Martin Richards

How to write a dissertation in LaTeX

Computer Science Tripos – Part II

St John's College

January 9, 2017

Proforma

Name:	**Martin Richards**
College:	**St John's College**
Project Title:	**How to write a dissertation in LaTeX**
Examination:	**Computer Science Tripos – Part II, July 2001**
Word Count:	**1587**[1] **(well less than the 12000 limit)**
Project Originator:	Dr M. Richards
Supervisor:	Dr Markus Kuhn

Original Aims of the Project

To write a demonstration dissertation[2] using LaTeX to save student's time when writing their own dissertations. The dissertation should illustrate how to use the more common LaTeX constructs. It should include pictures and diagrams to show how these can be incorporated into the dissertation. It should contain the entire LaTeX source of the dissertation and the makefile. It should explain how to construct an MSDOS disk of the dissertation in Postscript format that can be used by the book shop for printing, and, finally, it should have the prescribed layout and format of a diploma dissertation.

Work Completed

All that has been completed appears in this dissertation.

Special Difficulties

Learning how to incorporate encapulated postscript into a LaTeX document on both Ubuntu Linux and OS X.

[1] This word count was computed by `detex diss.tex | tr -cd '0-9A-Za-z \n' | wc -w`

[2] A normal footnote without the complication of being in a table.

Declaration

I, [Name] of [College], being a candidate for Part II of the Computer Science Tripos [or the Diploma in Computer Science], hereby declare that this dissertation and the work described in it are my own work, unaided except as may be specified below, and that the dissertation does not contain material that has already been used to any substantial extent for a comparable purpose.

Signed [signature]

Date [date]

Contents

1 Introduction **9**
1.1 Overview of the files . . . 9
1.2 Building the document . . . 9
1.2.1 The makefile . . . 9
1.3 Counting words . . . 10

2 Preparation **11**

3 Implementation **13**
3.1 Verbatim text . . . 13
3.2 Tables . . . 14
3.3 Simple diagrams . . . 14
3.4 Adding more complicated graphics . . . 14

4 Evaluation **17**
4.1 Printing and binding . . . 17
4.2 Further information . . . 17

5 Conclusion **19**

Bibliography **19**

A Latex source **23**
A.1 diss.tex . . . 23
A.2 proposal.tex . . . 29

B Makefile **33**
B.1 makefile . . . 33
B.2 refs.bib . . . 34

C Project Proposal **35**

List of Figures

3.1 A picture composed of boxes and vectors. 14
3.2 A diagram composed of circles, lines and boxes. 15
3.3 Example figure using encapsulated postscript 15
3.4 Example figure where a picture can be pasted in 16
3.5 Example diagram drawn using `xfig` 16

C.1 The center of the perceived image is sharp, while the edges get progressively blurrier 36

Acknowledgements

This document owes much to an earlier version written by Simon Moore [2]. His help, encouragement and advice was greatly appreciated.

Chapter 1

Introduction

1.1 Overview of the files

This document consists of the following files:

- `makefile` — The makefile for the dissertation and Project Proposal
- `diss.tex` — The dissertation
- `proposal.tex` — The project proposal
- `figs` – A directory containing diagrams and pictures
- `refs.bib` — The bibliography database

1.2 Building the document

This document was produced using LaTeX 2$_\varepsilon$which is based upon LaTeX[1]. To build the document you first need to generate `diss.aux` which, amongst other things, contains the references used. This if done by executing the command:

```
pdflatex diss
```

Then the bibliography can be generated from `refs.bib` using:

```
bibtex diss
```

Finally, to ensure all the page numbering is correct run `pdflatex` on `diss.tex` until the `.aux` files do not change. This usually takes 2 more runs.

1.2.1 The makefile

To simplify the calls to `pdflatex` and `bibtex`, a makefile has been provided, see Appendix B.1. It provides the following facilities:

`make`
: Display help information.

`make proposal.pdf`
: Format the proposal document as a PDF.

`make view-proposal`
: Run `make proposal.pdf` and then display it with a Linux PDF viewer (preferably "okular", if that is not available fall back to "evince").

`make diss.pdf`
: Format the dissertation document as a PDF.

`make count`
: Display an estimate of the word count.

`make all`
: Construct `proposal.pdf` and `diss.pdf`.

`make pub`
: Make `diss.pdf` and place it in my `public_html` directory.

`make clean`
: Delete all intermediate files except the source files and the resulting PDFs. All these deleted files can be reconstructed by typing `make all`.

1.3 Counting words

An approximate word count of the body of the dissertation may be obtained using:

```
wc diss.tex
```

Alternatively, try something like:

```
detex diss.tex | tr -cd '0-9A-Z a-z\n' | wc -w
```

Chapter 2

Preparation

This chapter is empty!

