden Planet			3.0	forbidde.rev		
		Journey Home			2.5	journeyh.rev		22
I/O								23
Status	Siskel	Cinema Critic Comments			Science Fiction			24

Figure 1.

Menu Commands

It is now time to explain how the pull down menus are to be accessed and how they will behave in CCC. Following this menu operation explanation, the rest of this section will discuss each pull down menu and each of the commands that they contain. A user will activate a desired menu by hitting the first letter of the menu, [**P** , **F** , **E** , **C** , **V** , **S**]. This will cause the corresponding menu and its commands to be displayed with the first command inversely displayed on a selection bar. The user may move the selection bar up/down the menu using the \uparrow \downarrow cursor keys, with the bar wrapping around from top to bottom and vice-versa. The user may choose a command by hitting the return/enter key when the bar is positioned over the desired command, which will of course cause the command to be performed. The user may also elect to NOT choose a command and exit the menu without selecting a command by hitting the delete/backspace key. After a user has chosen, or decided not to choose, a menu command the menu should be erased from the screen and the display area restored to its previous contents.

Not all menu options are functional all the time. Whether a particular option is functional is dependent upon the current contents of the display area. Options affecting review listings, (see figure 1), should not be operational when the category list is currently in the display area and vice-versa. For example, the Edit menu Category command would not be functional when the review listing shown in figure 1 is displayed. If a user selects a non-functional menu option at any time a bell and/or error message on the I/O line should occur. (Optionally the non-functional options could be disabled, not allowing the user to select them. Or non-functional

options could simply not be displayed on the menus when they are inactive. Both of these approaches are optional and require a more sophisticated menu implementation than required.)

Prog

The **Program** menu contains only two commands, About and Help. The About command will display the initial startup screen and will be removed after the user hits the return key, restoring the display area contents. The Help screen will present a brief one screen overview of the CCC system and will also be removed after the user hits the return key, restoring the previous display area contents.

Prog
About
Help

File

New will prompt the user for the name of an initial vrs file. The category.vrs file is opened and processed. This involves building a double-linked list of all of the movie and review files data derived from the category.vrs file, ordered upon the titles. This list is identical to the movie review set, (mrs), built in the Movie Review Manager program with one exception. (For specifications of the Movie Review Manager program see: <http://ei.cs.vt.edu/~cs1704/revman.pdf/>). As categories, (movie sets) are encountered they should be assigned a unique integer index which is also assigned to all of the reviews contained in the category. (Duplicate entries of course should not exist in the movie review set list. Review files, listed in multiple movie files, should only be stored in the movie review set list once.) The Open command will prompt the user first to see if they wish to close the current open CCC file, if one is open. (If the user decides not to close the current CCC file then the file menu is exited with no action taking place). If the user does wish to close the current CCC file then the close command is automatically executed and the user is prompted for the name of an existing CCC file to open. At this point the CCC file is read and the list data structure is created. The Save command simply writes the current contents of the internal CCC nested-list structure onto the disk in the CCC file format, (identical to the mrs format), in the current directory, over-writing any other version of the file that might exist in the directory. The save As command operates almost the same as the save command, except that the user is prompted for the name for a new CCC file, which is then created and written to the disk, becoming the current open CCC test file, (causing a status line update). The Close command will simply clear the display area and destroy the CCC list structure, (ideally it would be nice to inform the user if they have not saved the file if changes have been made and give the option for doing so, but this is not required). The Quit command will prompt the user to confirm that they wish to exit CCC, (☺), destroy the list structure and close any open files, before clearing the screen and returning to the operating system.

File
New . . .
Open . . .
Save
save As . . .
Close
Quit

Edit

The **Edit** menu will provide a simple mechanism for a user to change the category/review information. Changes will require an update of the list contents and the display area. The Category option prompts the user for a category name to replace the current highlighted category, (movie set), in the display area. The Title option prompts the user for the a title to replace the current highlighted review in the display area. Likewise the Rating option prompts the user for the a rating to replace the current highlighted review. The Filename option prompts the user for a filename to replace the current review filename in the display area. Note that the Filename option will not be active when categories are displayed as CCC will not maintain or store filenames for categories, (i.e. mov filenames).

