

Marko Vähä-Erkkilä

# DEVELOPING VISUAL RUG PLANNING SOFTWARE

Helsinki Metropolia University of Applied Sciences

Master's Degree

Information Technology

Master's Thesis

4 November 2019

Author(s) Title	Marko Vähä-Erkkilä Developing Visual Rug Planning Software
Number of Pages Date	71 pages + 3 appendices 4 November 2019
Degree	Master's Degree
Degree Programme	Information Technology
Instructor(s)	Ville Jääskeläinen, Principal Lecturer
<p>The purpose of this study was to continue the software development started in my bachelor's thesis. Developing the first version of Ryijytin software started over a decade ago to assist in the design of Finnish ryijy (rug). The focus of my bachelor's thesis in 2007, was to test whether computer software could produce artistic content when producing a Finnish ryijy. The first version of the Ryijytin software was therefore a prototype to test the hypothesis. As a conclusion it was possible to use a computer software to assist the design of the ryijy and produce an accurate design when compared to the original picture to the final ready-made ryijy. The software development continued after that with incremental builds to create a design aid not only for professionals but also to hobbyists and beginners.</p> <p>The programming in the project was done with free Java Standard Edition, using JDK versions 5, 6, 7 and 8 and with NetBeans Integrated Development Environment. For the Ryijytin software version 2.0 development an Open JDK 12 and Apache NetBeans IDE 11 were selected.</p> <p>The Software development was broken into modules and for testing several finished ryijy were produced in different states of the development of the software. Based on the success of the first version of the Ryijytin software, it is possible to design a Finnish ryijy using a computer software. The added features in version 2.0 further increased the usability of the software. New and improved features make the design and editing of pre-calculated work charts even more flawless and easy. Designing and editing are easy with an intuitive and simple to use graphical user interface. As the software grow into a point where it was impossible to improve any further, Java evolved and Java libraries used in the project got deprecated or obsolete, a need for next software version came imminent. This study gives guidelines for next version of the Ryijytin software.</p> <p>The success of the Ryijytin software has kept itself waiting. A long development process and working as sole entrepreneur has stalled the success of the project as well as poor marketing. This thesis also discusses potential marketing strategies and designed new service concept as a resolution.</p>	
Keywords	Java programming, ryijy design software, software development, service design, image manipulation, textile.

## Contents

### Abstract

### Table of Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Designing and Producing Ryijy</b>	<b>5</b>
2.1	Ryijy	6
2.2	Designing Ryijy by Hand	11
2.3	Software requirements	12
<b>3</b>	<b>Programming Language</b>	<b>19</b>
3.1	Selecting Programming Language	19
3.2	Programming language dependencies	20
<b>4</b>	<b>Software development</b>	<b>26</b>
4.1	Defining program flow	26
4.2	First Version	28
4.3	Adding features	33
4.4	Testing Ryijytin in Action	43
4.5	Conclusion of Tests	57
4.6	End of Lifetime of Ryijytin 2007	60
<b>5</b>	<b>Ryijytin Software Version 2.0</b>	<b>61</b>
5.1	Version 2.0 Requirements	63
5.2	Software Design	65
<b>6</b>	<b>Discussion and Conclusions</b>	<b>68</b>
6.1	Marketing	69
6.2	Service concept for Internet	70
6.3	Future development and updating software	71
	<b>References</b>	<b>72</b>

### Appendices

Appendix 1. Ryijy instructions at Kotiliesi magazine from year 1931

Appendix 2. Pirkanmaa Sirkka Ryijy Yarn Colour Chart

Appendix 3. Ryijytin Software Versions Timeline

## 1 Introduction

The project began as a collaborative project with the Helsinki University of Art and design student Marjukka Vuorisalo. The idea for testing whether it was possible to create a computer software to assist in a ryijy design process came up in the summer of 2006 when I discussed a possible subject for my bachelor's thesis for EVTEK University of Applied Sciences with Marjukka Vuorisalo. She came up with a software engineering problem that was part of her master's thesis as a textile designer. Together we decided to test our limits and see how the project would reach its target either proving it impossible to use computer aided design for ryijy, or that it would be helpful and ease the process. There was no previous software solely for that purpose. Designers were using cross stitch design programs or Adobe Photoshop and even Microsoft Excel to help designing their ryijy.

The First task was to decide a name for the software and it came up quite easily with "Ryijytin" which translates roughly to "Ryijy-o-matic" in English. My bachelor's thesis covered developing first version of the software that was introduced in spring 2007. The first version of Ryijytin software, referred later to as Ryijytin software -07, was developed from artistic approach using the waterfall model. Ryijytin -07 didn't have any saving functionalities for a produced design, but outputted results to html, text, images etc. All progress was lost when a user closed the software. Only yarn charts were possible to input and saved for further use, but loading yarn charts were not yet possible. Pre installed yarn charts were hardcoded into Ryijytin software -07.

From Ryijytin software version 1.0.0.070307 (build 10.3.2007) was also derived Ryijytin software MOA 2007 version that included a step by step wizard and was limited to selected functions. Feedback were collected from an exhibition by including a "guest" booklet to Ryijytin software setup desk. The MOA version also became Ryijytin software -07 demo version with small alterations. As the development of prior Ryijytin software -07 version 1.0 is covered in my bachelor's thesis, I will here in my master's thesis only describe the starting point and software requirements. My bachelor's thesis is only available in Finnish, so this master's thesis has some parts in common, but rewritten in English with additional information gathered later in the development.

Development of Ryijytin software -07 continued using an incremental build model and included necessary bundling libraries with a software distribution and installer program. The installer program included setup for Ryijytin software -07 and added layer of protection to prevent software's use without permission. I decided to continue working mainly with Windows platform and added a windows executable program file so Ryijytin software -07 looks and feels like a native Windows desktop application. The jar is still executable in all platforms that are capable of running Java desktop applications. However, the Java runtime environment should be installed before the use of Ryijytin software -07.

The first test cycle of Ryijytin software -07 included producing a test ryijy in 2007. As a conclusion the samples and the work chart done with the Ryijytin software -07 did not support the hypothesis that one cannot create a ryijy containing artistic meaning with the help of computer. The samples proved that a computer can, indeed, show a picture of the finished ryijy with different yarn colours and their combinations in the finished product. Colour shades are so accurate in comparison with the original sketch that human eye cannot achieve a similar accuracy. The whole process takes only a few minutes, and is a significant improvement to traditional methods for experimenting with yarn mixing. Nevertheless, the success of the Ryijytin software -07 has been impossible to evaluate, since a similar program has never before been available. (Vähä-Erkkilä 2007: 72)

This study defines a new perspective to Ryijytin software -07. Purpose is not to evaluate whether a ryijy procured with the aid of Ryijytin is art or not, but from the perspective of design aid. More specifically, the target is to create a software to assist in industrial design and in designing ryijy that has never been seen because of the complexity of the design. A consumer nowadays doesn't want to buy old designs, but something that is produced from photographs taken at memorable moments of them or images that are otherwise meaningful to them. Ryijytin software was developed to assist at the most challenging and time consuming part of a ryijy designing process. To most, the hardest part is the selection of yarn colours based on a picture and usually years of practice is needed to be a master at selecting right colours. A software with an ability of picking colours in minutes and producing a work chart in short time compared to the process that could take weeks from human would make a huge difference.