Chapter 3

Implementation

3.1 Verbatim text

Verbatim text can be included using `\begin{verbatim}` and `\end{verbatim}`. I normally use a slightly smaller font and often squeeze the lines a little closer together, as in:

```
GET "libhdr"

GLOBAL { count:200; all  }

LET try(ld, row, rd) BE TEST row=all
                        THEN count := count + 1
                        ELSE { LET poss = all & ~(ld | row | rd)
                               UNTIL poss=0 DO
                               { LET p = poss & -poss
                                 poss := poss - p
                                 try(ld+p << 1, row+p, rd+p >> 1)
                               }
                             }
LET start() = VALOF
{ all := 1
  FOR i = 1 TO 12 DO
  { count := 0
    try(0, 0, 0)
    writef("Number of solutions to %i2-queens is %i5*n", i, count)
    all := 2*all + 1
  }
  RESULTIS 0
}
```

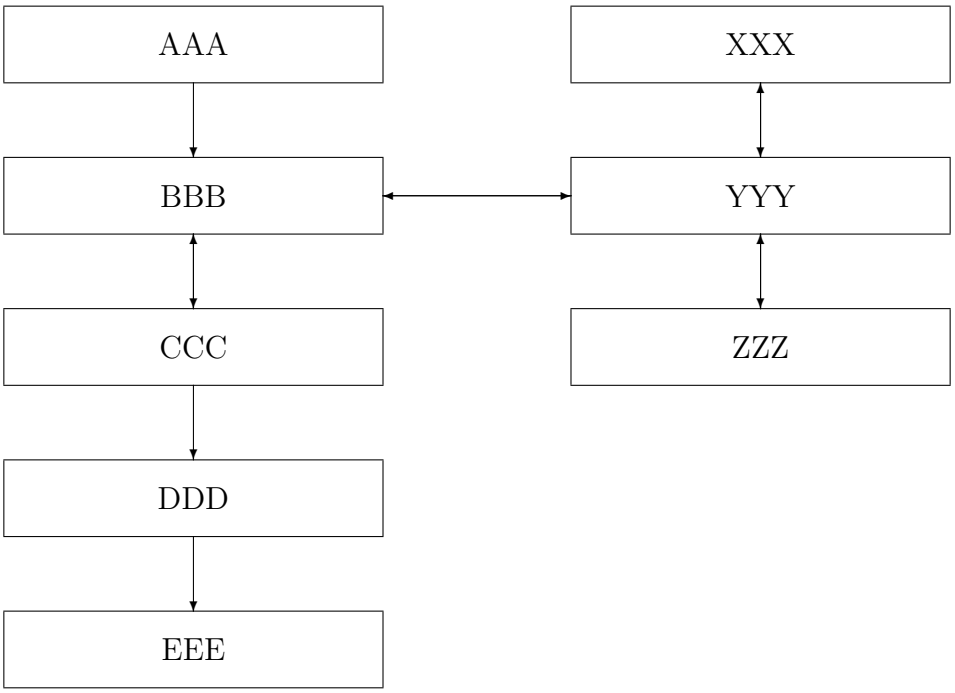

Figure 3.1: A picture composed of boxes and vectors.

3.2 Tables

Here is a simple example[1] of a table.

Left Justified	Centred	Right Justified
First	A	XXX
Second	AA	XX
Last	AAA	X

There is another example table in the proforma.

3.3 Simple diagrams

Simple diagrams can be written directly in LaTeX. For example, see figure 3.1 on page 14 and see figure 3.2 on page 15.

3.4 Adding more complicated graphics

The use of LaTeX format can be tedious and it is often better to use encapsulated postscript (EPS) or PDF to represent complicated graphics. Figure 3.3 and 3.5 on page 16 are

[1] A footnote

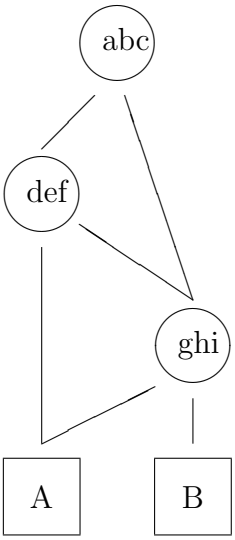

Figure 3.2: A diagram composed of circles, lines and boxes.

examples. The second figure was drawn using `xfig` and exported in `.eps` format. This is my recommended way of drawing all diagrams.

