**A project report submitted in partial fulfilment for the degree of**

**BSc (Hons) Computer Games Development**

**School of Psychology and Computer Science**

**University of Central Lancashire**

Cross stitch pattern generator

Lili Veszeli

**12 April 2021**

# Abstract

Remove any text highlighted in yellow before submission.

You are aiming for no more than 50 pages of report content, this count starts at Chapter 1 and does not includes your references or appendices.

The abstract is the summary of the project report within one page (aim for about 500 words). Unnumbered chapter headings, as above, are entered using the ‘Heading (Unnumbered)’ style, which automatically starts a new page.

This template starts the page numbering at the foot of this page. That is, the first page does not have a number.

It is suggested that the abstract be structured as follows:

* Problem: What you tackled, and why this needed a solution
* Objectives: What you set out to achieve, and how this addressed the problem
* Methodology: How you went about solving the problem
* Achievements: What you managed to achieve, and how far it meets your objectives.

IMPORTANT – your abstract is a SUMMARY of the content of this report. It’s like a synopsis of a movie, with spoilers. Write it in the present tense. For example:

This project report discusses the process of implementing xxxxx. This report reviews, amongst other topics the current literature available on the use of xxxxx in xxxxx along with how the project has been planned, designed and developed.

Very formal

Describe the creation of the rceation of across stitch pattern. Quantization was used, wpf was used to show this pattern. Was successful, managed to achieve the goal.

# Attestation

I understand the nature of plagiarism, and I am aware of the University’s policy on this.

I certify that this document reports original work by me during my University project.

Signature: Lili Veszeli

Date: 01/04/2021

# Acknowledgements

Acknowledge anyone who has helped you in your work such as your supervisor, technical support staff, fellow students, or external organisations. Acknowledge the source of any work that is not your own.

Laurent Noel, Sam Connolly, John Hooson

# Table of Contents

The table of contents below is automatically generated from the paragraphs of style ‘Heading 1’ to ‘Heading 3’ and ‘Heading (Unnumbered)’. To update this after revisions, right-click in the table and choose ‘Update Field’ for the entire table. It is also possible to left-click the table and press F9 to update the entire table. Delete this paragraph and the one below before submission.

Sometimes Word does not automatically update references to other parts of the document, such as captions, Table of Contents and cross-references. To update everything, select the whole document by pressing Ctrl+A and then either press F9 or right-click anywhere and click on ‘Update Field’. Confirm all prompts by selecting ‘Update entire table’ and clicking OK. **Sometimes the font changes on update so check it is the correct font face before submitting your work**.

[Abstract i](#_Toc69154885)

[Attestation ii](#_Toc69154886)

[Acknowledgements iii](#_Toc69154887)

[Table of Contents iv](#_Toc69154888)

[List of Figures viii](#_Toc69154889)

[List of Tables ix](#_Toc69154890)

[List of Listings x](#_Toc69154891)

[1 Introduction 1](#_Toc69154892)

[2 Background and Related Work 2](#_Toc69154893)

[2.1 Introduction 2](#_Toc69154894)

[2.2 Cross Stitch History and Basics 2](#_Toc69154895)

[2.3 Image Processing 3](#_Toc69154896)

[2.3.1 Detail Preserving Resampling 4](#_Toc69154897)

[2.3.2 Posterization 5](#_Toc69154898)

[2.4 Application Platforms for Consideration 5](#_Toc69154899)

[2.4.1 .NET Framework 5](#_Toc69154900)

[2.4.2 Windows Forms 6](#_Toc69154901)

[2.4.3 WPF 6](#_Toc69154902)

[2.4.4 Comparison 7](#_Toc69154903)

[2.5 User Interface 7](#_Toc69154904)

[2.5.1 Colour 8](#_Toc69154905)

[2.6 Existing Cross Stitch Software 9](#_Toc69154906)

[2.7 Summary 10](#_Toc69154907)

[3 Project Planning 11](#_Toc69154908)

[3.1 Introduction 11](#_Toc69154909)

[3.2 Methodology 11](#_Toc69154910)

[3.3 Requirements 11](#_Toc69154911)

[3.4 Potential Solutions and Tools and Techniques 11](#_Toc69154912)

[3.5 Legal, Social, and Ethical Issues 11](#_Toc69154913)

[3.6 Summary 11](#_Toc69154914)

[4 Design 12](#_Toc69154915)

[4.1 Introduction 12](#_Toc69154916)

[4.2 System Design 12](#_Toc69154917)

[4.3 User Interface Design 12](#_Toc69154918)

[4.4 Summary 12](#_Toc69154919)

[5 Implementation 13](#_Toc69154920)

[5.1 Introduction 13](#_Toc69154921)

[5.2 Section 13](#_Toc69154922)

[5.2.1 Subsection One 13](#_Toc69154923)

[5.2.2 Subsection Two 14](#_Toc69154924)

[5.3 Section 14](#_Toc69154925)

[5.4 Summary 14](#_Toc69154926)

[6 Testing 15](#_Toc69154927)

[6.1 Introduction 15](#_Toc69154928)

[6.2 Functional Testing 15](#_Toc69154929)

[6.3 Non-Functional Testing 15](#_Toc69154930)

[6.4 User Testing 15](#_Toc69154931)

[6.5 Evaluation 15](#_Toc69154932)

[6.6 Summary 15](#_Toc69154933)

[7 Evaluation, Conclusions and Future Work 16](#_Toc69154934)

[7.1 Project Objectives 16](#_Toc69154935)

[7.2 Self-Evaluation 16](#_Toc69154936)

[7.3 Project Evaluation 16](#_Toc69154937)

[7.4 Applicability of Findings to the Commercial World 17](#_Toc69154938)

[7.5 Conclusions 17](#_Toc69154939)

[7.6 Future Work 17](#_Toc69154940)

[References 19](#_Toc69154941)

[Appendix 1 – Project Proposal 22](#_Toc69154942)

[Computing Degree Project Proposal 22](#_Toc69154943)

[Project Context 22](#_Toc69154944)

[Specific Objectives 22](#_Toc69154945)

[Potential Ethical or Legal Issues 22](#_Toc69154946)

[Resources 22](#_Toc69154947)

[Potential Commercial Considerations - Estimated costs and benefits 23](#_Toc69154948)

[Proposed Approach 23](#_Toc69154949)

[References 24](#_Toc69154950)

[Appendix 2 – Technical Plan 25](#_Toc69154951)

[Computing Project Technical Plan 25](#_Toc69154952)

[Title 25](#_Toc69154953)

[Summary 25](#_Toc69154954)

[Deliverables 25](#_Toc69154955)

[Constraints 25](#_Toc69154956)

[Key Problems 26](#_Toc69154957)

[System and Work Outline 26](#_Toc69154958)

[Project Activities 27](#_Toc69154974)

[Risk Analysis 28](#_Toc69154981)

[Potential Ethical or Legal Issues 29](#_Toc69154987)

[Commercial Analysis 29](#_Toc69154988)

[Employability Contribution 30](#_Toc69154994)

[References 30](#_Toc69154995)

[Appendix 3 – Title of Appendix 31](#_Toc69154996)

# List of Figures

Similarly, you can automatically generate a list of ‘Figures’. Right-click a figure (e.g. image or diagram) and add a caption labelled ‘Figure’ and ‘below selected item’. To update this after revisions, right-click in this table and choose Update Field (or use F9) and then choose to update the entire table. Delete this paragraph before submission.

[*Figure 1 - Example of a cross stitch on aida (RTO, n.d.).* 2](#_Toc69154997)

[*Figure 2 - Simple cross stitch pattern (Fitzgerald, 2017)* 3](#_Toc69154998)

[*Figure 3 – The way a pattern is converted to stitches (Biedl et al., 2005)* 3](#_Toc69154999)

[*Figure 4- Box filter (Parthipan, 2017).* 4](#_Toc69155000)

[*Figure 5 - Superpixels and final result (Gertsner et al., 2012).* 5](#_Toc69155001)

[*Figure 6 - Posterized image (Kwon & Chien, 2011).* 5](#_Toc69155002)

# List of Tables

Similarly, you can automatically generate a list of ‘Tables’. Select a table, right-click it and add a caption labelled ‘Table’ and ‘above selected item’. To update this after revisions, right-click in this table and choose Update Field (or use F9) and then choose to update the entire table. Delete this paragraph before submission.

[Table 1 - Test Results 15](#_Toc69155003)

# List of Listings

This list only applies to you if you use code snippets in your report. If you don’t have any listings, remove this whole section including the heading ‘List of Listings’.

You can automatically generate a list of ‘Listings’. After formatting your Code, move the cursor to the first line below your code block and click ‘References’ -> ‘Insert Caption’ in the ribbon menu. Select the label ‘Listing’ or add a new Label called ‘Listing’ if it does not yet exist. To update this after revisions, right-click the table and choose ‘Update Field’ (or use F9) and then choose to update the entire table. Delete this paragraph before submission.

[Listing 1 - [Program.cs] The main class of the program 13](#_Toc69155004)

# Introduction

For editorial consistency, it is important to use Word styles properly. Word 2003 onwards has so-called ‘quick styles’. If the styles referred to below are not visible on the Home ribbon in the Styles category, choose ‘Apply Styles’ from the down arrow at the bottom right of the Styles category. Styles can then be applied from the drop-down box. To make a style visible as a quick style, choose Apply Styles, then click Styles (the AA icon) or use ‘Alt + Ctrl + Shift + S’, then right-click on the style and then ‘Add to Style Gallery’.

Chapters are entered using the ‘Heading 1’ paragraph style. The Heading 1 style automatically moves to the start of a new page and supplies the next chapter number. Pressing enter on a ‘Heading 1’ heading automatically inserts a ‘Heading 2’ heading underneath.

There should not be any text between a parent heading and its first sub-heading. For example, when you want to write an introductory section for the following sections, give that introductory section an own ‘Introduction’ heading instead of writing it between the heading preceding these sections and the first sub-heading.