The Ryijytin software version 1.0.0.070307 (build 10.3.2007) was first introduced to the public in Master of Arts exhibition in May 2007 (Moa Master Marjukka Vuorisalo, 2007). Due to schedule, the functionality of the program had to be limited (Vähä-Erkkilä 2007:72). This master's thesis follows software development from the first version to Ryijytin software version 1.0.3.6C (build 16.9.2018). In that time period new features were added to the point where the software was very difficult to improve any further. Improvements had been done since year 2007, but at that point Ryijytin software -07 has become difficult to manage, and Java libraries obsolete or deprecate. Also, the latest Ryijytin software -07 version 1.0.3.6C (build 16.9.2018) doesn't include all required functionalities for business use and those are impossible to be added to the current application. Problems were caused by Java itself as it evolved and changes to licensing policy of Java development kit by Oracle for commercial projects.

The next generation of Ryijytin software will be referred to from now on as Ryijytin software -20 or Ryijytin software version 2.0 to distinguish it from the previous version. The design process of the new software considers all the feedback and comments gathered during Ryijytin software -07 lifecycle. First commercial ryijy plans in were made in 2008, that gave some valuable information of the functionality of the software. Two more ready-made ryijy were produced in 2013 and 2014. The focus on Ryijytin software 2.0 development was set to a desktop application that is designed to be a vendible tool for hobbyists and professional users as well as for educational institutions. The aim was to develop an easy and fast solution to design a ryijy compared to traditional methods. Additionally, the target was to provide an accurate and editable conversion method from image file to ryijy work chart and calculate a reliable estimate of materials and costs.

The development focused on user experience as most important goals. For the best user experience it includes a graphical user interface so that the design would be as intuitive and easy to use as possible. The user interface was also designed for internalization. Internalization includes in the first phase Finnish and English versions. New languages should be easy to add afterwards.

This thesis is written in 6 sections. This section overviews briefly the software development process. Section 2 describes Finnish ryijy, explaining what ryijy, history of

ryijy is and how it has been traditionally designed. Section 3 defines requirements for the software. Starting from selecting a programming language and why Java was selected, and discussing what is required from software to be capable to convert picture to ryijy work charts. Section 4 presents the software design guidelines. How was the software design done to successfully fulfil its sole purpose? Section 4.1 describes the starting point when Ryijytin software -07 version 1.0 was introduced. Conclusions of Ryijytin software 1.0 (Beng version) are presented before continuing to section 4.3 and adding features to Ryijytin software -07. The software development was continued with the incremental build model. Chapter 4.4 presents several tests with different prototypes. Several ready-made ryijy were produced using Ryijytin software -07 to test functionality of software in different states of the development. Section 4.5 contains discussion of the obtained results. Results were used to evaluate working with Ryijytin software -07 and to test the capabilities of the software. Section 4.6 discusses why Ryijytin software -07 has come to the end of the line. The development of Ryijytin software -07 was ceased and developing a new version 2.0 begins in chapter 5. Section 6 discusses conclusions and outcomes of the project. Further development is discussed in chapter 6.1, chapter 6.2 discusses marketing of software, and chapter 6.3 introduces service concept for Internet usage.

The Ryijytin is a proprietary software owned by Marko Vähä-Erkkilä. The Ryijytin software could be licensed for use in companies, by individuals or in educational institutions from Koodikas that is a trade name for Marko Vähä-Erkkilä who is running the entrepreneurship as a sole trader (Yrityksen tiedot 2015).

## 2 Designing and Producing Ryijy

This chapter gives general background information about ryijy, its history, design and producing a ryijy by traditional methods as also background theory for computer aided designing of ryijy. To be able to design a ryijy one should understand what a ryijy is, what ryijy symbolizes, and it's meaning. The evolution of Finnish ryijy is covered briefly and main techniques are explained. From that point of view, a computer software has to be created to be usable aid in the design process.

Chapter 2.1 overviews history of Finnish ryijy and how it has evolved from bed covers or floor textile into functional piece of interior decoration of modern home. Well known ryijy by Akseli Gallen-Kallela called Liekki-ryijy (Figure 1), was created and realized for Paris World Fair 1900. It is a great example of Finnish design and lifted popularity of ryijy.

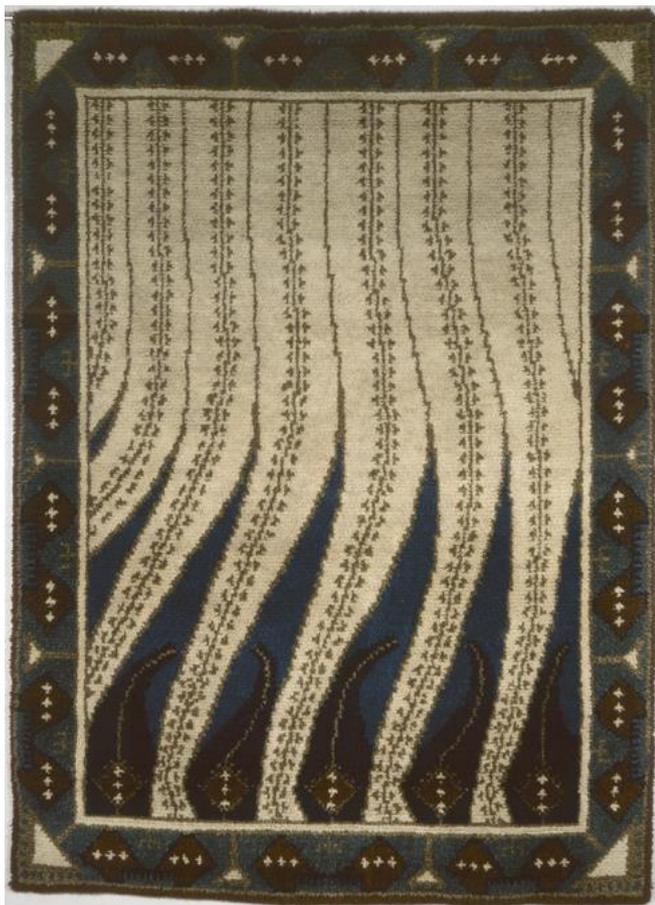


Figure 1. Akseli Gallen-Kallela: Flame-ryijy. 1900, original plan and realization 1899. Gallen-Kallela Museum. Picture: Gallen-Kallela Museum.

The evolution of ryijy began and continued throughout the 20th century. At the same time, the popularity of ryijy has had its ups and downs. Main methods and materials of producing a ryijy are explained briefly. Most commonly used different production processes are also described.

In chapter 2.2 is covered techniques needed to convert drafted image or photograph to computer understandable mathematical equations to produce work chart for ryijy. The same reasoning a human follows when creating ryijy design out from picture is followed programmatically. How the colours are chosen and reduced by the human had to be made by software. The decision path is explained and compared to computer decision making. Requirements for software are discussed in this chapter. The equations needed to calculate properties of ryijy are discussed and evaluated.