Figure 3.3: Example figure using encapsulated postscript

Figure 3.4: Example figure where a picture can be pasted in

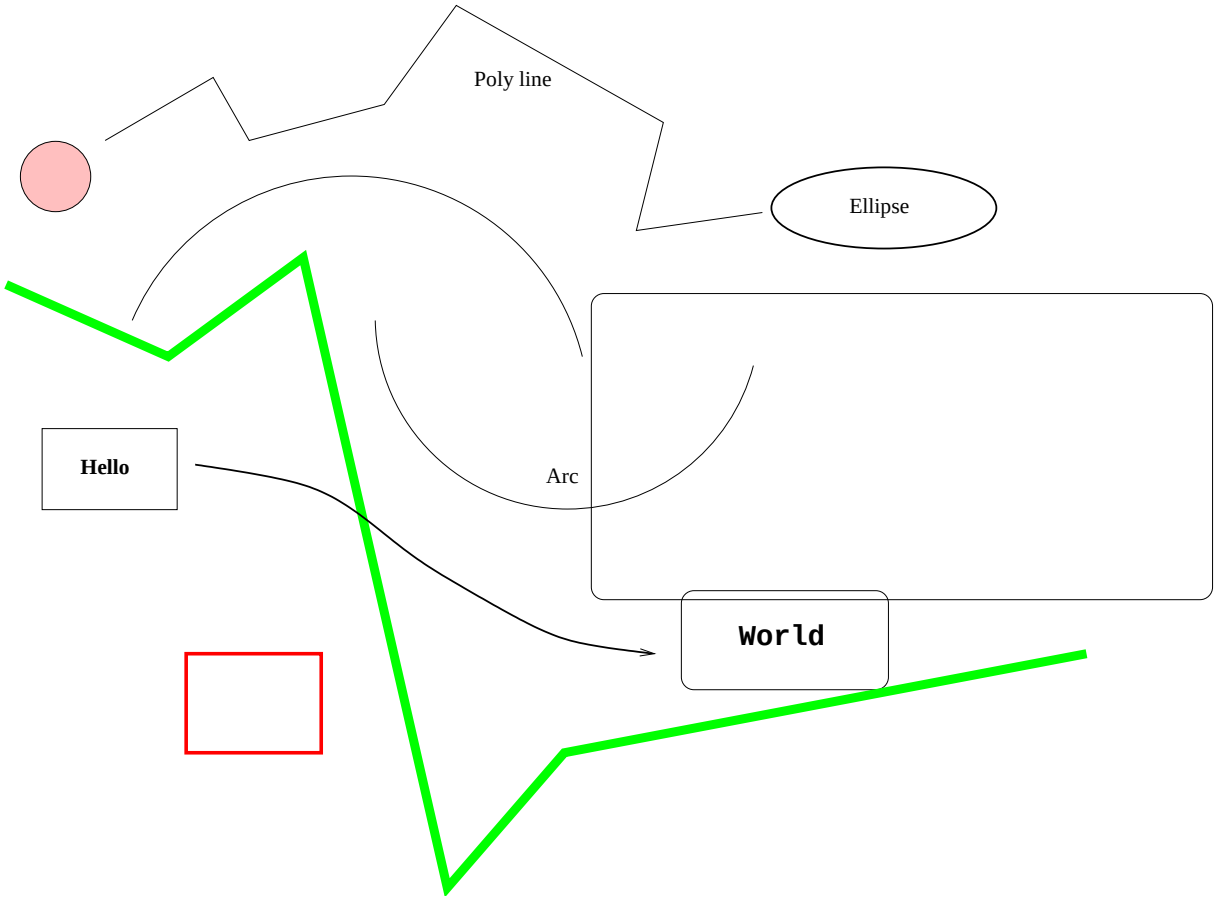

Figure 3.5: Example diagram drawn using `xfig`

Chapter 4

Evaluation

4.1 Printing and binding

Use a "duplex" laser printer that can print on both sides to print two copies of your dissertation. Then bind them, for example using the comb binder in the Computer Laboratory Library.

4.2 Further information

See the Unix Tools notes at

`http://www.cl.cam.ac.uk/teaching/current-1/UnixTools/materials.html`

Chapter 5

Conclusion

I hope that this rough guide to writing a dissertation is LaTeX has been helpful and saved you time.

Bibliography

[1] L. Lamport. *LaTeX — a document preparation system — user's guide and reference manual*. Addison-Wesley, 1986.

[2] S.W. Moore. How to prepare a dissertation in latex, 1995.