As an example: This comment text is incorrectly placed between a ‘Heading 1’ (‘Introduction’) and a ‘Heading 2’ (‘Background and Context’).

Most text uses the ‘Normal Project Body’ paragraph style with 10-point Calibri, 1.5-line spacing, single-sided pages.

In general, use the default spacing that headings and paragraphs give you. Avoid using new-lines or spaces to format text. If you need to use quotes, preferably use single curly quotes ‘…’. If you wish to emphasise something, use the ‘Emphasis’ style. In addition, also a ‘Strong’ style is preconfigured.

Remember to Save frequently while you are working! Check that AutoSaving is enabled under options -> save -> 'Save AutoRecover information every 5 minutes'.

# Background and Related Work

* 1. Introduction

The popularity of arts and crafts is increasing among all age groups. Textile based hobbies, like knitting and embroidery are the 2nd most popular craft in the UK (Mintel, 2020). This statistic suggests that there is a market for an arts and crafts program. The objective of the project is to create a software that converts a picture into a cross stitch pattern. The user will be able to edit the pattern and customize it using different features of the program. This chapter reviews relevant literature about the topics in the project. It explains the basics of cross stitch, to gain an understanding of the fundamentals, and existing software. Another theme is about different methods for image downsampling, which will help to create the pattern. Different application platforms are evaluated, and UI design practices are explored.

* 1. Cross Stitch History and Basics

For the purposes of this project a brief introduction to cross stitch is essential in order to understand how to implement the process digitally. Cross stitching is a type of embroidery stitch art (Setiabudi et al., 2017). A picture is created using colourful thread and little cross shaped stitches on fabric. Usually, two strands of cotton thread are used (Dyer, 1997). The stitching is done on fabric that has small holes: aida or evenweave (see Figure 1). The colour range of the threads can be vast, the most widely used is made by DMC. There are only a few types of stitches since the final result should look uniform. The simple cross is used most of the time, however, a half stitch, quarter stitch, three-quarter stitch and backstitching is also used occasionally (Atkinson & Roberts, 1999). Considering the different types of stitches used in this art form is important because they bear a close resemblance to the way that images are displayed on a computer screen because they both appear to be made of little squares (pixels).

A picture containing food, sushi

Description automatically generated

*Figure 1 - Example of a cross stitch on aida (RTO, n.d.).*

This art-form originated in Asia, and the oldest cross stitch dates back to 850 B.C. It became more popular in the Victorian era. Then, in 1980 cross stitch re-emerged again, and became how we know it today. It is one of the most popular type of needlework throughout the world (Leslie, 2007). It is a soothing and therapeutic hobby, and could help improve mental health (Hohmann, 2020).

A pattern is needed to make a counted cross stitch (see Figure 2). A pattern consists of a grid with colourful squares (Biedl et al., 2005). The squares signal the position and colour of the stitch, like in Figure 3 (Atkinson & Roberts, 1999). The size of a pattern is measured by the number of stitches across and down.

Diagram

Description automatically generated

*Figure 2 - Simple cross stitch pattern (Fitzgerald, 2017)*

*Diagram

Description automatically generated*

*Figure 3 – The way a pattern is converted to stitches (Biedl et al., 2005)*

* 1. Image Processing

Digital image processing means manipulation of an image with a computer. The cross stitch program will do exactly this: manipulating a photo into a pixelated version by compressing it. This section overviews some techniques and previous solutions to help with achieving the desired effect. Image compression is an application area of image processing. An image has to be resampled in order to upscale it, downscale it or rotate it. Downsampling is a widely used image operation. It is used to reduce the storage requirements (Youssef, 1999). However, it won’t be used in this project for that reason, but to get the pixelated look.

An image is stored in a bitmap. It is defined by the number of pixels and the information contained in them. Image scaling means to “re-sample a two-dimensional function on a new sampling grid” (Parthipan, 2017). Several algorithms can be used in image scaling. The simplest and fastest one is the nearest neighbour method, where the value of the new point becomes the value of the closest input point. Another method is linear interpolation, where the new point is interpolated between the two closest old points. Bilinear interpolation calculates the new point from the weighted average of the four closest input points. These might produce undesired effects like aliasing. A more sophisticated algorithm is bicubic interpolation. It samples the closest 4 by 4 pixels, so 16 pixels, and interpolates between them. This is the standard for most image editing software (Patil, 2018; Parker et al., 1983).

One of the simplest and most efficient downsampling algorithm is the box filter. It is a linear filter algorithm. This is considered to be used in the basic version of the program. The way it works is that each pixel in the target image is the average of the pixel values in a square from the source image. It is easier to understand with a diagram, see *Figure 4*.

A picture containing table

Description automatically generated

*Figure 4- Box filter (Parthipan, 2017).*

This is a simple way of converting a picture into bigger squares like on a pattern, but this algorithm can easily miss important details, and the resulting image/pattern might not be recognizable. Other research tries to solve this problem by using more complex algorithms.

* + 1. Detail Preserving Resampling

This algorithm by Gertsner et al. (2012) converts high resolution images into pixel art style ones with reduced colour palettes. It converts faces and other detailed images much more accurately than a regular downsampling algorithm. Pixel art is very close to a cross stitch pattern, it almost looks identical without the grid, so this article can provide great ideas to approach the problem.

Each pixel needs to be carefully placed, so that it accurately represents the original image. Areas such as the eyes and mouth are especially difficult to get right, however, it’s achievable by using “superpixels”, see *Figure 5*. The algorithm uses an iterative process. First, the superpixels are initialized in a grid, and then each original pixel is assigned to the closest superpixel. The palette is also set to an average colour. Then, the iteration starts, the superpixels are refined and associated with colours in the palette, then palette is also refined, and expanded (Gertsner et al., 2012). Unfortunately, it doesn’t allow users to pick the palette colours, so that has to be solved in another way in the program.

A picture containing text

Description automatically generated

*Figure 5 - Superpixels and final result (Gertsner et al., 2012).*

* + 1. Posterization

Posterization reduces the number of tones used in an image. There is no gradient between the areas and usually vivid colours are used (Afifi, 2018; Kwon & Chien, 2011), see *Figure 6*. This algorithm can be useful for the program since it also needs to reduce the number of colours from the original image.

First, the algorithm of Afifi (2018) removes small details from the image, then applies bilateral filtering. After the image is prepared for quantization (the process to reduce the number of colours in an image to a limited palette) (Orchard & Bouman, 1991), the pixels are sorted into ‘bright’, ‘grey’, and ‘dark’ categories using fuzzy logic.

A picture containing wall, person, clothing, suit

Description automatically generatedA picture containing text, suit

Description automatically generated

*Figure 6 - Posterized image (Kwon & Chien, 2011).*

* 1. Application Platforms for Consideration
     1. .NET Framework

A suitable application platform needs to be selected for the software. This desktop application will be Windows based, so the Microsoft .NET Framework will be used. Figure 7 illustrates the .NET stack. This technology supports running Windows and Web apps (Microsoft Docs., 2020). Both WPF and Windows Forms are part of the framework, however, they are best suited for different types of applications (Misra, 2016). Both of these platforms are evaluated and considered for the project below.

Graphical user interface, application

Description automatically generated

Figure - *.NET Framework (Soumyasch, 2007).*

* + 1. Windows Forms

Windows Forms was released by Microsoft in 2002 as part of the .NET framework. This greatly influenced how Windows applications are written, explains Griffiths and Adams (2003). Before Windows Forms developers could only use Win32 to make Windows applications, this is lower level and does not have a GUI (Graphical User Interface). By increasing the level of abstraction, a higher-level object-oriented API was created. This makes it much simpler to develop an application, allowing to concentrate on the task rather than the low-level details. In Visual Studio developers can make use of the Windows Form Designer, where they can drag and drop controls into the UI. WinForms is event driven meaning when the user interacts with the interface, for example, clicking a button, an event occurs. The application processes these events with the help of event handlers, which are programmed in C# or other high-level languages (Microsoft Docs., 2017).

* + 1. WPF

WPF (Windows Presentation Foundation) is a Graphical User Interface framework (Misra, 2016). It was released in 2006 with the new .NET 3.0 framework (Xu, 2010). The expectations for user interfaces were increasing, so new technology was needed, according to Nathan (2010). The user interface needed to be separated from the implementation, so that programmers and designers could work on the application without relying on one another. As an answer to this Microsoft released WPF. The user interface design is done in XAML (Extensible Application Markup Language), completely independent of the code. XAML is an XML based markup language designed for WPF (Microsoft Docs., 2016). This way the development is more efficient, and the cost is reduced. It is also easy to understand for designers. It is much more powerful than Windows Forms since it supports documents, multimedia, 2D and 3D graphics and animation (Xu, 2010). Before it would have required several different technologies to make an application containing all these. WPF is built on Direct3D, so it can provide high performance graphics.

* + 1. Comparison

The decision about which platform to use depends on the requirements of the application. WPF allows sophisticated UI design, while in WinForms the UI is not as customizable. UI is an important part of the program, so extensive customizability is preferred. WPF supports multimedia and graphics, while WinForms would need a third-party control to do the same. The cross stitch program mainly deals with images, so support for that is essential. WinForms uses less memory than WPF, however, memory is not a primary concern. WPF might have a steeper learning curve, and has less documentation, but it still proves to be the right choice for developing the application (Misra, 2016; Microsoft Docs., 2019).

* 1. User Interface

A user interface allows the user to interact with the program (Galitz, 2007). It is part of the field of Human Computer Interaction. This topic needs to be researched in order to be able to design a good UI for the program. A good UI means one that has a well-functioning input and output mechanism and lets the user accomplish tasks without putting any obstacles in their way (Gunderloy, 2006).

The most widely used type of UI is a Graphical User Interface (GUI). The first successful system using it was released by Apple in 1984. It includes lots of visible clues (buttons, windows, mouse) and real-world analogies (like dragging a file into a trash can meaning deleting it) (MacDonald, 2008).