## 2.1 Ryijy

Finnish ryijy should not be mistaken as Oriental rugs which are kept on the floor. In Scandinavia and especially in Finland, ryijy has become wall textile and it is seen often at public spaces of corporations and government agencies. Ryijy has been symbol of Finnish folk craft and later it became example of modern design and art. The word ryijy cannot be translated to any other language. (Rosenstock 2013, 16)

The origin of the word ryijy, is in ancient Scandinavian word rya (Swedish, Danish: rya, Norge: Rye) that was used for knotted-pile quilt and translates to “thick cloth”. In Iceland, Norway and Denmark the word ru was used for fur that got loose from sheep at springtime. Such wool was used in the oldest ryijy. Ryijy wefts are usually made of linen or cotton. In oldest ryijy, weft were sometimes made of hemp. Warps are usually made of wool, and sometimes linen. (Asplund and Kumpulainen 1991, 13)

Technically ryijy is a textile that consists of ground weave, warps and knotted piles. (Pylkkänen 1967, 81) Ryijy is a three-dimensional (3D) weaving as it has length, width and thickness. One knot consist of from three to seven yarns, each of length is from 15 mm to 30 mm each. Ryijy has similarities with oriental rugs that are constructed with

similar knot technique. Traditional ryijy used to have only one or all of the same colour yarns per knot, but it evolved to have different colours in one knot as it shifted to a decorative textile. (Asplund and Kumpulainen 1991, 13)

### **History of Ryijy**

Ryijy is considered to be a typical Scandinavian textile although similar techniques have been used all over the world. Archeological findings shows that pile woven woolen weavings have been used in Scandinavia at least since the Bronze Age. Ryijy arrived to Finland along with Vikings circa 800 - 900 AD. (Asplund and Kumpulainen 1991, 13-14) In the beginning of its history in Finland, ryijy were used in western parts of country as sleeping blanket and covering on boats and sleighs (Pylkkänen 1967, 45). Ryijy was used in seashores and islands as boat covers against cold weather. Ryijy was especially excellent at sea because it kept you warm, and wet and salty water did not harden it. Ryijy spread inland with traders where it became bed covering in stone castles. Ryijy also became valuable payment method for trade. Ryijy eventually replaced fur as a bed cover also in other habitations and was adapted to floor covering and chair covers. (Asplund and Kumpulainen 1991, 13-14).

Use and appearance of ryijy changed during the 17th century. Knot piles were added to both sides of ryijy and were transferred from everyday use to dowry gifts and for use by very important guests. Double-sided knotted ryijy is an interphase from bedding cover to decorative item hanged on wall. Ryijy's heyday was from 1750 AD to 1830 AD and all remaining Finnish ryijy are from 18th to 20th century. During that time period decorative and picture ryijy was developed. Pictures included simple geometric figures and simplified plantations, animals and human figures. Yarns were dyed with vegetable-based dyes and most common colours were yellow, red and brown. (Louhio 1970, 15) Foreign trade brought new colours for example indigo blue and Spanish red (Asplund and Kumpulainen 1991, 14). In Figure 2 is shown typical sledge ryijy from western Finland made circa at the beginning of the 20th century.

After mid-19th century, production of ryijy decreased due to industrialization and cheap factory produced items. Time of national romanticism, Finnish arts and crafts, founding The Friends of Finnish Handicraft (in year 1879), ryijy design competitions and Akseli

Gallen-Kallela's Liekki-ryijy (see Figure 1 at the Paris World Fair) brought ryijy back to public attention. (Asplund and Kumpulainen 1991:16) Ryijy became fashionable in the 1920s and at the beginning of that era only old designs were used. (Asplund and Kumpulainen 1991:16)



Figure 2. Sledge ryijy from the beginning of the 20th century. Owner/Photography Marko Vähä-Erkkilä.

Starting from 1930s ryijy altered pictorial motifs and type of piles changed, knots were trimmed to different length and at the same time different colors were used on pile. The

borders between patterns were no longer sharp, but designs included variety of colours in finely tuned shades and different colour surfaces blended seamlessly. The look of an artistic ryijy evolved in to a more ascetic direction. (Louhio, 1970, 15-18, 20)

Many designers were painters who's knowledge of color and composition, lent to new designs of ryijy. Well-known colour ryijy also known as valööriryijy designers were for example Impi Sotavalta, Eva Brummer and Uhra-Beata Simberg-Ehrstöm. Ryijy changed from a homemade product, to interior decoration and art textile was made in close collaboration between the designer of the work and the weaver. Today's ryijy is based on modern pictorial motifs and colour hues that overlaps with modern interior design. (Louhio, 1970, 15-18, 20)

### **Modern Ryijy**

Ryijy has become part of interior decoration and has evolved since previous decades. New materials have been used from bicomponent yarn, where two different filaments are added including metal, optical fiber, fluorescent component etc. And plastic bags, copper or glass fiber have been utilized to make ryijy flame-resistant or even functional as a flame retardant fabric while simultaneously being used in decor. Modern ryijy production doesn't have any rules as methods and materials can be chosen freely. (Sinervo 2016)

### **Common Methods of Producing Ryijy**

Common methods of producing a ryijy can be divided to three main methods that are ryijy woven in looms or in frame, hand sewing to bottom canvas and tufting made ryijy. (Sinervo 2016) Difference between knots in different production methods are shown in figure 3. In looms and hand sewing, knots are tied around warps and in ryijy made by tufting yarn is pushed through base cloth. (Vuorisalo, 2007-2013)

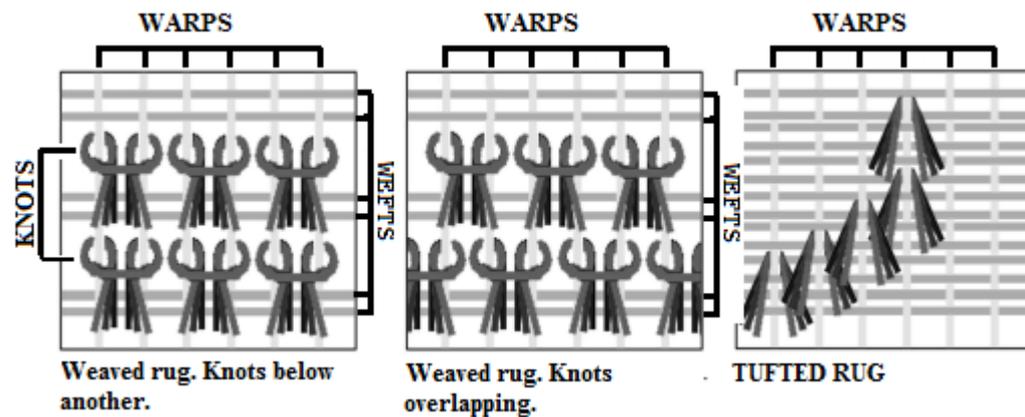


Figure 3. Difference between knots made by weaving, sewing and tufted (Vuorisalo 2007: JPG Image).