Appendix A

Latex source

A.1 diss.tex

```
% Template for a Computer Science Tripos Part II project dissertation
\documentclass[12pt,a4paper,twoside,openright]{report}
\usepackage[pdfborder={0 0 0}]{hyperref}    % turns references into hyperlinks
\usepackage[margin=25mm]{geometry}  % adjusts page layout
\usepackage{graphicx}  % allows inclusion of PDF, PNG and JPG images
\usepackage{verbatim}
\usepackage{docmute}   % only needed to allow inclusion of proposal.tex

\raggedbottom                           % try to avoid widows and orphans
\sloppy
\clubpenalty1000%
\widowpenalty1000%

\renewcommand{\baselinestretch}{1.1}    % adjust line spacing to make
                                        % more readable

\begin{document}

\bibliographystyle{plain}


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Title


\pagestyle{empty}

\rightline{\LARGE \textbf{Martin Richards}}

\vspace*{60mm}
\begin{center}
\Huge
\textbf{How to write a dissertation in \LaTeX} \\[5mm]
Computer Science Tripos -- Part II \\[5mm]
St John's College \\[5mm]
\today  % today's date
\end{center}

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Proforma, table of contents and list of figures

\pagestyle{plain}

\chapter*{Proforma}
```

```
{\large
\begin{tabular}{ll}
Name:               & \bf Martin Richards                       \\
College:            & \bf St John's College                     \\
Project Title:      & \bf How to write a dissertation in \LaTeX \\
Examination:        & \bf Computer Science Tripos -- Part II, July 2001  \\
Word Count:         & \bf 1587\footnotemark[1]
                      (well less than the 12000 limit)  \\
Project Originator: & Dr M.~Richards                    \\
Supervisor:         & Dr Markus Kuhn                    \\
\end{tabular}
}
\footnotetext[1]{This word count was computed
by \texttt{detex diss.tex | tr -cd '0-9A-Za-z $\tt\backslash$n' | wc -w}
}
\stepcounter{footnote}


\section*{Original Aims of the Project}

To write a demonstration dissertation\footnote{A normal footnote without the
complication of being in a table.} using \LaTeX\ to save
student's time when writing their own dissertations. The dissertation
should illustrate how to use the more common \LaTeX\ constructs. It
should include pictures and diagrams to show how these can be
incorporated into the dissertation.  It should contain the entire
\LaTeX\ source of the dissertation and the makefile.  It should
explain how to construct an MSDOS disk of the dissertation in
Postscript format that can be used by the book shop for printing, and,
finally, it should have the prescribed layout and format of a diploma
dissertation.


\section*{Work Completed}

All that has been completed appears in this dissertation.

\section*{Special Difficulties}

Learning how to incorporate encapulated postscript into a \LaTeX\
document on both Ubuntu Linux and OS X.

\newpage
\section*{Declaration}

I, [Name] of [College], being a candidate for Part II of the Computer
Science Tripos [or the Diploma in Computer Science], hereby declare
that this dissertation and the work described in it are my own work,
unaided except as may be specified below, and that the dissertation
does not contain material that has already been used to any substantial
extent for a comparable purpose.

\bigskip
\leftline{Signed [signature]}

\medskip
\leftline{Date [date]}

\tableofcontents

\listoffigures

\newpage
\section*{Acknowledgements}

This document owes much to an earlier version written by Simon Moore
\cite{Moore95}.  His help, encouragement and advice was greatly
```

```
appreciated.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% now for the chapters

\pagestyle{headings}

\chapter{Introduction}

\section{Overview of the files}

This document consists of the following files:

\begin{itemize}
\item \texttt{makefile} --- The makefile for the dissertation and
                        Project Proposal
\item \texttt{diss.tex} --- The dissertation
\item \texttt{proposal.tex}  --- The project proposal
\item \texttt{figs} -- A directory containing diagrams and pictures
\item \texttt{refs.bib} --- The bibliography database
\end{itemize}

\section{Building the document}

This document was produced using \LaTeXe which is based upon
\LaTeX\cite{Lamport86}.  To build the document you first need to
generate \texttt{diss.aux} which, amongst other things, contains the
references used.  This if done by executing the command:

\texttt{pdflatex diss}

\noindent
Then the bibliography can be generated from \texttt{refs.bib} using:

\texttt{bibtex diss}

\noindent
Finally, to ensure all the page numbering is correct run \texttt{pdflatex}
on \texttt{diss.tex} until the \texttt{.aux} files do not change.  This
usually takes 2 more runs.

\subsection{The makefile}

To simplify the calls to \texttt{pdflatex} and \texttt{bibtex},
a makefile has been provided, see Appendix~\ref{makefile}.
It provides the following facilities:

\begin{description}

\item\texttt{make} \\
 Display help information.

\item\texttt{make proposal.pdf} \\
 Format the proposal document as a PDF.

\item\texttt{make view-proposal} \\
 Run \texttt{make proposal.pdf} and then display it with a Linux PDF viewer
 (preferably ``okular'', if that is not available fall back to ``evince'').

\item\texttt{make diss.pdf} \\
 Format the dissertation document as a PDF.

\item\texttt{make count} \\
Display an estimate of the word count.

\item\texttt{make all} \\
Construct \texttt{proposal.pdf} and \texttt{diss.pdf}.
```