The main aim for it is to be intuitive and logical. It is also important to be consistent within the application and with other popular software (like Word). The reason for this is that users already learnt the layout and menu organization of them, and it would slow them down by having to learn a completely new UI (Gunderloy, 2006). The UI of the cross stitch program will be similar to widely used drawing programs like Photoshop (see Figure 8) or Paint Tool SAI, since it is also a type of art program.

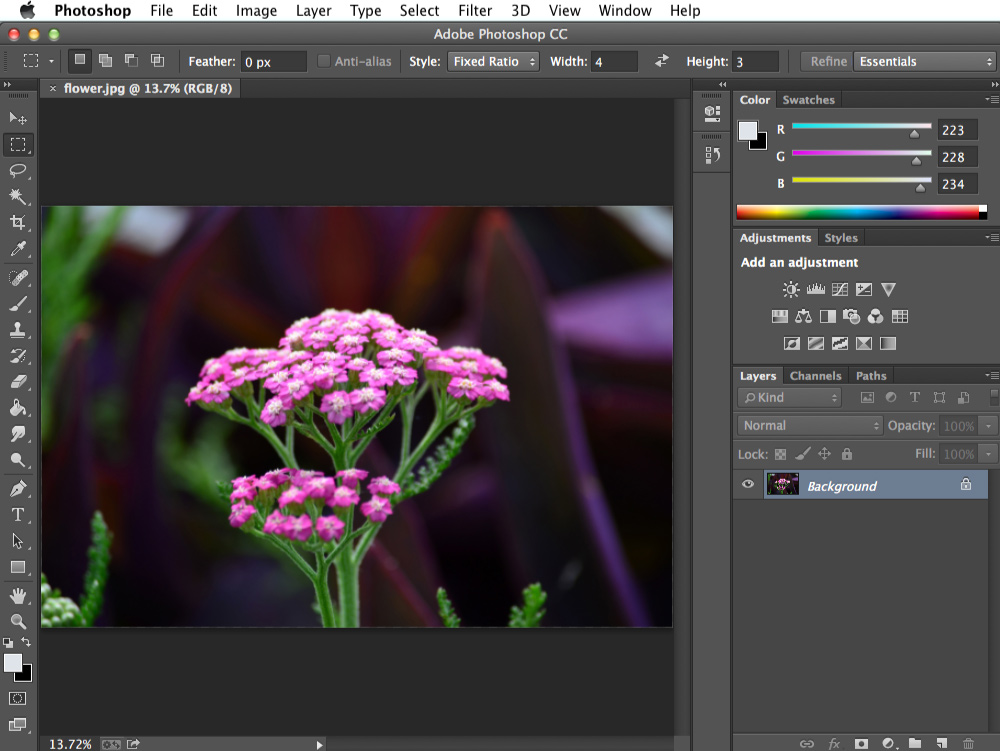


Figure - Photoshop User Interface (GCFGlobal, n.d.)

Another general principle for UI design is to be aesthetically pleasing. This is an important aspect of the project. It can be achieved by using colour and graphics effectively. It also helps to convey messages clearly.

Limiting the ability to make mistakes can greatly improve usability. The UI needs to be forgiving and always have an option to undo an action. Finally, a simple design helps to not confuse new users by not overwhelming them with an overcrowded interface. Limiting the number of things the user has to remember also helps to reduce memory work (Galitz, 2007).

Identifying the user base is also vital for the success of a program (MacDonald, 2008). More influential factors for the project are the users’ familiarity with the computer, frequency of use, gender, and age. The users of a cross stitch software will mostly be discretionary users, so it will only be used if the benefit of using it exceeds the effort learning it, according to Galitz (2007).

Failing to use the principles will make the UI inefficient and people will have difficulty doing their job. They might even stop using the program.

* + 1. Colour

Colour can be really powerful. It can convey emotions without saying anything (Barševska & Rakele, 2019). It is not a surprise that it has a great impact on the mood and behaviours of users. It provides structure and meaning to the screen. Research proved that colour in user interfaces improves performance and aids memory (Galitz, 2007). However, some combinations can strain the eye, so it is important to study colour theory to make the right decisions. There are different ways of selecting a good colour scheme. Based on colour theory, there are the following main combinations (see all of them in Figure 9):

* Monochromatic, which uses one colour with different hues.
* Complementary, which combines contrasting colours (red, green). It is difficult to harmoniously combine them in a UI.
* Split complementary, where a primary colour is used with two analogous colours to its complement.
* Triad combines three colours on the colour wheel with equal distance from each other.

Graphical user interface

Description automatically generated

Graphical user interface

Description automatically generated Graphical user interface

Description automatically generated

Graphical user interface, application, website

Description automatically generated

Figure - Colour combinations (Barševska & Rakele, 2019).

It is recommended to start designing the UI in black and white first. This helps focusing on the layout and simplifies the process. When the design is ready, then start adding colours as needed, but only with meaningful purposes. However, no more than five colours should be used. Avoiding pure black is encouraged since it looks unnatural. Keeping in mind the common meaning of colours will help the design. For example, red usually has meanings like stop and danger, so it shouldn’t be used for an OK button (Barševska & Rakele, 2019).

* 1. Existing Cross Stitch Software

This section provides an overview of existing cross stitch pattern making software. Knowing what is currently available will help inform this project. There are a number of cross stitch pattern making software on the market with varied quality and features. The most popular and advanced one is WinStitch/MacStitch (2019). This is a commercial software, and it is regularly updated. Another example is PC Stitch (2016), which was the preferred software until it stopped getting updates. Both can be quite expensive for a hobbyist, so a good alternative can be free, open-source software. These generally have less features and not as clean UI (user interface); however, they can still create a pattern. These include XStitch (Chestnut Pens, 2020) and CStitch (Klein, 2017).

All of the above software can convert a picture into a pattern. WinStitch has a wide range of features and a professional looking UI, so it is a good source of ideas for this project. Some features proving very useful are the ability to select the size of the pattern before generating it, as well as the number of colours used. These types of features are ones that will need to be considered in the design phase of this project. WinStitch uses real thread colours in the pattern, this makes it very practical for the user to buy supplies. See on Figure 10 that the thread colours are as close as possible to the original painting, and how detailed the pattern is. There are also a lot of options to edit the pattern, from changing thread colours to drawing on it. The biggest differences between the free and commercial applications are the number of features and how the user interface looks. WinStitch’s UI is very sleek and intuitive, while CStitch’s is quite antiquated and sometimes hard to use. One of the aims here is to design an attractive UI with all suitable features without having too many options that may overwhelm the user. The intention for this project is to be a good middle ground for users, available at a lower price while still maintaining quality.



Figure *- Painting and cross stitch. Pattern created in WinStitch (Batho, 2014).*

* 1. Summary

After reviewing and analysing past literature about relevant topics, I am more prepared to design and implement the application. A simple resampling algorithm can be used for the basic implementation and fast performance, however, a more advanced one is preferred to preserve detail. WPF enables developers to build a robust application with advanced graphics and highly customized UI. This is ideal based on the requirements of the project, therefore WPF will be used instead of Windows Forms. The UI needs to be simple, so that novice users can interact with it easily, and optimal colour palettes need to be considered for a harmonious look.

# Project Planning

## Introduction

Each of your chapters should have an introduction to tell your readers what they will find in the chapter. This should be written in the correct tense (i.e. this happens BEFORE you have embarked on any implementation). For example:

This chapter considers which methodology the project development will follow and what it is required to do. Furthermore, it presents the tools and techniques which will be used to implement the project alongside an explanation of legal and ethical issues that may occur.   
This chapter explores methodologies for project development and discusses possible tools and programming languages. Furthermore, the requirements are explained. And finally, it investigates legal and ethical issues in software development and considers possible problems in this project.

## Methodology

Spiral? Some agile things.

Write about waterfall and agile?

For a successful outcome it is important to decide the methodology used for the project management. There are a few things to take into consideration before deciding. The developer is only one person in this project so there is no team to be able to help; this means that some techniques from well knows methodologies can’t be used that were created for teams, like Scrum. There are limited resources and a limited time frame, so good time management is crucial. Requirements might change along the line, so the approach needs to be flexible enough to allow that.

Introducing agile and waterfall

The Agile method’s main characteristic is that it is more flexible than older methodologies like Waterfall (Tordrup Heeager, 2014). This is really useful in modern software development, since the features often change and evolve during the project. Based on this aspect it would be a good fit for this project. The high-level requirements have been decided, and the details are explored during development. It is likely that they could change in the future, which Agile can accommodate for. Another important aspect is effective time management, since time is one of the biggest constraints of the project. Timeboxing is an Agile technique that divides the development into iterations. Each iteration is one or two weeks, which fits in perfectly with the schedule of the supervisory meetings. This also allows a step-by-step approach to development, which makes it less overwhelming and divides up the workload to smaller parts. In Agile, testing is done after each iteration to make sure that the new features work together with the product (See Figure 11). This ensures that the software is functional by the time it is finished, unlike in the waterfall method where the testing is done at the end of the life cycle. Waterfall is usually used for bigger projects. Each phase is clearly defined, and the next phase can’t be started if the previous one is not completed. The features are also fixed (Balaji & Murugaiyan ,2012).

Diagram

Description automatically generated

Figure - Iterative Agile methodology (Hazevytch, 2020).

The Kanban method is also useful. A board can be created on Trello.com, which visualises tasks. With the help of this it’s easier to see which features need to be implemented and which were already done. It can also motivate the developer to see how much work has already been done. The tasks can be prioritised and put into different categories.

Communication with customer, like supervisor.

## Requirements

Requirements were gathered during the planning phase of the project. These requirements are based on existing cross stitch software, and on what I would want an application to have as a cross stitcher.

The requirements are prioritized using Moscow. This method sorts features into priority order in an easy to understand way (Waters, 2009). The priorities can change at the end of each timebox based on needs. The most important features are listed first.

### Must

* A Windows application that can create a pixelated downsampled image.
* Basic UI for the application.
* A grid over the pattern.
* The ability to save the pattern.

