Personal CAL Workbooks: The next chapters

S.C.Maddock

s.maddock@dcs.shef.ac.uk

Dept. Computer Science
University of Sheffield
Regent Court, 211 Portobello Street
Sheffield S1 4DP
U.K.
Fax: +44 742 780972
Phone: (0742) 768555 x.5592

1. Introduction

Traditional lectures can tend to be rather passive beasts, uninspiring and sometimes even sleep-inducing. (Of course, not all lectures are like this!!) Learning must be active: *to do is to learn*. However, lectures are cost effective, whereas individual attention to students is not, especially as class sizes increase. How can we reconcile these things? Perhaps Computer-Aided Learning (CAL) is an answer. The CAL system would replace the exposition of the lecture and a student would work at his own pace, mapping a journey through a graph of knowledge. In a system of lectures, tutorials and assignments, the majority of lectures could then be replaced with the CAL system. Lectures could still be retained for parts of the course, so that the student has a 'total learning experience'. With less lectures, more attention would be focused on the use of tutorials which could be used to promote group (here I make a distinction between groups and teams, teams being task-oriented - see [ROWL89] and [ABER93] for a discussion) interaction, discussion and reflection on ideas learned from the CAL system.

Of course, this assumes that a suitable CAL system is available. Otherwise one would have to be developed, which would be an expensive undertaking. Or would it? MADD93b and MADD94 report on the idea that students themselves can be used to develop CAL software for their own courses. One year's students can develop a CAL system for the next year's students. The students developing the initial system are the manpower that combats the time taken to develop CAL systems. Such a project could be ongoing, with each year's students in turn adding to the system.

The rest of this paper will discuss the results of the first year of employing such a strategy to develop a CAL system for teaching computer graphics. The only other CAL system for teaching computer graphics that I am aware of is Hypergraph [OWEN92]. However, this differs from graphiCAL (the working name for the complete CAL system) in that students cannot easily reconfigure the existing system and add new content without resorting to programming within the Guide authoring environment. Since graphiCAL is itself a constrained, graphical authoring environment based on a book analogy, authoreaders ("the reader is the author" [MADD94]) can use icons to manipulate chapters, sections and pages, can copy information to pages from word-processing software or from drawing and painting packages, can construct pages containing multiple choice questions and can link to executable programs. GraphiCAL is a "personal CAL Workbook" [MADD94], since it is intended to be each student's personal computer-based instructor and notebook for the module.

2. A Plan

Single Honours Computer Science students at Sheffield follow a three year degree programme, with each academic year being composed of two semesters, each of 15 weeks duration. (The semester system actually begins in September 1994 and takes over from the traditional academic year composed of three terms of 10 weeks.) Each semester is split into 12 weeks teaching time and 3 weeks of exams. A student concurrently takes six half-modules in each semester, and each half-module involves 30 hours contact time (normally 20 hours lectures and 10 hours tutorials) with assignments extra. Students can opt for three computer graphics half-modules, the first of which is Computer Graphics I (CG I) which is an introduction to computer graphics taught in

semester 2 (with a half-module on Human-Computer Interaction, taught in semester 1, being a prerequisite).
The text books for CG I are FOLE90 (to be replaced by FOLE93 in 94/95) and HILL90, and the syllabus for CG I is split into the following three sections:

- Graphics environments: components of current systems; input/output devices; graphics standards and languages;
- Basic two-dimensional techniques: linear transforms; coordinate systems; the two-dimensional viewing pipeline; clipping; interactive techniques - pick-and-drag, rubberbanding; raster techniques - line drawing, region filling, halftoning;
- Applications of two-dimensional graphics: image processing; computer-aided design (CAD) - parametric curves; modelling natural phenomena - fractals, L-systems; animation - interpolation, morphing; scientific illustration and simple visualisation.

CG I is populated predominately by second year undergraduates, some of whom are scouting the subject to see whether they should take the advanced final year half-modules in computer graphics (the imaginatively named Computer Graphics II and Computer Graphics III, which are sourced by WATT93 and WATT92 respectively).
Within this framework, a 4-stage plan for developing a CAL system for CG I was developed [MADD93b; MADD94]:

Stage 1. Building a base of enthusiasm (1992/93)
The goal of this stage was to establish a courseware repository, populated from a range of sources including software created by students as part of their practical work. This was needed for stage 3.