```
\item\texttt{make pub} \\ Make \texttt{diss.pdf}
and place it in my \texttt{public\_html} directory.

\item\texttt{make clean} \\ Delete all intermediate files except the
source files and the resulting PDFs. All these deleted files can
be reconstructed by typing \texttt{make all}.

\end{description}


\section{Counting words}

An approximate word count of the body of the dissertation may be
obtained using:

\texttt{wc diss.tex}

\noindent
Alternatively, try something like:

\verb/detex diss.tex | tr -cd '0-9A-Z a-z\n' | wc -w/


\chapter{Preparation}

This chapter is empty!


\chapter{Implementation}

\section{Verbatim text}

Verbatim text can be included using \verb|\begin{verbatim}| and
\verb|\end{verbatim}|. I normally use a slightly smaller font and
often squeeze the lines a little closer together, as in:

{\renewcommand{\baselinestretch}{0.8}\small
\begin{verbatim}
GET "libhdr"

GLOBAL { count:200; all  }

LET try(ld, row, rd) BE TEST row=all
                        THEN count := count + 1
                        ELSE { LET poss = all & ~(ld | row | rd)
                               UNTIL poss=0 DO
                               { LET p = poss & -poss
                                 poss := poss - p
                                 try(ld+p << 1, row+p, rd+p >> 1)
                               }
                             }
LET start() = VALOF
{ all := 1
  FOR i = 1 TO 12 DO
  { count := 0
    try(0, 0, 0)
    writef("Number of solutions to %i2-queens is %i5*n", i, count)
    all := 2*all + 1
  }
  RESULTIS 0
}
\end{verbatim}
}

\section{Tables}
```

```
\begin{samepage}
Here is a simple example\footnote{A footnote} of a table.

\begin{center}
\begin{tabular}{l|c|r}
Left      & Centred & Right \\
Justified &         & Justified \\[3mm]
%\hline\\%[-2mm]
First     & A       & XXX \\
Second    & AA      & XX  \\
Last      & AAA     & X   \\
\end{tabular}
\end{center}

\noindent
There is another example table in the proforma.
\end{samepage}

\section{Simple diagrams}

Simple diagrams can be written directly in \LaTeX.  For example, see
figure~\ref{latexpic1} on page~\pageref{latexpic1} and see
figure~\ref{latexpic2} on page~\pageref{latexpic2}.

\begin{figure}
\setlength{\unitlength}{1mm}
\begin{center}
\begin{picture}(125,100)
\put(0,80){\framebox(50,10){AAA}}
\put(0,60){\framebox(50,10){BBB}}
\put(0,40){\framebox(50,10){CCC}}
\put(0,20){\framebox(50,10){DDD}}
\put(0,00){\framebox(50,10){EEE}}

\put(75,80){\framebox(50,10){XXX}}
\put(75,60){\framebox(50,10){YYY}}
\put(75,40){\framebox(50,10){ZZZ}}

\put(25,80){\vector(0,-1){10}}
\put(25,60){\vector(0,-1){10}}
\put(25,50){\vector(0,1){10}}
\put(25,40){\vector(0,-1){10}}
\put(25,20){\vector(0,-1){10}}

\put(100,80){\vector(0,-1){10}}
\put(100,70){\vector(0,1){10}}
\put(100,60){\vector(0,-1){10}}
\put(100,50){\vector(0,1){10}}

\put(50,65){\vector(1,0){25}}
\put(75,65){\vector(-1,0){25}}
\end{picture}
\end{center}
\caption{A picture composed of boxes and vectors.}
\label{latexpic1}
\end{figure}

\begin{figure}
\setlength{\unitlength}{1mm}
\begin{center}

\begin{picture}(100,70)
\put(47,65){\circle{10}}
\put(45,64){abc}

\put(37,45){\circle{10}}
\put(37,51){\line(1,1){7}}
```

```
\put(35,44){def}

\put(57,25){\circle{10}}
\put(57,31){\line(-1,3){9}}
\put(57,31){\line(-3,2){15}}
\put(55,24){ghi}

\put(32,0){\framebox(10,10){A}}
\put(52,0){\framebox(10,10){B}}
\put(37,12){\line(0,1){26}}
\put(37,12){\line(2,1){15}}
\put(57,12){\line(0,2){6}}
\end{picture}

\end{center}
\caption{A diagram composed of circles, lines and boxes.}
\label{latexpic2}
\end{figure}



\section{Adding more complicated graphics}

The use of \LaTeX\ format can be tedious and it is often better to use
encapsulated postscript (EPS) or PDF to represent complicated graphics.
Figure~\ref{epsfig} and~\ref{xfig} on page \pageref{xfig} are
examples. The second figure was drawn using \texttt{xfig} and exported in
{\tt.eps} format. This is my recommended way of drawing all diagrams.


\begin{figure}[tbh]
\centerline{\includegraphics{figs/cuarms.pdf}}
\caption{Example figure using encapsulated postscript}
\label{epsfig}
\end{figure}

\begin{figure}[tbh]
\vspace{4in}
\caption{Example figure where a picture can be pasted in}
\label{pastedfig}
\end{figure}


\begin{figure}[tbh]
\centerline{\includegraphics{figs/diagram.pdf}}
\caption{Example diagram drawn using \texttt{xfig}}
\label{xfig}
\end{figure}


\chapter{Evaluation}

\section{Printing and binding}

Use a ``duplex'' laser printer that can print on both sides to print
two copies of your dissertation. Then bind them, for example using the
comb binder in the Computer Laboratory Library.

\section{Further information}

See the Unix Tools notes at

\url{http://www.cl.cam.ac.uk/teaching/current-1/UnixTools/materials.html}


\chapter{Conclusion}
```

```
I hope that this rough guide to writing a dissertation is \LaTeX\ has
been helpful and saved you time.


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% the bibliography
\addcontentsline{toc}{chapter}{Bibliography}
\bibliography{refs}

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% the appendices
\appendix

\chapter{Latex source}

\section{diss.tex}
{\scriptsize\verbatiminput{diss.tex}}

\section{proposal.tex}
{\scriptsize\verbatiminput{proposal.tex}}

\chapter{Makefile}

\section{makefile}\label{makefile}
{\scriptsize\verbatiminput{makefile.txt}}

\section{refs.bib}
{\scriptsize\verbatiminput{refs.bib}}


\chapter{Project Proposal}

\input{proposal}

\end{document}
```