### Should

• The ability to convert an image into different size of patterns.

• A more advanced UI.

• Saving the pattern as a PDF.

• The ability to edit the pattern once it was generated.

• The ability to reduce the number of colours used in the pattern (quantization).

### Could

• The ability to use real thread colours in the pattern.

• The ability to pan and zoom in on the pattern while editing.

### Won’t

• Using different types of stitches, like back stitches and French knots.

• Representation of how the pattern will look once finished (preview).

• Blended thread colours, which gives more colours.

## Potential Solutions and Tools and Techniques

Desktop application, web application?

Windows, mac, linux?

There are several potential ways of creating a program like this, one of them is developing a desktop application. It would reach a greater user base if it could run on several operating systems, however because of time limitations it will be developed for only Windows since this is the most widely used operating system (Shanhong, 2021). Another option is creating a mobile application; however this wouldn’t be sufficient since enough screen space is needed to be able to edit the pattern.

A web application could be considered too. This way the user doesn’t have to download anything. The main obstacle with that is that it wouldn’t be possible to do in the time frame due to no experience in that field and the steep learning curve. Another downside is that an internet connection would be also required while using the program.

### Application platforms

Wpf, winforms, uwp, win32, winui 3 different ways of doing it, confusing in self reflection. Different pltaforms

There are various platforms to consider, each with their own pros and cons.

The Win32 API is the most performant of all of them. It has direct access to hardware. It uses C or C++, which is the language I have the most familiarity with. It is an unmanaged environment, unlike .NET. It doesn’t offer an editor so he UI has to be created in code using Windows functions. It is a great option for applications that need the highest efficiency and low-level control, but this project doesn’t require that. It also has a steeper learning curve than APIs with an inbuilt UI editor.

UWP (Universal Windows Platform) is the newest API from the ones considered. It works on all platforms as long as it uses windows 10. XAML is used for the UI and C# or C++ for the code behind. It is suitable for sophisticated UI and graphics-intensive scenarios. A con is that there aren’t as much documentation available as for WPF or Winforms, since it is relatively new. It would also be important for the application to work on older versions of Windows too, since a lot of the potential users might not have the newest operating system.

WPF and Winforms were discussed in detail in section 2.4. The advantage of both is that they have an inbuilt GUI designer, which speeds up the UI development process, and they support Windows 7 and later versions. WPF supports high quality 2D graphics, like images, geometry, and shapes. It also supports the use of pixel shaders (version 3.0), which allows for the implementation of effects. Takes advantage of the GPU and minimizes CPU usage. A useful feature of WPF is data binding; this enables a connection between a UI control and the data it displays. If the data changes, the UI reflects that automatically. Winforms doesn’t have these previous features, so WPF is selected as a platform for development.

C#, XAML C++ VB

### Programming languages

The programming language used in the project could be C#, C++ or Visual Basic.NET based on which application platform is chosen.

C++ is a general-purpose programming language designed by Bjarne Stroustrup (2007). He first developed “C with Classes” as an extension of C, and then using his experiences gained from that project created C++. It is a low-level, strictly typed language. It allows direct access to the memory, which makes it very efficient. It is best suited for performance critical programs. A downside could be that the memory has to be managed manually, unlike in the two other languages. I am most familiar with C++ so being able to use it in the project would be an advantage. WPF supports C++ in theory, however, the UI editor can’t be used. This would undermine the main point of using WPF. UWP and Win32 supports C++.

C# was developed by Microsoft as part of .NET in 2002. It is a strictly typed, object-oriented language, higher level than C++ (Microsoft Docs., 2021). It provides automatic memory management called garbage collection. This ensures there are no memory leaks, and programmer don’t have to deal with memory management (Bates, 2004). This also means that memory is harder to access. It can only be accessed inside blocks marked with the unsafe keyword. Memory cannot be explicitly freed, only garbage collection can do it. Once there are no references to an object it is collected. It is similar to C++ but there is still a slight difference, which adds some difficulty to the development. It is supported by WPF and UWP.

Visual Basic.NET is an object-oriented language. It is similar to C# in the sense that they are both higher level managed languages and are built on the .NET framework (Bell, 2002). It also uses automatic garbage collection. It is considered to be easier to learn than C-type languages because it is more similar to normal English. However, because of previous experience in C based languages it wouldn’t necessarily be easier for me, and making a change to VB would just slow down development considerably. WPF supports both C# and VB. Functionally equivalent with c# in wpf.

A good challenge to learn XAML and improve my C# skills.

Not case sensitive. Cannot use pointers.

### Other languages?

#### XAML

XAML is an XML based markup language developed by Microsoft in 2008. It is used in several .NET technologies, like WPF and Silverlight. It is also used in UWP. In all the mentioned technologies it is used for creating the user interface. Everything can be done in code behind that is done in XAML, however the advantages of using it is that it’s more concise and readable (Microsoft Docs., 2017). The separation of the UI and code behind encourages clean architecture. The UI can be modified without changing the application logic. It is easy to learn, even graphic designers can use it. A disadvantage of it is that it cannot contain logic, so all event handlers need to be declared in code. In WPF the XAML can be automatically generated based on the UI editor, and it can be written manually too (Microsoft Docs., 2020). Case sensitive.

HLSL maybe GLSL

#### HLSL

HLSL (High Level Shader Language) is a graphics programming language developed by Microsoft in 2003. It was introduced with DirectX 9 and Shader version 2.0. Before HLSL shaders were developed in a low-level language similar to assembly. Registers had to be managed by the programmer. HLSL was made to simplify and abstract shader development (St-Laurent, 2005). The syntax is similar to C languages but it has its own data types. The advantage of using shaders is that the calculations are done parallelly on the GPU, not using CPU, making the program much faster. Data is sent over from the CPU side and it’s rendered using a pipeline. The main parts of the graphics pipeline are vertex shaders, geometry shaders and pixel shaders (Li & Xu, 2009).

WPF can apply inbuilt effects to a control, like blur or drop shadow. Custom shader can also be written in HLSL. WPF only supports Shader model 3.0 pixel shaders. The final colour of a pixel is decided in the pixel shader.

### Other tools

#### IDE

Visual Studio

Visual Studio is an integrated development environment created by Microsoft. It is a code editor that supports programmers in development. It helps reducing mistakes and increases speed. Its features include IntelliSense code completion, code refactoring, integrated debugger, and syntax highlighting for easier coding (Muşlu et al, 2012). Both native and managed code can be written. It supports several languages including C++, C# and VB. The Community edition is free for everyone and the Enterprise versions is accessible for free for students. The .Net framework was built to work with visual studio, so it provides a comprehensive set of tools for development with .Net. The downside of it is that it takes up a lot of memory on the computer.

breakpoints to find problems faster.

Visual studio Code, need windows API

Lots of languages, not as many features as VS.

Things I didn’t use also

#### Version control

Github

GitHub is a cloud-based repository hosting service and code sharing platform (Loeliger & McCullough, 2012). It is essential to back up work while developing a project and GitHub helps exactly with that while offering a lot more features too. The code can be managed and tracked using Git based version control. Git was developed by Linus Torvalds. The way it’s used is that first a repository is created and then uploaded on the server. This can be accessed from any computer with the log in details. The repository can be public or private. All changes are recorded, and the older versions of the project can be accessed too. If a mistake was made the code can be rolled back to a previous state. Changes can be committed without internet connection. It can be also used for cooperation (Spinellis, 2012). Users can browse its largest collection of open source software (Borges et al, 2016). Feliciano et al (2016) found that using GitHub benefits students in several ways, for example it familiarizes us with a popular industry tool.

## Legal, Social, and Ethical Issues

### Ethical Issues

The role of ethics in software design has increased recently. The issues can be categorized into privacy, accuracy, property, and accessibility (Thomson & Scmoldt, 2001).

Improper access to private information causes privacy concerns. This application does not utilize any personal data from the user, so privacy issues do not arise.

The issue with accuracy comes up when the program does not work as it was expected to. In theory the developer is bound to fix all problems that the users encounter, especially if it is a commercial product, however sometimes fixing a bug introduces bigger problems. Most of the time significant bugs can be avoided with thorough testing.

Property issues include the knowledge possessed by individuals that was used in the development process. People contribute knowledge and skills to the project, which can lead to issues with intellectual property rights.

To have appropriate access to a program, a user must be able to provide the required input and comprehend the information output according to Thomson & Scmoldt (2001). The lack of the availability of a language in a program that the user understands causes an accessibility issue. The languages should be chosen based on the target audience. If concepts are used beyond the user’s understanding, then that can cause a problem.

### Accessibility

Accessibility for people with special needs is an important issue in today’s society (Kavcic, 2005). It is often taken for granted that a user can read information displayed on the screen, can hear sounds and can use the keyboard and mouse. However, not all of the potential users are capable of these. Supporting alternative input devices and allowing keyboard only access can help people with mobility issues, who can’t operate a mouse. Resizable text and good contrast between colours can support people with visibility problems. Flashing elements should be avoided. For hearing considerations, any audio has to be represented visually, like with subtitles. This project doesn’t have any audio components, so it’s no applicable.

#### Colour blindness

Some colours in the application and pattern might be hard to distinguish for colour blind people. 1 in 12 man has a colour-vision deficiency. Green and red are usually the most problematic, but other colours containing them can be affected too. A solution would be using a colour blind friendly colour palette (Rigden, 1999). However, these palettes can drastically decrease the number of colours available, especially if real thread colours are used. Another solution is using symbols to illustrate colours during editing, this way all colours can be used.

### Legal Issues

A legal issue could be libraries used that need license for distribution. All libraries used are open source (allowing the code to be freely used, modified, and shared), however they still have licenses. AForge has a GNU General Public License, version 3 (GPLv3). This is the most restrictive open source license: the source must be made public, and the software must be released under the same license. Csv helper uses Apache License 2.0; with this license the source code doesn’t have to be public, the software can use any license, and modifications need to be documented. Other libraries use the MIT license, which is the most permissive one: the source doesn’t have to be public, it can use any license, and changes don’t need to be documented. The license needs to be added as a text file to the root directory of the project (Raicea, 2017).