In ready-made ryijy it is hard to tell which method was used. Professionals can usually tell the method by visually observing. It is easier to determine methods by examining backs of ryijy. Sometimes the back side of ryijy is glued to secure yarns from loosening, making it harder to figure used method. (Sinervo 2016)

### Ryijy woven in loom

Rugs woven in the looms knots are tied to warps evenly to lines. Between knot lines are weaved weft. Generally knots are tied using Smyrna knot also known as Ghiordes knot and knots are tied to two warps. Knot lines are aligned one below another or overlapping as seen on Figure 3. (Vuorisalo 2006-2013)

Technically speaking, rug woven in the loom is closest companion to oriental rugs. Rug wefts are usually made of linen or cotton and in earliest rugs hemp is sometimes used. Warps are usually made of wool, and sometimes linen. Knots are traditionally made out of wool, but nowadays different materials are used from paper yarn and metal wires to optical fiber. One knot usually consist of three to seven yarns, from 15 mm to 30 mm long. (Asplund and Kumpulainen 1991, 13)

### **Ryijy stitched with needle**

Stitched rug is made by stitching to base fabric with needle and method imitates woven rug. Base cloth has been woven on the loom and rows of holes are left open on purpose. Knots will be stitched by needle into holes left at base cloth. The appearance of the finished work is nearly the same as in rug woven in the loom. (Vuorisalo 2006-2013)

### **Tufted Ryijy**

Tufted rug is produced by forcing the pile yarns through a base fabric that is stretched into a frame. Yarn does not form a knot, but is simply folded into half, wherein the yarn loops are at the “wrong” side as shown in Figure 3. Compressed air operated device or hand tufting machine could be used for attaching the piles to the fabric (industry uses large tufting machines in manufacturing rugs). One pile consists of four to seven from 16 to 45 mm long yarns and the material used in yarns is usually wool, rarely jute, and linen or thin paper yarn. Background fabric is usually made out of jute or linen. Tufted rug is finished with gluing from the wrong side to permanently attach piles. (Vuorisalo 2006-2013)

## **2.2 Designing Ryijy by Hand**

Traditionally designing a ryijy was started by drafting motifs used in ryijy design. Popular motifs were for example crown, bride, trees, animals, flowers, diamond and other shapes etc. All were reduced to basic silhouettes or stick figures. Yarn colours were then picked based on availability in designers own stock or in case of general design from commonly used colours. Commonly used colours were those that could be produced by dyeing yarns with plants or with other ingredients available at times. In most cases listing colours by describing them as light gray, black, green, light green etc. would be sufficient. Corresponding colours were picked from draft by visually inspecting. Each colour was then marked with symbol, letter or number and marked into checkered work chart and correspondingly marked list on its side or written on work instructions. (Vuorisalo 2006-2013)

## Work Instructions for Ryijy

“At this time, when the ryijy is woven and sewn almost every house and bought in addition, it is worth noting that ryijy is a very valuable product. Its value requires that it has to be done well, and using the proper material ... the patterns change according to fashion and taste, colours change over time and on sunlight, but well done work will retain its value for over a hundred years.” -Jalava (1931)

Instructions for ryijy have not changed much over the years. Appendix 1 has instructions for making your own ryijy scanned from a magazine called “Kotiliesi”, published in 1931. The above mentioned work instructions are almost identical to those used today. First figure in work instruction is a picture taken of finished ryijy. Second figure is drawing on checkered paper that has bordered areas, which are numbered and marked with different symbols. Bordered areas corresponds to the yarn colours that are described in the article. Article contains work instructions and gives verbal description of the yarn colour tones, so the weaver will know what colour of yarn to use in that area. Finally in the end of the article is a price of materials ordered via publisher. (Jalava 1931, 301-301,316)

### 2.3 Software requirements

Process for designing ryijy fits well to be implemented with computer software. The designing of ryijy from draft to work plan for a ryijy follows logical reasoning that is possible to implement with the computer program. The process has repeatable routines that are able to represent with mathematical equations. The final product has dimensions and transferable qualifies from draft to final work chart. (Vähä-Erkkilä 2007, 13) Software development for Ryijytin Visual Ryijy Planning software started in July 2006 by defining software requirements and defining terms used in software and equations for calculating final textiles, yarn consumption and estimated cost. Distance between two RGB colours in colour space called Euclidean distance can be used to select the closest matching colours. TEX number and user inputted preferred length of pile can be used for estimating cost of materials. (Vähä-Erkkilä 2007, 13-15)

First task was to define what software should be capable of doing. So I discussed with my partner and collected list of functionalities. Software requirements as my partner Vuorisalo (2006-2013) explained to me at the beginning of project in year 2006:

- 1) Software should be capable of opening image file produced by scanner, digital camera, etc.
- 2) Software should be capable of making checkering out of image based on dimensions and density of knots in final ryijy.
- 3) Into software should be capable to feed thread count of knot and maximum number of different knots on final textile.
- 4) Into software should be capable to feed colours from one or more ryijy yarn colour map.
- 5) Software automatically calculates colour blends for different knots of ryijy and produces checkered model of ryijy colours
- 6) Software includes possibility to print checkered image with symbols i.e. ready-made plan for vowing.
- 7) Software should be capable of calculating consumption by colour based on into the software fed colours, yarn Tex number and knot length.

(Vähä-Erkkilä 2007, 14)

From that point of view, I started to collect equations needed to processing image loaded into the software and create work chart out of it. Ryijy production process is compared in diagram at Figure 4 drawn by Vuorisalo (2007a) side by side traditional versus computer aided design. Previously described manual process is drawn along with computer aided process. In Figure 4 is also defined parameters fed to the software and what decisions to be made human versus computer. Ryijytin software -07 designed to follow flow as described previously as list and diagram in Figure 4. (Vähä-Erkkilä 2007, 22)

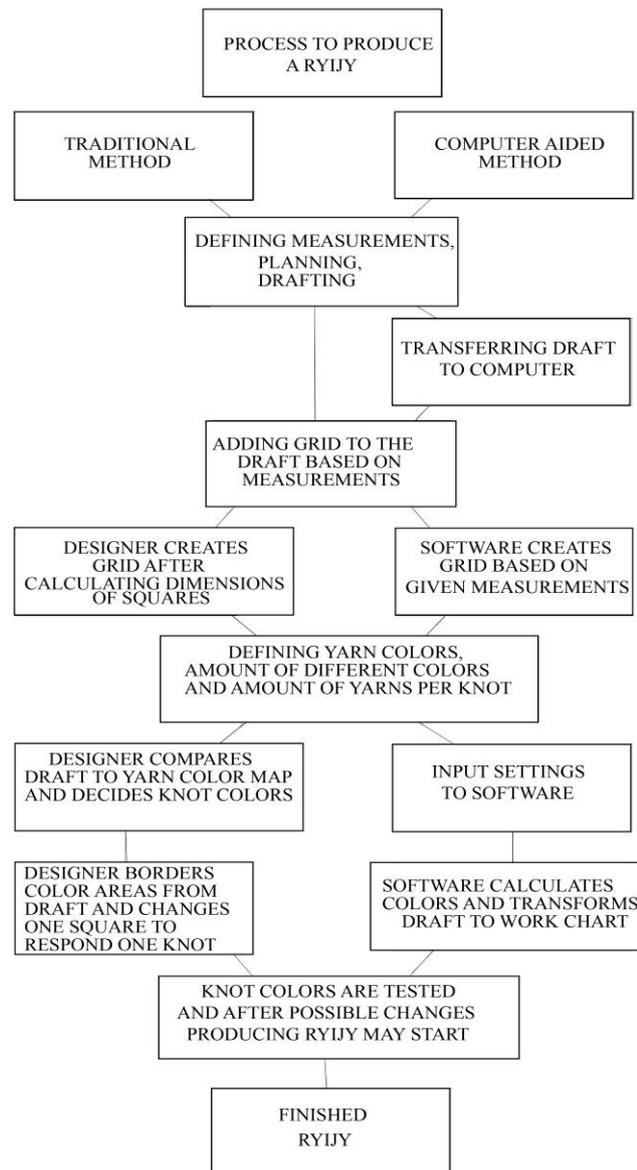


Figure 4. Traditional ryijy planning compared to computer aided design. (Vuorisalo 2007a)

### Distance Between Two RGB Colours

To be able to choose closest colours for yarns, I had to be able to compare two different colours and decide which one was the closest match. A formula used widely in mathematics to calculate distance between two points is called Euclidean distance. (Vähä-Erkkilä 2007, 49-50)

$$\text{Distance} = \sqrt{(R_1 - R_2)^2 + (G_1 - G_2)^2 + (B_1 - B_2)^2} \quad (1)$$