A.2 proposal.tex

```
% Note: this file can be compiled on its own, but is also included by
% diss.tex (using the docmute.sty package to ignore the preamble)
\documentclass[12pt,a4paper,twoside]{article}
\usepackage[pdfborder={0 0 0}]{hyperref}
\usepackage[margin=25mm]{geometry}
\usepackage{graphicx}
\usepackage{parskip}
\begin{document}

\begin{center}
\Large
Computer Science Tripos -- Part II -- Project Proposal\\[4mm]
\LARGE
Smart Anti-Aliasing for Virtual Reality \\[4mm]

\large
G.~Ash, Fitzwilliam College

12 October 2016
\end{center}

\vspace{5mm}

\textbf{Project Supervisor:} Dr R.~Mantiuk

\textbf{Director of Studies:} Dr R.~Harle
```

```
\textbf{Project Overseers:} Prof R.~Anderson  \& Prof J.~Bacon

% Main document

\section*{Introduction}

A problem with modern Virtual Reality headsets is that they use low resolution displays to cover a huge Field of View.
Graphical artefacts, such as moir\'e patterns and pixellated edges (jaggies), are pronounced on these displays.
A good technique to ameliorate these artefacts if super-sampling, but super-sampling is often too expensive in VR
devices where low latency is a requirement - to avoid simulation sickness.

Unfortunately, modern consumer headsets suffer from astigmatism because of a single lens between the viewer and the display.

\begin{figure}[tbh]
\centerline{\includegraphics[width=0.3\linewidth]{figs/blur.png}}
\caption{The center of the perceived image is sharp, while the edges get progressively blurrier}
\label{blurfig}
\end{figure}

We could exploit astigmatism in single lens VR headsets by sampling more in the sharp center of the image,
and sampling less as we move towards the blurrier edges (Figure \ref{blurfig}). This project aims to extend an existing open

\section*{Starting point}

Novel anti-aliasing techniques, such as subpixel-reconstruction and temporal antialiasing, are still actively being develope

Free and extensible renderers exist for VR headsets, such as Unity. However I will be extending the existing open-source OSV
OSVR is compatible with all modern VR headsets, and supports all modern graphics APIs (OpenGL, Vulkan, Direct3D).

I have good experience in C/C++ through small projects and the IB C/C++ course. I'm familiar with the graphics pipeline and

\section*{Resources required}

For this project I shall be using my own quad-core machine with a VR capable GPU. I will also be using an Occulus Rift Devel
I will use an MCS machine as a failsafe incase my machine should break.

\section*{Work to be done}

The project breaks down into the following sub-projects:

\begin{enumerate}

\item \textbf{Setup} Fork the existing Open Source Virtual Reality-RenderManager repository. Create a daily cronjob to backu

\item \textbf {Core development} Develop/modify a simple super-sampling algorithm. Extend the algorithm to allow for areas o

\item \textbf {Demo creation} Create a couple of OpenGL demos that highlight both moire patterns and pixellated edges, for u

\item \textbf {Optimisation} Make use of a GPU profiler to determine any redundancy or inefficiency. Refactor the algorithm

\item \textbf {Evaluation} Make use of benchmarking/profiling software to evaluate the algorithm with regards to performance

\end{enumerate}

\section*{Success citeria}

The project will be a success if I manage to do the following:
\begin{enumerate}

\item Improve the performance of the open source renderer with anti-aliasing enabled

\item Provide an alternative anti-aliasing technique that suffers only negligible loss in image quality to the end user.

\item Evaluate both full screen antialiasing, my selective antialiasing approach, and no antialiasing from a user perspectiv

\end{enumerate}
```

```
\section*{Possible extensions}

If I achieve my main result early I shall try the following
alternative experiment or method of evaluation:

\begin{enumerate}

\item Research using a heuristic to determine salient objects or regions in the scene, and extend my algorithm to more close

\item Further reduce the requirement to super-sample by determining which objects/samples can be shared between each eye.

\end{enumerate}




\section*{Timetable}

Planned starting date is 16/10/2011.

\begin{enumerate}

\item \textbf{Michaelmas weeks 2--4} Start project Setup, Begin refreshing knowledge on OpenGL. Research the rendering pipel

\item \textbf{Michaelmas weeks 5--6} Complete project setup. Test writing custom code in the renderer, start implementation

\item \textbf{Michaelmas weeks 7--8} Continue development of algorithm. Start on demo creation

\item \textbf{Michaelmas vacation} Finish development of algorithm and demos.

\item \textbf{Lent weeks 0--2} Write progress report. Generate corpus of
  test examples. Begin optimisation.

\item \textbf{Lent weeks 3--5} Finish optimisation, begin evaluation of image quality on users.

\item \textbf{Lent weeks 6--8} Finish user studies, start performance analysis. Write up User studies in disseration.

\item \textbf{Easter vacation:} Begin on extensions, flesh out dissertation, complete evaluation.

\item \textbf{Easter term 0--2:}  Complete dissertation, proof read. Submit to DoS and supervisor for comments.

\item \textbf{Easter term 3:} Further proof reading/refactoring and submit dissertation.

\end{enumerate}

\section*{References}

\begin{enumerate}

\item Using Astigmatism in Wide Angle HMDs to Improve Rendering D. Pohl, T. Bolkart, S. Nickels, O. Grau, 2015.
\item Foveated 3d graphics. B. Guenter, M. Finch, S. Drucker, D. Tan, and J. Snyder. ACM SIGGRAPH Asia, 2012.
\item Oculus VR. Oculus Rift, 2014. http://www.oculus.com/

\end{enumerate}

\end{document}
```