Stage 2. Development of a navigational framework (May-Sept 93)
Here, a tool (called HyperBookShell) was developed in order to enable CAL systems based on a simple book analogy to be easily created using graphical interaction [HOPK93].

Stage 3. Development of personal CAL Workbook (1993/94)
Students, working as part of 4-person groups, used the tool from stage 2 along with the courseware repository to construct team CAL Workbooks for the module they were taking, namely CG I.

Stage 4. Rationalisation (Summer 94)
The aim here was to turn the CAL Workbooks of 93/94 into a single CAL Workbook (graphiCAL) that would be given to each student on the CG I half-module in 94/95.

3. Realisation

Stages 1 to 3 are now complete. Stage 4 is however being delayed until summer 1995, so that stage 3 can be redone by the students on CG I in the 1994/95 academic year. There are a number of reasons for this, the main practical ones being time and disc space considerations. The original plan for stage 3 was for each team of (approximately) 4 students to develop a complete CAL Workbook for CG I. However, this would either have been prohibitively time-consuming or would have resulted in a rushed and superficial exercise. I thus decided that each team would develop a single chapter of the CAL Workbook, based on the chapters of the lecture notes for the module. Teams would also review each other's chapters as part of the assignment. The chapters would then be combined to make graphiCAL, the complete CAL Workbook.
We can consider the complete exercise as a number of stages, starting with the environment for the project and ending with assessment of both individual and team performance.

3.1 The environment for the assignment

University lecture rooms are generally set up for the traditional lecture with tiered seats arranged so as to focus attention on the lecturer. Even flat lecture theatres are not set up for discussion, as students have to twist uncomfortably in immovable seats. Thus, the traditional University environment is not conducive to discussion-based tutorials. This can be termed the institutional context to the problem. The institution must confront the problem and modernise lecture and tutorial environments so as to promote active learning and group-based work. A flexible arrangement is needed that can be reconfigured to promote groups and teams working in small circles of discussion.

3.2 Explicit instructions

The constraints for the assignment must be considered carefully. It needs to be made clear what is expected of the students not only in terms of assignment deliverables but also in their behaviour in group work. The aims and objectives for the assignment must be made explicit since this helps the students to understand the lecturer's thinking and also helps them to plan their time effectively. Here, a distinction is made between aims and objectives, with aims being quite general and few in number and objectives being measurable and many in number. For example, in

an introductory programming course, a general aim might be to provide a thorough grounding in the techniques of software implementation, whereas objectives would be to use stepwise refinement to divide a problem into simpler subproblems and use a structure chart to show the relationships among these subproblems or to write procedures with parameters.
Part of the handout that was given to the students for the practical assignment for CG I is included in Appendix A. Note that it explicitly states what the deliverables for the project are. This was followed by nearly all the teams that worked on the project and certainly made it easier to break the marking down into a series of stages. However, it is not so explicit on the aims and objectives of the project. This was covered to a certain degree in the introductory tutorial when the assignment was handed out but needs to be made more explicit in the handout. Likewise, more information is required in the marking section.

3.3 Team formation
Abercrombie [ABER93] draws a distinction between teams and groups. A team has a "more or less tangible end product, or at least has an easily understood aim". It is interesting to draw this distinction since it affects the way that the project is conducted. A team usually has a leader and a set goal, then makes a plan and works through it within the set time limits. Individuals can contribute their work tasks to the team whole and still remain individual. In a group-based situation, reflection and discussion are more important than producing a tangible product in a limited amount of time. Perhaps these two views are extremes and the real situation is that there is a mix of the two. However, for the purposes of this project I regarded the class as split into teams.
On this year's CG I module there were 41 second year Single Honours undergraduates, 11 third year Dual Honours undergraduates and 24 mature, overseas MSc students (who were on a one year MSc conversion course in Software Systems). In order to avoid 'cliques-and-leftovers', and to give more of a 'renaissance' feel to each team, I decided on the formation of the teams. Thus each team contained approximately two second year Single Honours undergraduates, one third year Dual Honour student and one mature, overseas MSc student. Initially there were 19 teams (although because of some students deciding to switch module options and also problems with some students not attending, thus leaving a team understaffed, this was reduced to 17 by the end of week 2).