Edit
Category . . .
Title . . .
Rating . . .
Filename . . .

CCC

The **CCC** menu contains options for a user to manage the category and review entries. The Add category will prompt the user for the name of a new category, set the initial size to zero and assign a movie index, internally. The Include review option will allow a user to enter a title for the movie, review rating and associated rev filename. The review is

CCC
Add category . . .
Include review . . .
Delete category . . .
Erase review . . .

automatically included in the currently displayed category review listing and assigned the index of the current category. The Delete category option will prompt the use for confirmation of the command and then will remove the highlighted category from the CCC list and also recursively remove all of the review entries in the category. The Erase option will remove the currently highlighted review. If selected in Figure 1 it would remove the Alien review from the CCC list and hence from the SciFi category.

View

The **View** menu will contain only three commands. The Categories option will allow the user to switch from viewing a particular category's review listing to a display of all of the current CCC categories. Analogously the Movies command will display currently highlighted category's review listing. The Review option will display the review file contents for the currently highlighted review entry. The first five lines of the review file will be labeled and displayed. The user will then be prompted to view the full movie review, displayed 20 lines at a time with the user prompted after every 20 lines to continue. After the last line of the review file has been output the display area will return to showing the previous category's review listing.

View
Categories
Movies
Review

Search

The **Search** menu contains only inactive (future) commands. The Search options will not be implemented in this version of CCC. If the user selects any Search option, a brief message is displayed informing them that these are future enhancements to CCC.

Search
Name . . .
Title . . .
Filename . . .

Whenever an CCC action will take more than a couple of seconds to execute, a brief specific message should be displayed, followed by "Please Wait...". An attempt to activate an illegal or invalid operation should cause a bell to sound or the screen to flash and an appropriate error message to be displayed on the message line.

Text Windowing

The use of a text windowing package is NOT required for this program, but is recommended. The ability to define multiple windows to refer to the same screen area will greatly aid the task of screen content management. The restoring of the display area's contents after displaying pull-down menus and switching between different display area contents will be made simple. The text windowing facilities will greatly enhance the professional look-and-feel of the project.

For students coding a non-text windowing interface, the interface specifications described above must still be adhered, with the following modifications. The display area will be lines 1-22, the menu bar on line 23, the status area on line 24 and the I/O area on line 25. When a user selects a menu, the complete screen should be redrawn with the menu area line, (23), replaced by the menu commands. The user will be allowed to select a menu command by hitting the first letter of the command. The lines in the display area should be numbered to allow a user to select a line's contents for operation upon by a selected menu operation. A non-text windowing interface will require the creation of a window ADT and menu ADT for controlling the screen contents.

A specification addendum for this project may be forthcoming shortly if required. The addendum will describe minor enhancements or changes to CCC.

PROJECT GRADING and DOCUMENTATION

Successful completion of this software development and implementation project will result in the following External and Internal Specification Documents for submission:

1. A complete **STRUCTURE CHART** of the system's design, including all interface specifications. The structure chart must be of sufficient detail to communicate to others WHAT functions are needed to implement this system and how the various parts are interconnected. The structure chart is complete if the algorithms for each module could be developed from it, along with a Data Dictionary, although no data dictionary is required. Use only 8 1/2" X 11" sheets of paper to record the chart, DO NOT use larger sized pages. Pages must be bound or stapled together. Use the symbols described in class for the components and the interfaces. Legibility counts and points will be deducted for sloppily presented work. Careful and CORRECT notation is also required and points will be deducted for a poorly annotated chart. The final chart will be compared with the initial chart to determine the amount of modification made to the original system design. Penalties will be incurred according to the amount of modification. Note — calls to standard C/C++ system routines: cin, cout, etc. need not be annotated on the chart.