Appendix B

Makefile

B.1 makefile

```
# This is the makefile for the Part II demonstration dissertation
#
# Note that continuation lines require '\' and
# that a TAB character preceeds any shell command line

.DELETE_ON_ERROR:

# Rules for building LaTeX documents (see Unix Tools course)
%.pdf %.aux %.idx: %.tex
        pdflatex -halt-on-error $<
        while grep 'Rerun to get ' $*.log ; do pdflatex $< ; done
%.ind: %.idx
        makeindex $*
%.bbl: %.aux
        bibtex $*
# Rules for exporting xfig diagrams into PDF or EPS
%.pdf: %.eps
        epstopdf --outfile=$@ $<
%.eps: %.fig
        fig2dev -L eps $< $@
%.pdftex %.pdftex_t: %.fig
        fig2dev -L pdftex_t -p $*.pdftex $< $*.pdftex_t
        fig2dev -L pdftex $< $*.pdftex

help:
        @echo
        @echo "USAGE:"
        @echo
        @echo "make                display help information"
        @echo "make proposal.pdf   format the proposal as PDF"
        @echo "make diss.pdf       format the dissertation as PDF"
        @echo "make all            make proposal.pdf and diss.pfd"
        @echo "make view-proposal  format and view the proposal"
        @echo "make view-diss      format and view the dissertation"
        @echo "make count          display an estimated word count"
        @echo "make pub            put demodiss.pdf onto your homepage"
        @echo "make clean          delete all intermediate files"
        @echo

view-%: %.pdf
        ( okular --unique $< || evince $< ) &

diss.pdf: figs/cuarms.pdf figs/diagram.pdf makefile.txt proposal.tex diss.bbl

makefile.txt: makefile
```

```
        expand makefile >makefile.txt

count:
        detex diss.tex | tr -cd '0-9A-Za-z \n' | wc -w

all: proposal.pdf diss.pdf

pub: diss.pdf
        rsync -t $+ $(HOME)/public_html/demodiss.pdf

clean:
        rm -f *.aux *.log *.err *.out
        rm -f *~ *.lof *.toc *.blg *.bbl
        rm -f makefile.txt

distclean: clean
        rm -f figs/*.pdf proposal.pdf diss.pdf
```

B.2 refs.bib

```
@BOOK{Lamport86,
TITLE = "{LaTeX} --- a document preparation system --- user's guide
and reference manual",
AUTHOR = "Lamport, L.",
PUBLISHER = "Addison-Wesley",
YEAR = "1986"}

@REPORT{Moore95,
TITLE = "How to prepare a dissertation in LaTeX",
AUTHOR = "Moore, S.W.",
YEAR = "1995"}
```

Appendix C

Project Proposal

Computer Science Tripos – Part II – Project Proposal

Smart Anti-Aliasing for Virtual Reality

G. Ash, Fitzwilliam College
12 October 2016

Project Supervisor: Dr R. Mantiuk
Director of Studies: Dr R. Harle
Project Overseers: Prof R. Anderson & Prof J. Bacon

Introduction

A problem with modern Virtual Reality headsets is that they use low resolution displays to cover a huge Field of View. Graphical artefacts, such as moiré patterns and pixellated edges (jaggies), are pronounced on these displays. A good technique to ameliorate these artefacts if super-sampling, but super-sampling is often too expensive in VR devices where low latency is a requirement - to avoid simulation sickness.

Unfortunately, modern consumer headsets suffer from astigmatism because of a single lens between the viewer and the display. We can't remove these distortions without (another) anastigmatic lens, however we do currently give equal preference to image quality across the whole of the display even when the edges of the image become distorted.

We could exploit astigmatism in single lens VR headsets by sampling more in the sharp center of the image, and sampling less as we move towards the blurrier edges (Figure C.1). This project aims to extend an existing open source system to include this optimisation, and to measure the impact on both the performance of the system, and on the image quality to the user.