Each team was given a chapter of the CAL Workbook to write (based on the chapters of the lecture notes for CG I). With seven CAL chapters to author, this resulted in teams developing alternative versions of the same chapter. The aim was to use this fact productively at a later stage in the project, with teams reviewing each other's equivalent chapters. However, part way through the project, this was changed so that the teams each reviewed a different chapter to the one they had worked on. This was for two reasons. First, a number of teams had expressed concern that their work might be copied by another team working on the same chapter, especially if the team had been given a copy of the system to review. Second, it was hoped that each team would learn more about another topic on the course if they reviewed a completely different chapter.
Many of the MSc students commented favourably on the assignment, stating that it was the only assignment where they had worked together with undergraduates and that they had enjoyed it. Some of the undergraduates stated this too. Both said how they felt that each had brought different skills to the team. However, there were also a few teams where it didn't work. Some of the undergraduates felt that the MSc students weren't as adept as them at programming and computer graphics. They failed to realise the other skills that these students had. One other point here is the fact that nearly all the undergraduates were male, whereas approximately half the MSc students were female. In some cases this meant one overseas female student in a team with three male undergraduates. However, in one team this worked extremely well with a quiet, mature, overseas, female student commenting that she had learnt a lot from the other team members and had thoroughly enjoyed the exercise.
The mixed teams meant that there was a lot of useful social interaction in the assignment. This of course had been engineered in the formation of the teams. However, the idea of a 'community' is perhaps something that should have been made more explicit and stated as a general aim of the assignment. This might then have helped in those few teams where the mix didn't work.

3.4 Using tutorials to review progress
Many of the tutorials allocated for CG I were used to give advice on different stages of the assignment and to review progress. The first tutorial of the module was used to set the scene for the assignment and to organise the teams. Although cramped by the lecture room layout, we were able to set teams up in small circles of discussion. The students were then asked to sit in their teams for every tutorial session thereafter.

This meant that they got to know each other through their discussions on the assignment and through work on other tutorial exercises.
In the first tutorial session, I forced each team to pick a leader. This was for practical reasons. Each team leader's network userid was used as the location for storing the most up-to-date version of the assignment so that I could easily find and retrieve it. Some teams chose a leader fairly quickly. Others chose to delay the decision to the next tutorial. Perhaps more time should have been spent on the practical problems so that a leader wouldn't have been required. I tried to emphasise to the teams that the 'leader' was just the person who would look after the final copy of the system. But perhaps the teams saw it as a leader who would be responsible for the team. That person was then different. He or she was 'responsible'.
The tutorial in week 2 featured a demonstration of the authoring tool for the assignment (HyperBookShell [HOPK93]) along with a demonstration of some of the programs in the module repository. This provoked a lot of interest in the assignment and the subject itself, a finding that reflects comments made by Schweitzer [SCHW92] when he used interactive computer graphics software in his classes. Design was considered in tutorial 3 and evaluation of CAL systems was considered in tutorial 8 in week 8.
Week 7 saw the feed-in and feed-back loops shown in figure 1 being exercised. An extra laboratory session was arranged during which the CG I teams demonstrated their work to volunteer first and third year undergraduates. The first years could feed in what they thought should be in such a module, whereas the third years could feed back the knowledge they had acquired when they had done the module in their previous academic year. The only real problem with this session was the lack of first year volunteers. 14 third year students volunteered quite readily, although this may have had something to do with the fact that I was also teaching them in the third year advanced computer graphics modules!!

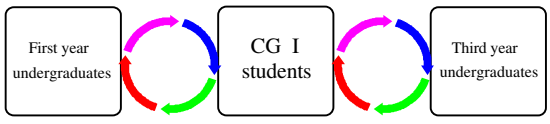

Fig. 1. CG I students test their products on those that have not done the module before (first year undergraduates) and those that have (third year undergraduates).