Working with multidimensional RGB colour space gives the possibility to calculate the distance between two points in RGB space with Euclidean distance. To compare two different RGB colours in colour space needs to be simply calculate squares the difference of every colour and sums them. (Schade, 2015:10)

### **TEX number**

TEX number is the most important value of yarn. TEX number is given with yarn can be used for calculating yarn consumption. TEX number gives straight out how much one kilometer of yarn weights in grams. For example Tex 300 means that 1000 meters of that yarn weights 300 grams. The equation is therefore  $1 \text{ TEX} = 1 \text{ g} / 1000 \text{ m}$ . To make it a bit more complex, yarn TEX numbers can be for example TEX 2 X 400 which means that the yarn is made from two TEX 400 thread plied together. Said yarns weight can be calculated with simple multiplication 2 times 400 equals 800 and therefore yarn in question weights 800 g / 1000 m. (Sinervo, 2017) When developing input for TEX number had to take into account the possibility that TEX number has more characters than integer number.

### **How to Calculate Yarn Length and Number of Yarns**

At the beginning of the Ryijytin software -07 development I believed that one TEX number per yarn colour chart is what is needed. However, Vuorisalo (2006-2013) corrected me in that every yarn colour could have its own TEX number and therefore the TEX number should be able to enter the Ryijytin software -07 yarn colour chart separately for each colour. The first prototype introduced setting TEX number for yarn colour chart as for testing were used only yarns from one manufacturer. Later in the development that was changed so that each thread colour had its own TEX number in order to use different yarns together. (Vähä-Erkkilä 2007, 32-33)

The Ryijytin software -07 is fed with TEX number of yarn and the pile length as straight yarn. The thickness of knotted pile is approximately reduced to half of given pre-cut length of yarn in ready-made ryijy. Therefore thickness of ryijy would be half the length of pre-cut yarn minus the length that goes to knot. Knot is simply tied around warps as

shown in Figure 2 and final thickness is subject to change depending how tightly knot has been tied. (Vuorisalo, 2006-2013) Yarn length can be calculated simply by dividing yarn weight by TEX number. Equation for getting yarn consumption in meters is therefore:

$$\frac{(1000 * \text{Yarn weight (g)})}{\text{TEX}} = \text{Yarn length (m)} \quad (2)$$

Because TEX number gives yarn weight in kilograms per 1000 meters yarn weight has to be multiplied by 1000 before dividing it with TEX number. With overall length of yarn can be calculated number of pieces with equation:

$$\frac{\left(\frac{1000 * \text{Yarn Weight(g)}}{\text{TEX}}\right) * 100}{\text{Length of knot (cm)}} = \text{Number of yarn pieces to cut} \quad (3)$$

Where the length of the knot is a straight yarn and not the thickness of the ryijy. The number of pieces can then be used to cut a sufficient number of pieces of yarn for the ryijy. The yarn is sold in batches of 100 grams. By calculating the number of knots in each colour and multiplying with length of knot, the consumption of the yarn in meters per colour can be calculated. Final consumption is calculated both in meters and in exact grams and rounded up to the nearest 100 grams to estimate the price of ryijy. Of course, the overall yarn consumption depends on how exact length of knot yarn is able to be cut. (Vuorisalo 2006-2013, Vähä-Erkkilä 2007, 32-33)

### **Image aspect ratio**

Original draft plan for ryijy has dimensions drawn in the proposed scale for final ryijy. In general, the woven ryijy is rectangular in which either the height or width is significantly larger than other dimension. In some cases ryijy can be also a square with the same width as the height. The image aspect ratio can be calculated by formula:

$$\text{Aspect Ratio} = \frac{\text{Image width (Pixels)}}{\text{Image Height (Pixels)}} \quad (4)$$

Where image width in pixels is divided by image height in pixels that gives image aspect ratio as a result. Image aspect ratio is needed in deciding ready-made ryijy dimensions. Aspect ratio in each should be nearly identical in order to use all image pixels for designing ryijy based on draft. (Vähä-Erkkilä, 2007, 29)

### **Number of Piles per Height and Width**

The dimensions of ready-made ryijy are needed to calculate for the number of knots at height and width. The number of piles at height and width can be calculated by multiplying the dimensions of ryijy by density of ryijy. Density of ryijy gives how many tied knots per centimeter resides in ready-made ryijy. Therefore equation ryijy height (cm) times knot density at height (cm) equals knots per ryijy width. And ryijy width (cm) times knot density at width (cm) equals knots per ryijy width. The dimensions of the finished ryijy are integer numbers and densities of tied knots are decimal numbers, therefore the result of multiplication is a decimal number. Resulting numbers are converted into an integer, because there may not be partial knots in ryijy, resulting in rounding.

An addition was made in December 2006 for requirements of Ryijytin software in case of losing image information. To avoid losing the image information in rounding, the image will be resized. After weaving the finished ryijy may shrink, therefore the height of image could be stretched. Likewise, the rows can be inserted in the height direction without affecting the final result. For technical reasons, the width of the finished ryijy may not change. When entering the dimensions of the finished ryijy, the user chooses whether to keep the scale of the draft image. If the user chooses not to maintain the aspect ratio, user will at the same time approve stretching the draft image. As long as the draft aspect ratio is selected to be preserved, any changes made are so small that they do not affect the dimensions of the finished ryijy. The sketch of the ryijy could also be drawn in

a different aspect ratio than the scale of the finished ryijy, whereby the user adjusts the finished ryijy dimensions to the wanted aspect ratio, and the image will be resized to the dimensions given. (Vähä-Erkkilä 2007, 30, 38, Vuorisalo 2006-2013)

### 3 Programming Language

This chapter discusses the choice of programming language and software dependencies, as well the methods utilized in this study. The Chapter is divided into two parts. First part explains why Java was chosen as the programming language and briefly describes the history of Java programming language. The second section discusses the Java technologies used in this study and the software dependencies that Java brings into consideration.

#### 3.1 Selecting Programming Language

My main consideration went around programming languages that I had studied so I didn't have to start learning new programming language from scratch. Still, I kept in mind the availability of free software development environments. In current time as we lived year 2006 were quite limited options what tools were allowed to use for commercial software projects for free. Java was, and still is cross platform object oriented programming language and have free development platforms. Also, NetBeans IDE that I chose for my project supports JavaScript, HTML, PHP, C/C++ etc. programming, not only Java. For even more convenient feature is that NetBeans has set of ready-made Swing components as windows, buttons and dialogs that are able to be customized. In NetBeans IDE you are able to design graphical user interface by dragging and dropping components into place. After placement, all components are able to be fully customized. Features that save lots of programmer's time and still gives ability to make everything look and feel as intended in user interface design. These were the main reasons I chose Java and NetBeans IDE for my project and kept them in further versioning. (Vähä-Erkkilä 2007, 16-17)

#### History of Java programming language

The language was initially called "Oak" when its development started in 1991 at Sun Microsystems. Developing 1st version took 18 months with a team consisting of James Gosling, Patrick Naughton, Chris Warth, Ed Frank, and Mike Sheridan. Language was renamed "Java" in 1995 as name "Oak" was already in use. Between first initial

implementation in the fall 1992 and public announcement of Java in 1995 many more people contributed to the design and evolution of Java. (Schildt 2007, 3-9)