License text as appendix.

## Summary

In this section, Waterfall and Agile methodologies were considered, proving Agile techniques to be more appropriate. Further on, several APIs were compared resulting in WPF being the most suitable. Additional tools were explored with their advantages and disadvantages. And legal and ethical issues were discussed, highlighting important design considerations for inclusivity.

# Design

## Introduction

Each of your chapters should have an introduction to tell your readers what they will find in the chapter. Remember to change the sub chapter headings to some that are suitable to your project. Note that all heading are written in Title Case. Look this up if you are unsure what it means.

## Software design

Design patterns. Singleton, command.

Solid principle.

Design patterns are a solution to common programming problems (Johnson & Vlissides, 1995). It is like a template that can be applied to different scenarios. An architectural pattern is similar to a design pattern, but it applies to the overall system structure (Avgeriou & Zdun, 2005).

### MVVM Pattern

Model-View-ViewModel (WVVM) is an architectural pattern often used in WPF applications. The pattern was published by John Gossman, an architect of WPF, in 2005 (Sharma, 2019). It is similar to the widely used MVC (Model-View-Controller) pattern in the way that both separate the UI from the presentation logic (Microsoft Docs., 2012). One of the advantages of using this pattern is that it prevents tight coupling between the UI controls and the code behind. Tight coupling can cause difficulties when trying to update the user interface. The whole UI can be redesigned without any changes in the code if using MVVM. It is also easier to test the program using unit testing. Furthermore, it enables data binding, which is a powerful feature for displaying data in WPF. The code is also cleaner and easier to understand for future developers. A disadvantage of the pattern is that debugging is difficult when using data binding. Additionally, using the pattern for a simple UI can be excessive, and for a complex UI it could prove difficult to design.

There are three key components in the pattern: model, view, and viewmodel. Figure 12 shows the relationship between them.

Helps designers and programmers work together. Loose coupling.

It’s based on Martin Fowler’s Presentation Model pattern

Graphical user interface, application

Description automatically generated

Figure - Model-View-ViewModel components.

The view represents what can be seen on the screen. It is defined only with XAML.

The model is the data of the application.

<https://docs.microsoft.com/en-us/archive/msdn-magazine/2009/february/patterns-wpf-apps-with-the-model-view-viewmodel-design-pattern>

<https://www.tutorialspoint.com/mvvm/mvvm_advantages.htm>

observer with data binding, INotifyPropertyChanged interface and events.

Sharma, N. (2019). MVVM Design Pattern for building robust and scalable iOS apps. Retrieved April 20, 2021 from <https://medium.com/codewave/mvvm-design-pattern-c5d9f4a10758>.

Microsoft Docs. (2012). The MVVM Pattern | Microsoft Docs. Retrieved April 20, 2021 from https://docs.microsoft.com/en-us/previous-versions/msp-n-p/hh848246(v=pandp.10)?redirectedfrom=MSDN

## System Design

Load image, quantize image, save.

User stories.

## User Interface Design

Based on popular creative applications and existing cross stitch programs. I want to make it simple to use but also pleasing to the eye.

Storyboard – take picture.

Change based on user testing

## Summary

Write a short summary at the end of each chapter. Do not use the words ‘In summary’, we know what it is from the title.

# Implementation

## Introduction

Each of your chapters should have an introduction to tell your readers what they will find in the chapter. Depending on what type of project you are doing, you may name this chapter ‘Investigation’.

## Pattern creation

### Downsampling

Bicubic, using gdi library. Can input desired height of pattern.

Issue with black line, was fixed. Needed to wrap image.

### Colour quantization

Aforge , Wu quantizer , others, Implement myself, then exactly know how it does it

Chosen because it has a function that accepts the number of colours to what it reduces the palette.

It is also able to make an image use a certain palette.

Not the most precise. Sometimes returns less colours than the input value, especially if it is a bigger number. However, it is not always possible to get that exact number of colours, it depends on the image.

### DMC Thread colours

The final quantized colours of the image are converted into real life DMC thread colours. The colours are first converted to Lab colour space, since that is a better representation of how humans perceive colour, so better chance to get a good thread colour match. Blue value had to be tweaked using a heuristic.

Introduce Lab colours.

DMC colours stored in a csv file (RGB, name, number). Read in using csvHelper library, stored in List. DMC class created specifically for these colours. [code snippet]

Find another alternative for colormine, or implement for myself.

Colormine library searches for closest colours. Smallest delta E.

Colour palette created on the side.

## Pattern editing

Click on pixels, changes to current colour.

Colour picker (Dsafa). Choose rgb colour that is automatically converted to DMC. Or choose dmc colour. Can change a specific colour. Can delete a colour. Can erase stitches.

Palette updated.

## Pattern symbols

Small symbols added on top of stitches, to differentiate them easier. Each colour has different symbol. Drawing image, drawing group.

## Preview

Show how the finished cross stitch would look, using a picture of a cross stitch on aida canvas.

### Colour blending

Multiplicative blending. Getpixel, setpixel. Pointers. Slow.

### Shaders in WPF

Fast. Multiply original image with the passed colour.

Shader wrapper. In .fx file. Need to compile to make .ps file that is actually used. Need to set the effect to a control.

## User Interface

Palette, buttons (draw, colour picker, eraser, symbol, preview). Menu, undo redo, new, save, about.

Put in design.

### Undo/Redo functionality

Image stored in a list.

Explain how the pattern is created?

The libraries used?

* Aforge - LGPL v3(GNU Lesser General Public License) – used as DLL and not as static library.
* Colormine – open source. Why lab colours. Rgb hsl
* Dsafa – MIT license
* CsvHelper – open source (Microsoft Public License (MS-PL), Apache License, Version 2.0)
* GDI+ API

Class diagram? No

Code can be formatted using the ‘Code’ style. An example is shown below. It can be a little bit tricky to keep the formatting when pasting from an IDE but the following works for most IDEs: Copy the text from the IDE, paste it in Word, select the pasted code and change the style to ‘code’. It is worth noting that spell checking is deactivated for the ‘Code’ style.

Using System;

namespace HiWorld

{

    internal static class Program

    {

        private static void Main(string[] args)

        {

            Console.WriteLine(“Hello World!”);

        }

    }

}

Listing - [Program.cs] The main class of the program

Captions are entered through the ribbon menu under ‘References’ -> ‘Insert Caption’. Select ‘Listing’ (or add a new Label called ‘Listing’ if it does not already exist) and add the caption text in the white box, separated with a dash as the example above shows. Think about a naming convention for listings and stick to it throughout the report. For example, as seen above, ‘[ClassName or Filename] Description’.

In case you are mixing multiple programming languages: Consider stating the language name in the caption if it is not obvious from the file name or when there is no file name to refer to. For example, when you use XML and HTML, JavaScript and TypeScript or other languages with similar syntax. A suggestion might be to add the language in parenthesis at the end.

It is also possible to use the ‘code’ style “inline” to highlight commands in normal text by selecting the words to highlight and choosing the ‘code’ style. For example: This example demonstrates the ping 127.0.0.1 command.

## Summary

Write a short summary at the end of each chapter. Do not use the words ‘In summary’, we know what it is from the title.

# Testing

## Introduction

Each of your chapters should have an introduction to tell your readers what they will find in the chapter. In this chapter you should introduce your test strategy – how have you tested your artefact. You should also talk about user testing. How did you test with real people? How did you select them? What did you ask them to do? What ethical considerations did you adhere to? In this chapter you will also discuss how you have carried out an evaluation of your artefact. This is not the same thing as a total project evaluation.

This chapter introduces key testing strategies and describes how the application was tested.

About testing and how the app was tested.

Importance and aim of testing.

Talking about different types of testing: white box, black box, unit tests

Automation testing

Cat stripe testing

Singh, S. K., & Singh, A. (2012). Software testing. Vandana Publications.

Singh, Y. (2011). Software Testing. Cambridge: Cambridge University Press. doi:10.1017/CBO9781139196185

Mili, A., & Tchier, F. (2015). *Software testing : Concepts and operations*. ProQuest Ebook Central [https://ebookcentral.proquest.com](https://ebookcentral.proquest.com/)

## The Importance of Testing / General?

Nowadays many different fields depend on software. The expectations are much higher since the quality of the software can be critical in many real life scenarios and the financial stakes can be high if it fails (Mili & Tchier, 2015). Software testing can assure software quality. The aim of testing is to make sure that the program does what it is intended to do. Testing also increases user confidence, ensures customer satisfaction, and lowers maintenance cost. There are different strategies to testing, each one tests different aspects of the program. They can be divided into functional (unit, integration, acceptance), and non-functional (security, performance, usability) test. The way testing works generally is, the software is executed with desired inputs, and the outputs are compared with the expected ones. If the two don’t match, then there is a bug. A good test case is one that has a high probability of finding errors (Singh, 2012).

## Unit Testing

Unit testing tests the individual parts of the software, it is done at the lowest level. It is usually performed by the developers.

To add a caption to a table, either select the whole table (e.g. by clicking on the + symbol in the upper left corner of the table), right-click it and choose ‘Insert Caption’ or click in any table cell and select ‘References’ -> ‘Insert Caption’ from the ribbon menu. Choose ‘Table’ as label and ‘above the item’ as position. Add the caption text in the box, separated with a dash as the example below shows.

Table - Test Results

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## Non-Functional Testing

## User Testing

This testing method involves real users interacting with the application.

## Automation Testing

Diagram

Description automatically generated

Figure - Ideal test automation pyramid (Palamarchuk, 2015).

Palamarchuk, S. (2015). Best Testing Practices for Agile Teams: The Automation Pyramid. Retrieved April 22, 2021 from <https://abstracta.us/blog/test-automation/best-testing-practices-agile-teams-automation-pyramid/>.

## Evaluation

## Summary

Write a short summary at the end of each chapter. Do not use the words ‘In summary’, we know what it is from the title.