In those tutorials that were used to check progress, more time was spent on teams that were having problems. This was mainly because of time problems. It was impossible to visit 17 teams in the fifty minutes allocated to the tutorial. One solution here may be to split the class into two divisions of teams and see them every other week.

3.5 Evaluation of CAL systems

The assessment of CAL systems is still in its infancy. We can draw upon HCI literature to provide assessment criteria for the interface, we can draw upon software engineering literature to provide some criteria for judging the design of the system, and we can compare content with the course text book or the course notes. However, how can we evaluate the educational value of the system? What makes a good CAL system? This of course relates directly to the question of how to assess the effectiveness of any teaching system. Can this satisfactorily be turned into a number? Should it be turned into a number? These are *big* questions.
For this assignment, the assessment was broken down into three parts: the Workbook (60%), the interactive Modula-2 program (20%) and the report on the evaluation of another team's system (20%) (see Appendix A).
For the Workbook, the following were considered: the user guide (5%), the design & structure of the system (10%), presentation & content of text, pictures and questions (35%) and the resources researched and used in the system, e.g. text, pictures, interactive programs (10%). Since the design was constrained by the linear nature adopted in HyperBookShell, relatively few marks were apportioned for this section. In assessing the design increasing marks were awarded for including design diagrams, explaining the diagrams, considering the user in the balance of the system and considering what resources would be required and alternatives if they proved not to be available. The majority of the marks were reserved for the content. This was done as a series of bands of marks. The first band of 0-22 out of 35 was for a Workbook that just used the lecture handouts as the source material and missed out some sections or structured them poorly. The next band was 23-27 and here something extra to my notes was needed, 28-32 was for adding something interesting to the previous band and 33-35 was for excellence.
In considering the structure of the content, it was expected that there would be one main point being made on each page of the Workbook. Pages with too much detail were penalised. The regular use of questions was also considered in the mark banding. Out of the 17 teams, 8 fell in the top two bands, whilst the other 9 were in the lower two bands. This seemed to indicate that the marking had differentiated fairly well across the teams. For the resources portion of the marks, the use of a variety of text sources and scanned images was

considered along with the number of interactive programs found to illustrate parts of the Workbook.
For the program, the following were considered: explanation in the CAL system, the interface, illustration, interaction and maintenance of user's interest. Explanation attracted a maximum mark of 5%, whilst the other factors were combined in a set of marking bands: 0-10, 11-13, 14-15.
Finally, the evaluation report was considered. This was the team's evaluation of another team's system. Here, a structured review was expected which looked at a number of aspects of the system. The students were told to use HCI literature (from their HCI module in semester 1) to judge the interface, to look at the content coverage (based on the CG I lecture notes), to judge the effectiveness of the multiple choice questions used throughout the CAL system and to review whether the system used a balanced and interesting set of resources. Again, bands were used to mark this section: 0-10 was for a 'bitty' report with little structure; 10-12 was for some structure, but limited scope; 13-16 was for a structured report with a range of viewpoints, some of which were in-depth; 17-20 was for a well-structured report that was wide-ranging and offered constructive suggestions for improvement. The evaluation reports were the weakest part of the assignment. Many of the teams left this to an individual team member to do and approached it in a very unstructured way. This is definitely an area where more guidance needs to be given.

3.6 Student self-assessment
Each team wrote a self-assessment document, comprising the following two parts (see Appendix A for more details):

- a team assessment sheet
- individual self assessment sheets for each team member

These assessments revealed (or in some cases confirmed) some problems. How should the marks be distributed amongst the team members when some members had contributed more than others? In the laboratory work I had overheard some conversations along the lines of "[paraphrased:] I'm doing all the work and the others are useless". Splitting the self-assessment stage into two parts also caused problems for some of the teams. One student commented that she had enjoyed the assignment but that now a number of teams were squabbling about how to apportion their marks. In some teams the self-assessments (which could be handed in separately) didn't quite match the team assessment.
Out of all the teams, 8 opted for sharing the marks equally and 9 asked for certain members marks to be reduced because they had done very little, or increased because they had done more work than the others. Of those that asked for unequal distributions, some of the team assessments were in disagreement with the individual assessments. In general, I used the team assessment document to apportion the marks. However, in a few cases, the individual assessments and my knowledge of the progress of the team throughout the semester were used to change the marks slightly. There were no complaints about this from the students.
These leaves many questions. Even if the team product can be assessed, how should the individuals in a team be assessed? This year the students themselves decided. This could be backed up in future by more use of progress checks, perhaps by the students themselves, or could be improved by using a more detailed self-assessment document.
One final note about the self-assessment documents was that the students were quite honest in answering the questions. The fact that they told me what the main weaknesses and areas for improvement were made some parts of the marking somewhat easier.