Java was initially developed for the consumer market where need for platform independent language that could be embedded in various consumer electronic devices. Before C programming language there was not one programming language usable for all tasks and one had to choose programming language based on programming task. From C programming language Java derives its syntax. Java is related to C++ which is a direct descendant of the C programming language. However C++ had been the most efficient and powerful programming language until World Wide Web and the Internet which led to Java's success. (Schildt 2007, 3-9)

### **NetBeans Integrated Development Environment**

NetBeans started as a student project in year 1996 by name "Xelfi" in Czechoslovakia, now the Czech Republic and is an open-source integrated development environment (IDE) available for Windows, MacOS, Linux and Solaris. NetBeans was the first Java IDE written in Java. NetBeans DeveloperX2 was released in the spring of 1999. (A Brief History of NetBeans). NetBeans were acquired by Sun in 2000 and Oracle picked up NetBeans as part of its 2010 deal for Sun. Oracle announced in JavaOne 2016 conference contributing the NetBeans IDE as a new open-source project within Apache incubator (Oracle Proposes NetBeans as Apache Incubator Project). Transition to Apache foundation completed during year 2018. Netbeans IDE 9 (incubating) was released in July 29, 2018, NetBeans (incubating) 10.0 announced on December 27, 2018 and Apache NetBeans (incubating) 11.0 announced on April 4, 2019 (Wielenga 2018a, 2018b, 2019).

### **3.2 Programming language dependencies**

Choosing a programming language brings up dependencies. This section introduces what dependencies Java presented. Java application programming interface (API) includes packages, classes and interfaces, along with their methods, fields and constructors. These prewritten classes gives enormous amount of functionality to a programmer. Ready-made packages in API includes packages for GUI programming, managing input and output

and more. By importing classes from Java API allows a programmer to use any method listed in the above-mentioned class. (Norton and Stanek 1996)

### **Java AWT and Swing API**

Because of the platform independent nature of Java its windowed features has to be flexible. Java AWT (Abstract Window Toolkit) is the original Java's API for developing graphical user interfaces or window-based applications. Components such as windows, buttons, dialogs etc. are created by calling native platform (operating systems) subroutine. Each component therefore look and feel like current systems own components. (Cadenhead 1999, 296)

AWT is rarely used as its platform dependent and heavy-weight nature after Swing was introduced. Swing was built upon AWT and it was introduced in 1997 to fix the problems with AWT. Swing is lightweight and its look and feel is unrelated to underlying operating system (Singh 2015). Swing can handle all platforms and has everything needed to build graphical interface. By using Swing is possible to change application look and feel from default Java theme to Windows OS or Mac OS look and feel.

(Cadenhead 2012, 170)

### **Internalization and Localization**

Java has support for internalization so that an application can be adapted to various languages and regions without rewriting software every time. Instead of hardcoded textual elements they are stored outside of the source code and retrieved dynamically. Adding new language can be done just translating text phrases hold by resource file without recompiling the whole program. Dates, currencies etc. appear in formats that conform the end user's region. (Oracle 2019)

### **Java and XML**

Abbreviation XML comes from words Extendable Markup Language. XML is meta-data language that is really simple and straightforward in nature. Whole XML is text-based and has only strict syntax. Meaning it defines the core language itself and provides

metadata-type framework. XML does not define tag sets nor strict language but defines markup tag sets that have meaning to parser. Therefore programmer has freedom to define which tag points to which data in software and is not tied to predefined phrases. (McLaughlin 2000, 1-3)

XML was paired with Java from its inception and therefore Java has most complete API's to use XML directly within Java code (McLaughlin 2000, 12). XML provides platform independent way to transfer and save data within Java software. Therefore XML files could be used to store configuration and settings. Characteristic of XML makes it possible to communicate between different modules of the application. Furthermore with XML it is possible to transfer data from application to application and even over network as it's not tied to any particular software. (McLaughlin 2000, 14-19)

### **Java Advanced Imaging**

Java AWT had serious limitations with image processing when I started my project. I found out that Java Advanced imaging (later referred as JAI) from Sun had its stable version 1.1.3 released in 2006 and provided essential classes for image handling. With using JAI it was possible to reduce time to develop applications. (Sun Microsystems 1999) Since Ryijytin version 1.0.0.5b (build 31.8.2008) Ryijytin software -07 was not able to start running without JAI classes. In start phase was added check whether JAI was present and in case it was missing software gave an error message and prompted for installation of JAI from Ryijytin software installation media. Ryijytin software then made necessary finalizing procedures and exited. Without JAI Ryijytin software was unable to open image files, so the software was incapable of process any image data. (Vähä-Erkkilä 2007, 57)

In further development JAI was bundled with Ryijytin software -07. As Oracle acquired JAVA in 2010 from Sun Java Advanced Imaging library was no longer developed and latest stable version is 1.1.3 dated October 2006. Therefore in software should instead be used standard java classes that had become faster and more reliable in image processing in comparison to JAI. Also many Java Advanced Imaging classes were merged and developed further within standard Java.

## **Bundling Java Runtimes with software**

The purpose of bundling java runtimes with software is to prevent situation in which software could not start because of missing or wrong runtimes. Even if Ryijytin software -07 would run with different version of Java runtime environment that with it was tested there is no guarantee that Ryijytin software -07 would function as intended. That is main reason why I was looking if it is possible to bundle JRE with my software. I soon found out that there was no easy way to do it so at the current time, and had to drop the idea. The only possible way to bundle runtimes with software was to copy all files from java runtime installation folder (several hundred of megabytes) and embed it to Ryijytin software -07. I therefore decided to include needed Java runtime with Ryijytin installation media and limit in NetBeans IDE that Ryijytin software -07 would run only with Java runtime that I had tested. That would mean limiting Java runtimes to Java 6 and 7 (between years 2006 – 2014) and if detected version were older include prompt to install from Ryijytin installation media.

My intent was to get Ryijytin software 2.0 done with Java 8 in year 2014 and release software as Ryijytin software 2015 (version 2.0) in the beginning of year 2015. That did not actualize at 2015 and further development was delayed even further after Oracle announced transitioning NetBeans IDE to Apache foundation in 2016 (Oracle 2016). I found out in 2018 that current version of Launch4j allows bundling JRE with software (Kowal 2017). Ryijytin software -20 was therefore planned to be used with bundled JRE.

## **Launch4j – Cross-platform Java Executable Wrapper**

While searching whether bundling Java runtimes with my software would be possible or not in late 2007, I came across with software called Launch4j. Launch4j is a cross-platform Java executable wrapper. Wrapped executable is Windows native executable and runs like any regular Windows program. The Java jar file is not extracted from wrapped executable when running executable. Wrapped executable can have custom icon (i.e. programs own icon) and it could show native pre-JRE splash screen until the Java application starts. With Launch4j is possible to bundle JRE with executable. Also if bundled JRE is not present Launch4j could search for from system within given range for

JRE and opens Java download page in case of error. Launchj4 supports both GUI and console applications. Also Launch4j passes command line arguments to Java software. Setting initial and maximum memory is supported dynamically in percent of free memory. Through system properties can be accessed to environmental variables, the registry and executable file path. Launch4j is free software and it may be used for commercial purposes. (Kowal 2017)

## **iText**

To solve the problem with printing, I searched the Internet and came across iText Java toolkit. iText is a Java library for creating, modifying, manipulating and setting properties for PDF files. Originally, iText (formerly rugPdf) was created to solve problem with printing an HTML document as in-house project at Ghent University for the student administration. As I already had noticed, it could be painstakingly hard to define how a simple HTML table would print out using different browsers or even with different printers. From Ryijytin software -07 it was rather easy to send HTML file to printer but without any control how it would print out. After exploring different file types and printing them I was convinced usability of PDF files over any other option. I discovered that Bruno Lowagie had had that exact same problem with his project at 1998 and he solved it by writing iText PDF library. I found iText in early 2008 and version was back then 2.0.6. Version that Bruno Lowagie had been developing with Paolo Soares, who joined the project in summer 2000. (Lowagie Bruno, 2007:5-6) The latest release that is available for commercial use is version 2.1.7 released 19 th December 2009. Further versions need either to be open source projects or purchasing a license. For legal reasons it is no longer recommendable to use iText 2.1.7 library in commercial applications. (iText PDF: Your JAVA-PDF library - History) Ryijytin software -20 would have to use different library for creating PDF files.