# Evaluation, Conclusions and Future Work

## Project Objectives

Summarise what you have achieved.

The objective of this project was to produce a program that can generate a cross-stitch pattern based on an image, which can be edited and saved. The aim of the program was to be easy to use for hobbyists and to be aesthetically pleasing.

The technical objectives were reached and even exceeded. The design objectives can be improved.

## Self-Evaluation

This section is about yourself. Be honest. Look at where you were situated at the beginning of the project and where you are now. What have you learnt on a personal level, what have you found out about yourself? Try to reflect upon individual goals, experiences, and incidents. No one is perfect, and it is very likely that you will recall both good and bad experiences.

The purpose of the evaluation process is to highlight strengths, correct performance weaknesses, and develop unused skills and abilities. To do this, you must be willing to recognise areas that need improvement or development.

The experience of delivering a longer project with time-constraints was very valuable both for developing my technical skills and improving as person. Since I haven’t done anything like this before it pushed me in a lot of ways. I think one of the main take away from this experience is that I need more confidence in myself. It was really difficult to start to do things alone without knowing if I am doing them right, but when I managed to start the application, I realized it is not as hard as I expected. I felt relieved and was able to achieve and even exceed my goals. I couldn’t have done this without the help of my project supervisor.

I became much better at individual work and research. I learnt to understand code that was not written by my lecturers and modified it to be able to be used in my application. My debugging skills and my understanding of shaders are way better, which proved useful in other modules too. I was wary of C# before, but now I am more confident in using it, even wish some features were available in C++.

I achieved much more than I expected. I learnt WPF and XAML alone. And was able to do things I would have never imagined without help.

And need to put less pressure on myself. My perfectionistic standards need to be lowered – already have been working on this. Need to believe that I can do something and not give up if I can’t do it or can’t understand it immediately. . Because of my perfectionism I sometimes kept procrastinating doing the project, because if I don’t do it then I can’t fail. I became so scared of it and was paralyzed by this fear, which spiralled into thinking I will never achieve anything in life and will never get a job. Things seemed dark

## Project Evaluation

Stand back and evaluate what you have achieved and how well you have met the objectives. Evaluate your achievements against your objectives in section 3.2. Demonstrate that you have tackled the project in a professional manner.

(The previous paragraph demonstrates the use of automatic cross-references: The ‘3.2’ is a Cross-reference to the text in a numbered item of the document, it is not literal text but a field. The number that appears here will change automatically if the number on the referred-to section is altered, for example if a chapter or section is added or deleted before it. Cross-references are entered using Word's Insert or References menu. Cross-references are set to update automatically when printed but may not do so on-screen beforehand; you can update a field manually on-screen by right-clicking on it and selecting Update field from the pop-up menu or by selecting the whole document and pressing F9.)

I exceeded the objectives of the project based on the Moscow requirements in my technical plan. I implemented all must have features except the grid, which was deemed unnecessary because the stitches were easy to see without it. All the should haves have been implemented, except saving it as a PDF, which was proven to be low priority and maybe unnecessary, since it can be saved as a picture. All the could haves have been implemented. Even the preview feature from the won’t have list was implemented which was a surprise for me, since I didn’t expect to get that far. Especially because the shader was quite involved.

In terms of the UI design I feel like I couldn’t quite achieve what I expected to. WPF UI design turned out to be more difficult than expected. Making a basic UI is easy enough, but making it look beautiful is very very hard. I think I managed to make the UI simple and easy to use. The colours are also aesthetic. I decided on a simple pastel design, since most cross-stitch patterns have a similar mood so it would suit them well.

## Applicability of Findings to the Commercial World

Summarise what you have achieved and how it can apply to the commercial world.

The finished application can be used by cross stitchers. It is a great alternative to expensive programs and provides a good range of features. An everyday user can easily make a usable and nice pattern.

It could be sold commercially, maybe with more added features. Alternatively, if it is released as a free open-source program, then parts of it could be reused and could help others who are trying to make similar applications.

the libraries used allow that. (Most of the libraries are open source and Aforge - LGPL v3(GNU Lesser General Public License) – used as DLL and not as static library).

## Conclusions

Summarise what you have achieved. Do not use the words ‘In conclusion’ or ‘to conclude’ or any derivative of those. We know this is the conclusions from the title.

Satisfied with the project. All features were added that were expected to be added in this timeframe. However, there is always room for improvement. Acquired a good base understanding of WPF, which adds to my skillset. Experienced working on a longer project, which helps with expectations about future development projects.

## Future Work

Explain any limitations in your results and how things might be improved. Discuss how your work might be developed further. Reflect on your results in isolation and in relation to what others have achieved in the same field. This self-analysis is particularly important. You should give a critical evaluation of what went well, and what might be improved.

The application is a functional pattern generator, however compared to the leading programs there is possibility for a lot of improvement.

Some of the already implemented features could be more polished. For example, when the pattern is first generated, the exact number of colours might differ from the input value. This is due to the colour quantization library used. However, colours can be added or removed afterwards.

The colours also sometimes appear to be duller than expected. This could be because of the conversion of RGB colours to the DMC thread colours. Now the RGB colours are converted to Lab space and then matched to the closest DMC colour. This is the best way for colour matching; however, another algorithm could make the process more sophisticated and choose colours closer to what a person would match them to.

The preview takes longer to load for bigger patterns. For example, a pattern with > 400 height would take a few minutes to generate. This would be dramatically reduced by using shaders for big patterns too (which is already used for smaller ones), however it takes up too much memory and the system runs out of GDI objects.

The downsampling of the image could be improved. Based on the paper written by Gertsner et al. (2012), a smart detail preserving algorithm can be used to downsample images. This could be especially useful for cross stitch patterns, since the details are the most important parts that make the image recognizable, and the backgrounds are usually plain and unimportant.

As for additional features, based on existing top applications (WinSticth), colour blending could be added. This generates more colours by blending existing thread colours together, resulting in a wider variety and closer colour matches to the original picture. At the moment only the simple cross stitch type is supported. This is the case for most free/smaller programs, however more stitch types could be added in the future, like French knots and backstitches.

An additional useful feature could be printing the DMC colour list together with the pattern. The list containing the colours would have to be exported together with the pattern when saving it. An option to save work in progress could be added. This way the user can continue to work on a pattern later even after closing the application. Similarly to other applications, more brands of threads could be added, like Anchor, and the user could choose which type they want to use.

Can’t use transparent images – transparent part will be black. I guess it can be edited afterwards.

# References

Cuevas, R., Cuevas, Á., Cabellos-Aparicio, A., Jakab, L., & Guerrero, C. (2010a). A collaborative P2P scheme for NAT Traversal server discovery based on topological information. *Computer Networks, 54*(12), 120-122.

Waters, K. (2009). Prioritization using moscow. Agile Planning, 12, 31.

Shanhong, L. (2021). Market share held by the leading computer (desktop/tablet/console) operating systems worldwide from January 2012 to February 2021. Retrieved April 15 2021 from: https://www.statista.com/statistics/268237/global-market-share-held-by-operating-systems-since-2009/.

Tordrup Heeager, L. (2014). How Can Agile and Documentation-Driven Methods be Meshed in Practice? Retrieved April 15 2021 from: https://link.springer.com/content/pdf/10.1007%2F978-3-319-06862-6.pdf.

Balaji, S., & Murugaiyan, M. S. (2012). Waterfall vs. V-Model vs. Agile: A comparative study on SDLC. International Journal of Information Technology and Business Management, 2(1), 26-30.

Hazevytch, L. (2020) Agile Methodology. [online image]. Devcom. https://devcom.com/tech-blog/agile-advantages-for-business/.

*A Tour of C# - C# Guide | Microsoft Docs*. (2021). Retrieved April 16, 2021, from https://docs.microsoft.com/en-us/dotnet/csharp/tour-of-csharp/

Stroustrup, B. (2007). Evolving a language in and for the real world: C++ 1991-2006. In Proceedings of the third ACM SIGPLAN conference on History of programming languages (pp. 4-1).

Bates, B. (2004). C# as a first language: a comparison with C++. Journal of Computing Sciences in Colleges, 19(3), 89-95.

Bell, D. (2002). Visual basic. Net as a first language: an evaluation. ACM Sigcse Bulletin, 34(4), 107-108.

XAML overview - WPF .NET | Microsoft Docs. (2020). Retrieved April 18, 2021, from https://webcache.googleusercontent.com/search?q=cache:h0oqkNyXIe4J:https://docs.microsoft.com/en-us/dotnet/desktop/wpf/fundamentals/xaml+&cd=1&hl=en&ct=clnk&gl=uk

Xamarin.Forms XAML Basics - Xamarin | Microsoft Docs. (2017). Retrieved April 18, 2021, from https://docs.microsoft.com/en-us/xamarin/xamarin-forms/xaml/xaml-basics/

Li, X., & Xu, M. (2009). Water simulation based on HLSL. In 2009 IEEE International Conference on Network Infrastructure and Digital Content (pp. 1066-1069). IEEE.

St-Laurent, S. (2005). The complete effect and HLSL guide. Paradoxal press.

Muşlu, K., Brun, Y., Holmes, R., Ernst, M. D., & Notkin, D. (2012). Speculative analysis of integrated development environment recommendations. ACM SIGPLAN Notices, 47(10), 669-682.

Loeliger, J., & McCullough, M. (2012). Version Control with Git: Powerful tools and techniques for collaborative software development. " O'Reilly Media, Inc.".

Spinellis, D. (2012). Git. IEEE software, 29(3), 100-101.

Borges, H., Hora, A., & Valente, M. T. (2016). Understanding the factors that impact the popularity of GitHub repositories. In 2016 IEEE International Conference on Software Maintenance and Evolution (ICSME) (pp. 334-344). IEEE.

Feliciano, J., Storey, M. A., & Zagalsky, A. (2016, May). Student experiences using GitHub in software engineering courses: a case study. In 2016 IEEE/ACM 38th International Conference on Software Engineering Companion (ICSE-C) (pp. 422-431). IEEE.