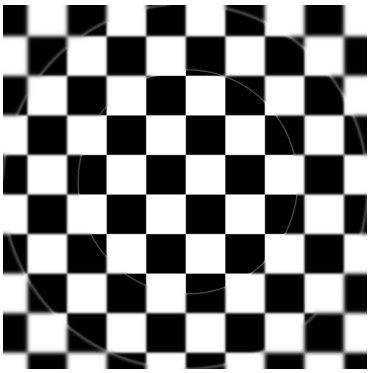

Figure C.1: The center of the perceived image is sharp, while the edges get progressively blurrier

Starting point

Novel anti-aliasing techniques, such as subpixel-reconstruction and temporal antialiasing, are still actively being developed. I intend to build on Intel's research[1] into a hybrid raytracing/rasterizing VR renderer by creating an entirely rasterized solution that retains image quality while allowing for GPU hardware acceleration.

Free and extensible renderers exist for VR headsets, such as Unity. However I will be extending the existing open-source OSVR-RenderManager, which will allow me to easily create separate reusable demos to show off certain graphical artefacts, and, if neccesary, to modify the entire rendering pipeline. OSVR is compatible with all modern VR headsets, and supports all modern graphics APIs (OpenGL, Vulkan, Direct3D).

I have good experience in C/C++ through small projects and the IB C/C++ course. I'm familiar with the graphics pipeline and have experience in WebGL, with some experience writing vertex and fragment shaders in OpenGL. I will need to refresh my knowledge of shader programming, and will need to take some time learning most of the OpenGL API.

Resources required

For this project I shall be using my own quad-core machine with a VR capable GPU. I will also be using an Occulus Rift Development Kit 2 [3](lent to me by the Hackers at Cambridge group) for testing and user studies. Source backups will be made both to a private Github repository and MCS daily. I will use an MCS machine as a failsafe incase my machine should break.

Work to be done

The project breaks down into the following sub-projects:

1. **Setup** Fork the existing Open Source Virtual Reality-RenderManager repository. Create a daily cronjob to backup this to github and MCS. Research my chosen renderer's pipeline.

2. **Core development** Develop/modify a simple super-sampling algorithm. Extend the algorithm to allow for areas of the screen to be ignored. Further extend to seamlessly composite draws that we sample differently.

3. **Demo creation** Create a couple of OpenGL demos that highlight both moire patterns and pixellated edges, for use in visual quality experiment.

4. **Optimisation** Make use of a GPU profiler to determine any redundancy or inefficiency. Refactor the algorithm to make it easily configurable.

5. **Evaluation** Make use of benchmarking/profiling software to evaluate the algorithm with regards to performance. Perform visual quality experiment to evaluate the impact of the algorithm on image quality to the user, recruit college members across fields to participate. Compare my approach against no anti-aliasing, and fullscreen anti-aliasing.

Success citeria

The project will be a success if I manage to do the following:

1. Improve the performance of the open source renderer with anti-aliasing enabled

2. Provide an alternative anti-aliasing technique that suffers only negligible loss in image quality to the end user.

3. Evaluate both full screen antialiasing, my selective antialiasing approach, and no antialiasing from a user perspective by constructing demos that show off artefacts masked by antialiasing.

Possible extensions

If I achieve my main result early I shall try the following alternative experiment or method of evaluation:

1. Research using a heuristic to determine salient objects or regions in the scene, and extend my algorithm to more closely resemble a foveated rendering[2] technique.

2. Further reduce the requirement to super-sample by determining which objects/samples can be shared between each eye.

Timetable

Planned starting date is 16/10/2011.

1. **Michaelmas weeks 2–4** Start project Setup, Begin refreshing knowledge on OpenGL. Research the rendering pipeline of my chosen renderer. Formulate an implementation strategy.

2. **Michaelmas weeks 5–6** Complete project setup. Test writing custom code in the renderer, start implementation of selective antialiasing algorithm
3. **Michaelmas weeks 7–8** Continue development of algorithm. Start on demo creation
4. **Michaelmas vacation** Finish development of algorithm and demos.
5. **Lent weeks 0–2** Write progress report. Generate corpus of test examples. Begin optimisation.
6. **Lent weeks 3–5** Finish optimisation, begin evaluation of image quality on users.
7. **Lent weeks 6–8** Finish user studies, start performance analysis. Write up User studies in disseration.
8. **Easter vacation:** Begin on extensions, flesh out dissertation, complete evaluation.
9. **Easter term 0–2:** Complete dissertation, proof read. Submit to DoS and supervisor for comments.
10. **Easter term 3:** Further proof reading/refactoring and submit dissertation.

References

1. Using Astigmatism in Wide Angle HMDs to Improve Rendering D. Pohl, T. Bolkart, S. Nickels, O. Grau, 2015.
2. Foveated 3d graphics. B. Guenter, M. Finch, S. Drucker, D. Tan, and J. Snyder. ACM SIGGRAPH Asia, 2012.
3. Oculus VR. Oculus Rift, 2014. http://www.oculus.com/