Finally, for this assignment, it is worth stating that the process is as important as the product. The students did learn something about how to (i) assimilate information and then structure it for others, (ii) devise assessments and thus learn how to assess others, (iii) work in teams and thus foster management skills, and (iv) react to user feedback during product development [MADD93a].

4. The Next Chapters

The assignment was successful in that it involved the students in their course more. It encouraged them to share their knowledge through team discussions and through reviewing the work of other teams, and it also facilitated the mixing of students across different degree years. It was an active learning experience for both myself and the students.
For next year's CG I module, the assignment will involve the students reviewing this year's work, before working in teams to develop similar chapters again. The review will also encompass a review of Hypergraph [OWEN92] since it covers similar material. Stage 4 of the plan will thus be realised and a complete graphiCAL can then be constructed.

GraphiCAL is an example of using graphical construction for developing CAL systems. However, we are also considering the use of mark-up languages such as HTML (which is used for creating documents on the World Wide Web). We are currently investigating a product called IDC [LEED93] which will convert marked-up documents into Asymetrix Toolbooks. This may enable us to use marked-up documents for HyperBookShell and thus solve one of it's major problems which is that the CAL Workbooks that are created using it are very big. This is because HyperBookShell is actually stored with the CAL Workbook that is created using it which is akin to storing word-processing software with every file that is created using it. This is obviously not a good idea. We need to consider mark-up languages because it is not yet clear which system and technique will prove the best for creating a CAL system for teaching computer graphics. For instance, it may be better to consider a modular system, where a package of module topics can be chosen from a wider range. The work of Lilly et al [LILL93] could prove fruitful here.

The final aim is that each student taking CG I will be given a personal copy of graphiCAL at the start of the module. Ideally they would also be provided with a portable computer. They would sign up for CG I by collecting graphiCAL and any other associated material (via network or CD-ROM) and then using it as core learning material, adding new information to it as the module proceeded. In subsequent tutorials, the teams could then become groups, discussing their work on their personal CAL Workbooks. A related question here is whether such groups would ever have to meet physically. With modern technology, a virtual campus could be created. This would be similar in style to the Open University, but would be done via computer networks, e-mail and teleconferencing.

Of course, there are still many questions that remain unanswered: What makes a good CAL system? Can (and should) lectures be replaced with CAL systems? Can CAL help in an educational system that is taking on ever increasing numbers of students? How can we evaluate such things? I hope that by continuing with the development of graphiCAL and continually involving the students in the process (which I believe is very important) that I will eventually be able to answer some of these questions.

References

ABER93 Abercrombie, M.L.J., "Selections from the work of MLJ Abercrombie", ed. J.Nias, SRME and OUP, 1993, pp.69-95.

FOLE90 Foley, J.D., A. van Dam, S.K. Feiner and J.F. Hughes, "Computer Graphics: Principles and Practice (2nd edition)", Addison-Wesley, 1990.

FOLE93 Foley, J.D., A. van Dam, S.K. Feiner, J.F. Hughes and R.L. Phillips, "Introduction to Computer Graphics", Addison-Wesley, 1993.

HILL90 Hill, F.S., "Computer Graphics", MacMillan, 1990.

HOPK93 Hopkins, J., "HyperBookShell - CAL framework", MSc dissertation, Dept. Computer Science, Univ. Sheffield, UK, 1993.