2. An **INTEGRATION PLAN**, that is a plan for HOW and WHEN the various phases of the project will be developed and which modules will be added at each integration point.

a. THREE integration points must be given. The points reflect a semi-equal division of the project work. (Specifying only 2 modules to be completed at the first 2 points, while delegating the remaining, approximately 90% of the project for the last point would be unacceptable.)

b. For each integration point, you must name the TASKS that will be completed by the identified date and clearly define WHAT BEHAVIOR can be expected to be exhibited by your program at that point.

c. The three integration points for the system are: 1. 9/29/97 2. 10/29/97 3. 12/3/97. Each student will be expected to demonstrate to a GTA the behavior of their project second and last integration points. Failure to meet with the GTA for the demo will result in point penalties.

d. At any integration point or when the program is submitted, any non-functioning commands, actions, etc. of the system are not expected to bomb the system. A message should be displayed informing the user that this operation has not currently been implemented or is still under development. On the final submission, all non functioning aspects of the system MUST be accompanied by a short explanation describing the suspected problem(s).

3. A **USER MANUAL** generated by the word processor of your choice must be written. The manual should enable the NOVICE CCC user to access all the features of the system. This document should be at least 6-8 pages in length, excluding the title page, table of contents, index, etc. The user manual must be written from the perspective of a novice CCC user, not from a programming implementation point of view.

QUALITY OF CODE

It is expected that the code will be WELL-DOCUMENTED, appropriately indented and VERY READABLE. Points will be deducted for poorly presented code. Individual functions, excluding their documentation and declarations may be no longer than one page. Each compilation module should contain only related functions. Function headers should resemble the following template:

```
(***** )
(*FUNCTION NAME *)
(*DESCRIPTION OF FUNCTION *)
(*DESCRIPTION OF ALGORITHM : FUNCTION IMPLEMENTATION *)
(*)
(*CALLED BY: (LIST OF FUNCTIONS) *)
(*CALLS: (LIST OF FUNCTIONS) *)
(*)
(*PARAMETERS: NAME AND ROLE IN ALGORITHM OF EACH *)
(*)
(*AUTHOR: name of author *)
(*REVISIONS: DATE, REASON *)
(*)
(* AUTHOR FOR EACH (if different) *)
(*VERSION: x.xx *)
(***** )
```

Each of these documents must be placed in ONE binder or folder. Each separate document must be clearly labeled, bound and separated from the others. The diskette(s) containing the code must be inside and securely fastened. ALL WRITTEN MATERIAL MUST BE EASILY READABLE AND THE PAGES SHOULD BE EASY TO TURN IN THE BINDER. ILLEGIBLE WORK WILL NOT BE GRADED AND YOU WILL RECEIVE A "ZERO" FOR ANY SUCH PRODUCT SO REJECTED. Failure to conform with these standards will result in an immediate loss of points.

The following scale will be used for grading:

PRODUCT	PERCENTAGE OF PROJECT GRADE	PERCENTAGE OF FINAL GRADE
Structure Charts:	20	5.00
Integration Plan:	10	2.50
Code/Execution:	60	15.00
User's Manual:	10	2.50
Total:	100%	25.00%

Due Date schedule:

There will be no extensions or late submissions for this project! Any project or portion thereof NOT submitted on time will be rejected!

<u>PRODUCT</u>	<u>DATE</u>
Structure Chart:	11 - 14 - 97 (Initial Version)
Structure Chart:	12 - 03 - 97 (Final Version)
Integration Plan:	12 - 03 - 97
Code/Execution:	ALL code is due by 5pm on 12- 03 - 97. • D-Day •
User Document:	12 - 03 - 97.

The following deliverables are due on December 3rd:

- a. Hardcopy and softcopy LISTING for all code.
- b. Diskette(s), (system labeled: WIN 95 or NT), with all source code, project files and a single executable image. The executable image must run by typing **ccc**. The files on the disk may be zip compressed in self extracting archives and file protection must be set to allow anyone access. Only 3 1/2 inch DOS (FAT)1.4 MB disks are to be submitted. No "extra" files (other than project source files, I/O, data files, structure chart and user manual files) should be on the diskette(s).
- c. A hardcopy and softcopy of the user's manual. The softcopies (2) should be in the word processor format used to create the manual and also in standard ASCII text format.
- d. Modified final up-to-date Structure Chart + the initial design version.