Thomson, A. J., & Schmoldt, D. L. (2001). Ethics in computer software design and development. Computers and Electronics in Agriculture, 30(1-3), 85-102.

Kavcic, A. (2005). Software accessibility: Recommendations and guidelines. In EUROCON 2005-The International Conference on" Computer as a Tool" (Vol. 2, pp. 1024-1027). IEEE.

Rigden, C. (1999). 'The Eye of the Beholder'-Designing for Colour-Blind Users. British Telecommunications Engineering, 17, 291-295.

Raicea, R. (2017). How open source licenses work and how to add them to your projects. Retrieved April 20, 2021, from: apps.quanticfoundry.com/surveys/answer/gamerprofile/.

Johnson, R., & Vlissides, J. (1995). Design patterns. Elements of Reusable Object-Oriented Software Addison-Wesley, Reading.

Avgeriou, P., & Zdun, U. (2005). Architectural patterns revisited-a pattern language.

Afifi, M. (2018). Image Posterization Using Fuzzy Logic and Bilateral Filter. *arXiv preprint arXiv:1802.01009*.

Atkinson, K. R., & Roberts, J. C. (1999). Graphics and Visualization within Cross-Stitch. In Eurographics UK, Conference Proceedings, 1999 (pp. 129-141). Eurographics.

Barševska, Z., & Rakele, O. (2019). COLOR IN UI DESIGN. *PROCEEDINGS OF THE 61st INTERNATIONAL SCIENTIFIC CONFERENCE OF DAUGAVPILS UNIVERSITY*, 79.

Batho, D. (2014). Pieta [painting and cross stitch]. <https://peacockandfig.com/2015/02/from-a-painting-to-a-cross-stitch/>.

Biedl, T. C., Horton, J. D., & López-Ortiz, A. (2005). Cross-Stitching Using Little Thread. In *CCCG* (pp. 199-202).

Chestnut Pens (2020) Cross Stitch Chart Creation Program. Retrieved from: http://www.chestnutpens.co.uk/misc/index.html.

*Choose your Windows app platform - Windows applications | Microsoft Docs*. (2019). Retrieved November 24, 2020, from <https://docs.microsoft.com/en-us/windows/apps/desktop/choose-your-platform>.

Dyer, K. (1997). Counted Cross Stitch Tutorial. Retrieved November 18, 2020, from: [https://www.crosstitch.com/nfaq/nfaq1.html#](https://www.crosstitch.com/nfaq/nfaq1.html).

Fitzgerald, S. (2017). A beginner’s guide to cross stitch – Red Gate Stitchery. Retrieved November 18, 2020, from https://redgatestitchery.com/blogs/red-gate-stitchery/a-step-by-step-guide-to-cross-stitch.

Galitz, W. O. (2007). *The essential guide to user interface design: an introduction to GUI design principles and techniques*. John Wiley & Sons.

Gerstner, T., DeCarlo, D., Alexa, M., Finkelstein, A., Gingold, Y. I., & Nealen, A. (2012). Pixelated image abstraction. In *Expressive* (pp. 29-36).

GFCGlobal (n.d.) Photoshop User Interface [Online image]. GCFGlobal. https://edu.gcfglobal.org/en/photoshopbasics/getting-to-know-the-photoshop-interface/1/.

Griffiths, I., & Adams, M. (2003). *NET Windows forms in a nutshell*. " O'Reilly Media, Inc.".

Gunderloy, M. (2006). *Developer to designer: GUI design for the busy developer*. John Wiley & Sons.

Hohmann, J. (2020) “We are all mad here”: A Content Analysis of Facebook Posts and Comments about the Impact of the Fibre Arts on Mental Health and Wellness.

*Introduction to WPF | Microsoft Docs*. (2016). Retrieved November 26, 2020, from https://docs.microsoft.com/en-gb/dotnet/desktop/wpf/introduction-to-wpf?view=netframeworkdesktop-4.8.

Klein, T. (2017) Cstitch. Retrieved from: http://cstitch.sourceforge.net/index.shtml.

Kwon, O. Y., & Chien, S. I. (2011). Improved posterized color images based on color quantization and contrast enhancement. In *Proc. Int'l Conf. Machine Vision, Image Processing, and Pattern Analysis* (Vol. 5, pp. 1203-1206).

Leslie, C. A. (2007). Needlework through history: An encyclopedia. Greenwood Publishing Group.

MacDonald, M. (2008). *User Interfaces in C#: Windows Forms and Custom Controls*. Apress.

MacStitch and WinStitch. (2019). Retrieved November 18, 2020, from <https://www.ursasoftware.com/macstitch/index.html>.

Mintel (2020). Hobbies and Interest – UK – February 2020. Retrieved from Mintel Reports database.

Misra, A. (2016). Use of Windows Presentation Foundation and Windows Forms in Windows Application Programming. *International Journal of Advanced Research in Computer Science*, *7*(7).

Nathan, A. (2010). *WPF 4 unleashed*. Pearson Education.

Orchard, M. T., & Bouman, C. A. (1991). Color quantization of images. *IEEE transactions on signal processing*, *39*(12), 2677-2690.

*Overview - Windows Forms .NET Framework | Microsoft Docs*. (2017). Retrieved November 25, 2020, from <https://docs.microsoft.com/en-us/dotnet/desktop/winforms/windows-forms-overview?view=netframeworkdesktop-4.8> .

*Overview of .NET Framework | Microsoft Docs.* (2020). Retrieved November 23, 2020, from https://docs.microsoft.com/en-gb/dotnet/framework/get-started/overview.

Parker, J. A., Kenyon, R. V., & Troxel, D. E. (1983). Comparison of interpolating methods for image resampling. *IEEE Transactions on medical imaging*, *2*(1), 31-39.

Parthipan, V. (2017). Image Down-Scaler Using the Box Filter Algorithm.

Patil, M. (2018). Interpolation Techniques in Image Resampling. International Journal of Engineering and Technology. 7. 567-570. 10.14419/ijet.v7i3.34.19383.

PCStitch: Premier Cross Stitch Software. (2016). Retrieved November 19, 2020, from http://www.pcstitch.com/PCS10/PCStitch10.aspx.

RTO (n.d.). RTO Iris Flower Cross Stitch Kit-4-inch x 4inch, 14 Count [Online image]. Amazon. <https://www.amazon.co.uk/RTO-Flower-Stitch-Kit-4-inch-4-inch/dp/B00EW4NCEG>.

Sells, C., & Weinhardt, M. (2006). *Windows Forms 2.0 Programming (Microsoft Net Development Series)*. Addison-Wesley Professional.

Setiabudi, D., Isa, S. M., & Iswanto, B. H. (2017). Digital color classification for colorful cross stitch threads using RGB+Euclidean Distance and LAB+CIE94. Proceedings of 2016 International Conference on Information and Communication Technology and Systems, ICTS 2016, 150–156. <https://doi.org/10.1109/ICTS.2016.7910290>.

Soumyasch (2007). The .NET Framework stack [Online image]. Wikipedia. <https://en.wikipedia.org/wiki/.NET_Framework_version_history#/media/File:DotNet.svg>.

Xu, J. (2010). Practical WPF Charts and Graphics. In *Practical WPF Charts and Graphics*. Apress. <https://doi.org/10.1007/978-1-4302-2482-2>.

Youssef, A. (1999). Image downsampling and upsampling methods. *National Institute of Standards and Technology*.

This report uses the APA notation style: <https://apastyle.apa.org/style-grammar-guidelines/references/examples>

It is important that you reference correctly, if you are not sure, ASK! You must be consistent, check your work! All references in this list must be listed alphabetically by author**. If you do not refer to them in the body of your report, then they should not be here**. Remember that you should be using references throughout all your chapters. For example, in your testing chapter you can reference different testing techniques. In your evaluation chapter you can refer to different ways in which evaluations are carried out. There are even sources available on how best to carry out a self-reflection. There is no rule as to how many references you should be including in your report. However, as a baseline, a good aim would be 25-30.

You should use a reference manage to manage your references throughout the whole project, it will also help you insert them into you report.

Please note that this chapter heading is NOT NUMBERED. This is not a mistake.

# Appendix 1 – Project Proposal

## Computing Degree Project Proposal

|  |  |
| --- | --- |
| **Student Name: Lili Veszeli** |  |
| **Course: Computer Games Development** |  |
| **Project Title: Cross Stitch Pattern Generator** |  |

## Project Context

I would like to create a computer software that could convert a picture to a cross stitch pattern. There are other software like this available online for free or for a price. (Chestnut Pens, 2020; Klein, 2017) This would be an interesting project that involves all the techniques I learnt at university.

## Specific Objectives

* Generate a pattern based on a picture
* Choose desired number of colours
* Choose size of pattern
* Edit pattern (for example change colour or draw on it)
* Save the pattern at the end

## Potential Ethical or Legal Issues

Copyright issues. The software might be too similar to one previously created. However it would not be illegal to do it for an educational purpose (Copyright, Designs and Patents Act 1988).

Some users could abuse the software to make patterns from copyrighted images/characters and sell them, which would be illegal.

Some users could make patterns with disturbing content, which might be seen by children if the patterns are sold.

## Resources

Computer – available at university and also own laptop

Visual Studio – available.

UI library for C++ - there are some free options, like JUCE (Robinson, 2013)

OpenCV – available

DirectX – if needed available

## Potential Commercial Considerations - Estimated costs and benefits

The software would be mainly used by people who enjoy cross stitching, which can be done any time. Hobbies including arts and crafts are becoming more popular (Kelly et al, 2020), a huge rise was experienced during lockdown (The Guardian, 2020). There are people who enjoy it from all generations, so I am confident that it will continue to be practiced and hence the program will be used. With the help of the application, it is possible to make patterns from pop culture, so it might help to make it more popular among young people.

Cost of programming the software: it will take around 30 weeks (academic year), so the salary for a programmer for that period would be approximately £12750.