LEED93 Interactive Document Compiler Version 1.36, Sampler 1.0, December 1993, K. Tait, Computer Based Learning Unit, The University, Leeds, LS2 9JT, UK. E-mail: ken@uk.ac.leeds.cbl. Available free from HENSA. ftp to micros.hensa.ac.uk where the file is micros/ibmpc/win/i/i015/idc136.zip

LILL93 Lilley, C.C., F.Lin, W.T.Hewitt, T.L.J.Howard, "The design and development of distance learning materials for graphics and visualisation", Computers and Graphics, 18(3), 1994, pp.269-275.

MADD93a Maddock, S.C., "Bootstrapping computer graphics education using student power", Proc. 1st Eurographics Workshop on Graphics and Visualization Education, Eurographics Technical Report Series ISSN 1017-4656, eds. K. Brodlie and G. Owen (Barcelona, 4-5 September 1993), 5.1-5.4

MADD93b Maddock, S.C., R.H. Minors, J.J. McGregor, "Disce Doce: The didactic experience of developing a CAL tool", First United Kingdom Toolbook Users Conference, (Glasgow Caledonian University, 8-9 September 1993), pp 34-35.

MADD94 Maddock, S.C., "A Personal CAL Workbook", Computers and Graphics, 18(3), 1994, pp.299-304.

OWEN92 Owen, G.S., "Hypergraph - A Hypermedia System for Teaching Computer Graphics", Interactive Learning Through Visualisation, S. Cunningham & R.J. Hubbold, eds., Springer-Verlag, 1992.

ROWL89 Rowland, S., "Groups and Teams", British Journal of Inservice Education, 15(3), 1989, pp. 209-212.

SCHW92 Schweitzer, LtCol. D., "Designing Interactive Visualization Tools for the Graphics Classroom", 23rd SIGCSE Technical Symposium on Computer Science Education, Kansas City, MI, March, 1992 (reprinted in SIGCSE Bulletin, 24(1)).

WATT93 "3D Computer Graphics (second edition)", Addison-Wesley, 1993.

WATT92 "Advanced Animation and Rendering Techniques: theory and practice", Addison-Wesley, 1992.

Appendix A

2. Assignment: A Personal CAL Workbook

This project is worth 15%.
It is a group project. You will work in teams of four. I will choose the teams.

Handout date: week 1, term 2
Handover date: (Stage 1) am, Friday 25 February, week 7, term 2.
Handin date: (Stage 2) 5pm, Tuesday 15 March, week 10, term 2

2.1 The Project

The aim of this project is for the whole class to author a CAL system for the Computer Graphics half of C11. My notes will be the starting point. Word for Windows versions of them are available in r:\data\graphics\c11notes.
Each team will be assigned a different chapter of the CAL book. The different chapters should cover my notes as follows:
Chapter 1 covers 1. The Discrete Pixel
Chapter 2 covers 2. Geometry; 3. Coordinate Systems; 4. Windows and Viewports.
Chapter 3 covers 5. Interaction
Chapter 4 covers 6. Raster Techniques (don't include 6.7 image processing)
Chapter 5 covers 7. Parametrics
Chapter 6 covers 8. Modelling Natural Phenomena
Chapter 7 covers 9. Animation
Chapter 8 covers 10. Scientific Visualisation

The team working on Chapter 2 can invent their own title for the chapter. Others should use the title of the relevant chapter of my notes.
It is up to you how you structure the contents of the chapter - my notes are only a starting point. In writing your chapter, you should assume that the reader is familiar with the material in the previous chapters. Only the text of my notes is available in r:\data\graphics\c11notes. You will have to generate the diagrams yourself - use drawing packages or paint systems or scanning them in from papers and books.
As part of your Workbook (I am using the term Workbook to mean a Toolbook-based CAL system created using HyperBookShell - see 2.2, HyperBookShell) you should regularly assess the reader. This is to be done using multiple choice questions. It is possible to use these quite creatively. I will assess your ability to ask a reader pertinent questions at regular intervals in your chapter.
You should link in some programs from r:\data\graphics to illustrate relevant parts of your chapter. You must also *write* an interactive Modula-2 program that illustrates some part of your chapter. This may be based on a program from r:\data\graphics, but should improve on it in some way. I will assess this program for its illustrative abilities.
There will be at least two teams working on each chapter. At week 7, you will hand over a copy of your system to the team working on the same chapter as you. They will do likewise. As part of your deliverables in week 10 you will supply a review of the other team's system. You might compare it against your own system, assessing its strength's and weaknesses.
So, to summarise, there are three things you have to do:
1. Create a Workbook for the particular chapter you are assigned.
2. Write a Modula-2 program to illustrate some part of your chapter.
3. Write an evaluation of another team's system.