## **Apache FOP**

To solve the legal problem with using iText I came across with Apache FOP (Formatting Objects Processor). As I was planning to use XML on my Ryijytin software -20 Apache FOP suits my project. Apache FOP takes XML data and format it's to various file formats. Best output support goes to PDF format, but Apache FOP also supports ASCII, Postscript,

direct printer output, AFP, RTF and Java2D/AWT for display, printing and page rendering to PNG and TIFF. For input Apache FOP supports embedding number of image formats that include SVG, PNG, BMP, JPEG, Postscript and some TIFF formats. (The Apache™ FOP Project, 2016)

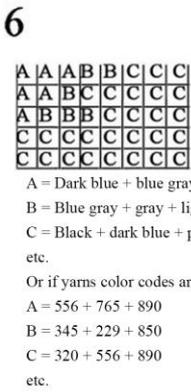
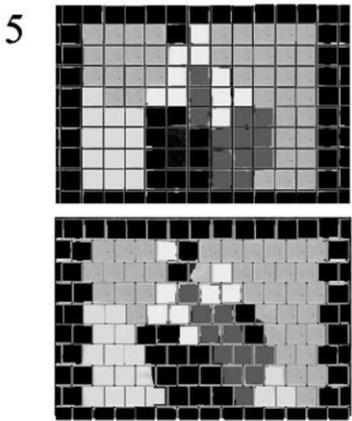
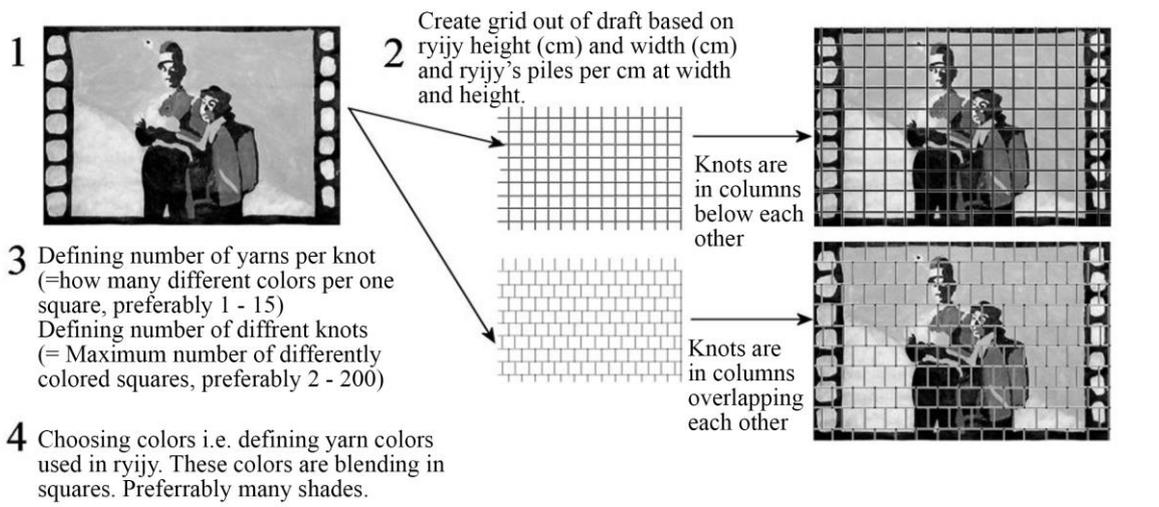
## 4 Software development

In this section is described software development as it was done between years 2007 and 2019. Development started with defining program flow and creating first prototype with waterfall model. Development of first version was covered in my bachelor's thesis. After first version development of Ryijytin software -07, an evolutionary value delivery model came. Ryijytin software -07 was broken into modules that were implemented and tested incrementally. Continued software design therefore followed incremental build model. Here we discuss what was done in prototyping, testing and planning for the next version of software.

Chapter 4.1 focuses on defining program flow. Chapter 4.2 describes first version of Ryijytin software -07. Ryijytin software -07 was starting point for my master thesis project. Chapter 4.3 focuses on adding features and incremental testing of Ryijytin software -07. Chapter 4.4 describes tests made with Ryijytin software -07, where ready-made ryijy was produced with the aid of Ryijytin software -07. Chapter 4.5 discusses about conclusions about tests and information gathered during the process. Chapter 4.6 ends development of Ryijytin software -07, giving explanation why development was ceased.

### 4.1 Defining program flow

Figure 5 is describes with images and text how ryijy design process goes as even engineer would get it. Phase 1 represents preferred draft for ryijy. In this case draft is hand painted image by Marjukka Vuorisalo of Pekka Puupää and Pätkä, fictitious characters in several very popular Finnish Pekka and Pätkä movies. After drawing the draft image is digitized using flatbed scanner and opened into Ryijytin software. On phase 2 to the images are added grid either columns in rows or columns overlapping each other depending on which sort of ryijy is selected to be designed. On phase 3 into software are fed how many colours per knot are used and how many different knots ready-made ryijy will be made up. Preferred amounts are 1 – 15 different yarn colours per one knot (default 4) and 2 – 200 different knots per ryijy (default 36).



7 Calculating yarn consumption per color based on given yarns TEX number (weight Gr/1000 m) and given knot length (usually 2 - 30 cm)

Software calculates color blend for each square from available colors given in part 4. Chosen color blend equate original drafts similar spot (=square) as close as possible.

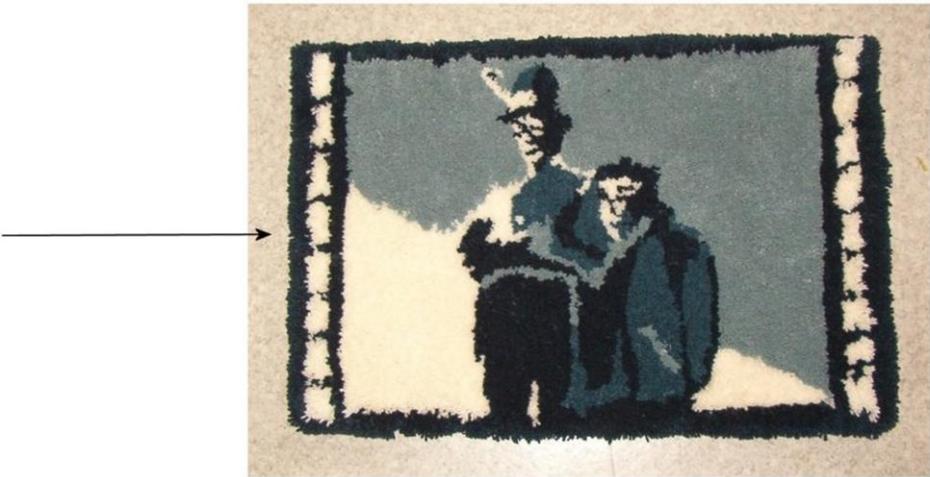


Figure 5. Ryjy planning process (Marjukka Vuorisalo:JPG file)

On phase 4 are yarn colours chosen and inputted to software. Colours are to be blended in different knots so preferably as many different shades as possible. On phase 5 software calculates colour blends for each square from available colours given in phase 4. Each colour mix presents as accurately as possible each of the corresponding squares in the original draft image. On phase 6 the work chart for ryijy is produced programmatically. Each colour blend is marked with unique letter and shown in table. On phase 7 software calculates yarn consumption based on given yarns TEX number and given knot length. With work chart created with software is produced ryijy.