The software would be sold for individuals for a low price, £10-15, because most similar applications start at £50, and I think a casual cross stitcher would be more likely to buy a product with a lower price.

All similar free applications have a bad quality UI, so by making the UI of my program clean and compelling it will be more attractive for users.

It would need to be advertised in some way, so that would add more to the costs depending which way it is advertised. If used Facebook advertisement, it would cost £300 for a month.

## Proposed Approach

Writing Technical Plan – 1 week

Researching how to load an image using C++ and code it – 1 week

Researching how to convert the pixels in the image to the squares in the pattern and code it – 5 weeks

Researching how to draw the grid over the squares and code it – 1 week

Researching how to edit the pattern and code it – 10 weeks

Researching how to make UI and code it – 3 weeks

Researching how to save file and code it – 1 week

Testing the application – 1 week (Testing would be involved every stage, but a more comprehensive testing period would be at the end)

Writing report - 4 weeks

## References

Chestnut Pens (2020) Cross Stitch Chart Creation Program. Retrieved from: <http://www.chestnutpens.co.uk/misc/index.html>

Copyright, Designs and Patents Act 1988, c.3. Retrieved from: https://www.legislation.gov.uk/ukpga/1988/48/

Kaehler, A., & Bradski, G. (2016). Learning OpenCV 3: computer vision in C++ with the OpenCV library. " O'Reilly Media, Inc.".

Kelly, C. M., Strauss, K., Arnold, J., & Stride, C. (2020). The relationship between leisure activities and psychological resources that support a sustainable career: The role of leisure seriousness and work-leisure similarity. Journal of Vocational Behavior, 117, 103340. Retrieved from: https://doi.org/https://doi.org/10.1016/j.jvb.2019.103340

Klein, T. (2017) Cstitch. Retrieved from: <http://cstitch.sourceforge.net/index.shtml>

OpenCV (2019) Load, Modify, and Save an Image. Retrieved from: <https://docs.opencv.org/2.4/doc/tutorials/introduction/load_save_image/load_save_image.html>

Robinson, M. (2013). Getting started with JUCE. Packt Publishing Ltd.

The Guardian (2020) Hobbycraft reports 200% boom in online sales since start of pandemic. Retrieved from: https://www.theguardian.com/business/2020/aug/03/hobbycraft-reports-boom-in-online-sales-since-start-of-pandemic-crafts-coronavirus.

Wang, M., & Lai, C. H. (2016). A concise introduction to image processing using C++. CRC Press.

Appendix 2 – Technical Plan

## Computing Project Technical Plan

**Name:** Lili Veszeli

**Course:** Computer Games Development

**Supervisory Team:** Games Team

## Title

Cross Stitch Pattern Generator.

Summary

I will create a computer application that will generate a cross stitch pattern based on an image. The user would be able to modify the pattern, for example change colours, change stitch location, change overall size. The pattern can be saved and printed. I will also have to a make a User Interface.

The difficult part in this project will be figuring out how to edit the pattern, making sure that the pattern looks right and uses the correct colours. I will need to research different methods to create these features. Working with C# will also be challenging since I am more familiar with C++.

I will use the Waterfall model as my development methodology, since we have set deadlines and once we submitted the required material we can’t change it. Some tools from Agile will be used too, like Moscow; this can make sure that all the essential features will be complete (Waters, 2009). Testing will be done too to make sure the program works as expected.

## Deliverables

The main product of this project will be a Windows based computer software that will be able to generate a cross stitch pattern from a picture. Other deliverables will be the report and poster.

## Constraints

The most serious constraint is the deadlines. It might be the case that the application won’t have all the features because of the limited time available for development.

Another constraint could be lack of resources. I might not find specific enough resources about how to create different features for my project.

The methodology used in my project could be a constraint. Using the Agile methodology is not possible since we can’t change our project after submitting the proposal. This might not be flexible enough for the project.

## Key Problems

The main difficulties will be

* to figure out how to edit the generated cross stitch pattern. I am not familiar enough with WPF to know how it could be done, so I will need to research it more.
* to design an efficient and clean user interface. I will need to research examples for it and experiment with my UI.
* to figure out how to work with pixels in WPF.

## System and Work Outline

The project will be developed will be developed using WPF in C# and XAML. The UI component will be made using WPF’s built in drag and drop UI editor. This can be further customized with XAML. The behaviour of the UI and the backend will be programmed in C#. Visual Studio will be used for the development environment. GitHub will be used for version control and backup.

Features of the program using Moscow prioritization:

|  |
| --- |
| **Must have:** |
| * A Windows application that can convert a picture’s pixels to bigger squares. * Basic UI. * A grid over the picture/pattern. * The ability to save it and print it. |
| **Should have:** |
| * The ability to resize the pattern. * More advanced UI. * Saving it in a PDF. * The ability to edit the pattern. * The ability to reduce the number of colours used in the pattern. |
| **Could have:** |
| * The ability to use real thread colours. * Zoom in while editing. |
| **Won’t have (but Would like in the future):** |
| * Other types of stitches. * Representation of how the pattern will look once finished. * Blended colours. |

(Agile Business Consortium, 2019)

Regarding WPF, I will need to research how images could be loaded in, how to make a modern UI and how to manipulate graphics (Sells & Griffiths, 2007).

I will also need to research how to navigate Windows folders, save and print files (Microsoft Docs., 2020).

## Project Activities



## Risk Analysis

|  |  |  |  |
| --- | --- | --- | --- |
| **Risk** | **Severity** | **Likelihood** | **Action** |
| My computer breaks | Severe | Small | Get it fixed – would hinder development by a few days.  Alternatively, work in the games lab – good solution, but might be difficult because of the limited availability this year. |
| Not being able to implement every planned feature. | Low | High | To be able to demonstrate my program it only needs the most important features, which are highly likely to be implemented. Using Moscow will help to ensure this. |
| Loss of data | Severe | Low | Making copies of the project every time I add to it, and storing it on a cloud based server. |

## Options

As an alternative to the Waterfall method, Agile could be used. However, because the deadlines for the deliverables are already set it makes the development less flexible. Main features cannot be changed either. Since I know what kind of features I want, I won’t have to do major changes to the program. It is also harder to use agile techniques if developing alone. Because of these constraints the best suited project management methodology for this project is Waterfall (Balaji & Murugaiyan ,2012).

An alternative I considered to develop the program in was Windows forms. However, WPF is a more modern and more flexible version of Windows forms, with more customization options (“WPF vs. WinForms”, 2020*)*. In WPF the UI editor is in XAML, with a drag and drop menu.

Another option could be DirectX 11. It would need a Windows library to be able to make the UI and it would be unnecessarily complicated to make the pattern using shaders.

Windows is the target environment. This program can be used most effectively on a computer, in a mobile app version the user wouldn’t be able to use the features properly. Later on, the project could be expanded to macOS and Linux, but this is not part of the objectives now because of the time limitation.

## Potential Ethical or Legal Issues

Legal issues might arise regarding the copyright law (Copyright, Designs and Patents Act 1988). If the program turns out to be too similar to one previously made, it could be a problem. However, this can be avoided by writing original code, not copying code that isn’t allowed to be freely used, and designing my own UI and mechanics of the application.

## Commercial Analysis

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Factor name** | **Description** | **Is this a cost or a benefit** | **Estimated Amount** | **Estimate of when paid** |
| Hardware | Computer to develop the project on. | Cost | Cost of my laptop - £900 | Start of the project |
| Software | Windows operating system | Cost | Cost of Windows 10 Pro - £220 | Start of the project |
| Salary | Salary for developing the program | Benefit | Salary for 30 weeks (academic year) based on average junior games developer salary - £12750  (Prospects, 2020) | Every month |
| Advertisement | Cost to advertise the product to the targeted audience | Cost | Facebook advertisement for 2 months –  £600 | End of project |
| Price | Price of the product | Benefit | £15 every time it’s purchased | End of project |

## Employability Contribution

This project will help me get a better understanding of making software for the public. It will improve my skill at UI design and my coding abilities in C# and XAML. I will gain experience in programming a Windows application. After writing the report I will be better at written communication.

The project can be expanded with a lot more features and could be marketed.

## References

Agile Business Consortium. (2019). *Chapter 10: Moscow Prioritisation*. Retrieved from: <https://www.agilebusiness.org/page/ProjectFramework_10_MoSCoWPrioritisation>

Balaji, S., & Murugaiyan, M. S. (2012). Waterfall vs. V-Model vs. Agile: A comparative study on SDLC. *International Journal of Information Technology and Business Management*, *2*(1), 26-30.

Copyright, Designs and Patents Act 1988, c.3. Retrieved from: <https://www.legislation.gov.uk/ukpga/1988/48/>

*Game developer job profile | Prospects.ac.uk*. (2020). Retrieved October 26, 2020, from <https://www.prospects.ac.uk/job-profiles/game-developer>

*PrintDocument.Print Method (System.Drawing.Printing) | Microsoft Docs*. (2020). Retrieved October 26, 2020, from https://docs.microsoft.com/en-us/dotnet/api/system.drawing.printing.printdocument.print?view=dotnet-plat-ext-3.1

Sells, C., & Griffiths, I. (2007). *Programming WPF: Building Windows UI with Windows Presentation Foundation*. " O'Reilly Media, Inc.".

Waters, K. (2009). Prioritization using moscow. *Agile Planning*, *12*, 31.

*WPF vs. WinForms - The complete WPF tutorial*. (2020). Retrieved October 26, 2020, from https://www.wpf-tutorial.com/about-wpf/wpf-vs-winforms/

# Appendix 3 – Title of Appendix

You may have one or more appendices containing detail, bulky or reference material that is relevant though supplementary to the main text: perhaps additional specifications, tables or diagrams that would distract the reader if placed in the main part of the report. Make sure that you place appropriate cross-references in the main text to direct the reader to the relevant appendices.

Do not blindly include all your code in the appendix or the body. Only include the parts you refer to in the report. You can put those parts either in the appendix or in the body (e.g. in the “Implementation” part).