2.2 HyperBookShell

From the user guide (see 3. User guide for HyperBookShell): HyperBookShell is a simple, structured framework which allows a user to create and manage chapters, sections and pages, and to fill those pages with text and graphics - a sort of book processor. It also allows easy linking

to external programs and the formulation of simple multiple choice questions.
It is written in Asymetrix Toolbook's Openscript. You do not have to write Openscript. You just use HyperBookShell. It is a graphical interface, with icons for constructing and manipulating chapters, sections and pages. The front cover of this practical handout shows the interface to HyperBookShell.
Further details of how to use HyperBookShell are given in the user guide (see 3. User guide for HyperBookShell). For the purposes of your project you should be aware that you can do the following things in HyperBookShell:

- create and edit chapters, sections and pages;
- create pages of text and graphics
- create a page of multiple choice questions
- create a launch page - a page that allows you to link to an executable program
- text can be cut and pasted from Word for Windows to the pages of HyperBookShell
- pictures can be cut and pasted from the Windows clipboard to HyperBookShell

Obtaining HyperBookShell
Copy everything using the following:

```
XCOPY r:\data\graphics\hbs  n:\c11prac  /s
```

Important Note
HyperBookShell is big, approximately 600K big. As you add pictures it gets much bigger. Make careful use of pictures. If the size becomes unmanageable, then you can include all pictures as a separate printed booklet and refer to them from within your CAL Workbook.
When you have nominated your team leader, I will arrange that his or her disc quota will be increased by 4Mb. Thus, the team leader will be responsible for the project. It is *extremely important* that you all make use of p:\data\graphics\pkzip to *regularly* make compressed copies of the system on floppy discs.

2.3 Deliverables

- first introduction page - name and userid of team leader, names of other team members, module title (ie. C11 - HCI and Computer Graphics), project title, date. Also list the different files (programs and data) that are required for your system. The project should be in a subdirectory called **`C11PRAC`** in the DOS root directory of the team leader (1 page)
- a brief user guide for the project - overview explaining what the system covers along with details of how to get started (1 or 2 pages of text).
- a brief design document. Include a brief discussion of design approach(es) used, a block diagram of the contents of the Workbook and a discussion of how and why you decided on the chapter structure and contents. Include details of how you apportioned the work amongst team members (max 4 pages of text). Sections (but not individual pages) of your workbook can be labelled with the name of the person(s) who wrote them.
- evaluation of another team's system (max 4 pages of text).
- your system (!!). I will copy all the software from your directory into my directory the day after the handin date.
- self assessment document - this has 2 parts:

(i) team assessment sheet (1 page). Answer the following:
What is good about this work is...
What is weak about this is ...
What we would need to do to make it better is...
We took ... amount of time because...
It deserves a mark of... because...
This mark should be distributed amongst the team members in the following ratio...

(ii) individual self assessment sheets - each member of the team should describe their contribution to the project.

Text should be word-processed. Diagrams are extra to text. Hand-drawn diagrams are ok. In fact, use lots of diagrams. I like ***good*** diagrams. DO NOT hand in program listings.

2.4 Marking

The marking will cover three things:
1. Your Workbook
2. You Modula-2 demonstration program
3. Your evaluation of another team's system.

Marks will be deducted for bad documentation. All deliverables must be included. Do as many stages of the project as you can in the time that you allocate to the project.
I will mark the complete project as a team mark. Then, I will consider your indication of the split of marks between team members. I will cross reference this with my opinion of each team member's contribution and finally award a mark to each team member.

Endnote
If everything goes according to plan, I should be able to put all your chapters together into a complete system and give this to each student at the start of next year's module. You will have been actively involved in understanding the content and structure of C11, both for yourselves and for those on next year's module - Disce Doce!!