## 4.2 First Version

Development versions prior version 1.0 were implemented using waterfall model between September 2006 and March 2007. Developing version 1.0 (BEng version build 10.3.2007) is covered in my bachelor's thesis. In this section is briefly explained functionalities of version 1.0 and what were planned to be next steps in software's development cycle. First public show of Ryijytin software was on Masters of Arts exhibition during time of my Bachelor's thesis was reviewed. Ryijytin MOA 2007 was version 1.0 with some limitations for functionalities, and some improvements in program logic and performance. (Vähä-Erkkilä 2007, 22-69)

### **Ryijytin version 1.0 (BEng version build 10.3.2007)**

Development of Ryijytin software started 25 th of September in 2006 as project was created in NetBeans IDE 5.5 (released on October 30th, 2006) that day. Development process leading to version 1.0 is covered in my bachelor's thesis. First official release version of Ryijytin software was version 1.0 build 5th of March in 2007 which software architecture is shown in Figure 6. The programming was done with free Java Standard Edition, using JDK version 1.5 update 11 and NetBeans IDE 5.5 BETA 2 with Java Advanced Imaging 1.1.3. (Vähä-Erkkilä 2007, 22-69)

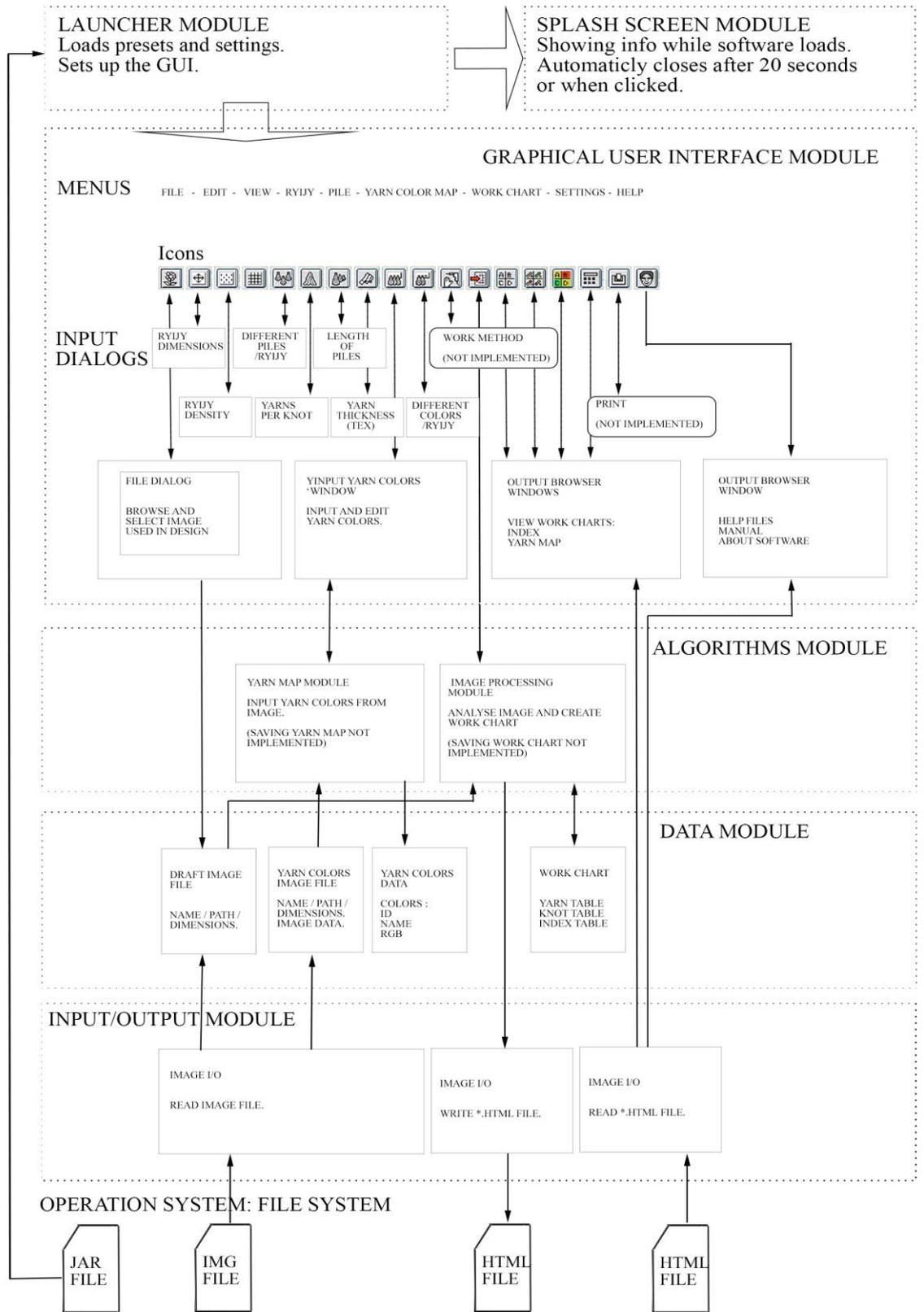


Figure 6. Ryjytin software Version 1.0 architecture.

Software design follows modular programming principles. In first version all settings were hard coded to launcher module. Launcher module set simple undecorated windows that contains image as splash screen. After that launcher module sets up graphical user interface and sets it visible. In that point Ryijytin software is up and running.

### **Masters of Art 2007 Edition**

Collaboration with textile artisan Marjukka Vuorisalo lead to Ryijytin software -07 to be on display at Masters of Arts festival (MOA07) at year 2007. Vuorisalo's statement was to overcome limitations on textile and find new design- and production processes. Sample pieces woven by Pirjo Pajusalo shown in Figure 7 were also displayed. (Vuorisalo 2007b).

Master of Arts 2007 exhibition version was derived from Ryijytin version 1.0 and had a step by step wizard for setting up final ryijy dimensions, preferred outputted amounts of different colours and different piles. Also, its options were reduced and you were able to test making of yarn colour chart, but you were only allowed to use the one that came preset with program to prevent anyone from using software for their own design without permission. (Vähä-Erkkilä 2007, 69)

The new feature included in Ryijytin MOA 2007 demo version (build 24.4.2007) was a small server. Purpose of server was to check whether software was already running on background and not start a new thread. In case Ryijytin software launcher gets response from server, after giving warning Ryijytin software ceases launching the new GUI, gives signal to already running application to bring its GUI to the front and exits the second application.



Figure 7. Ryijytin on display at Masters of Art 2007 exhibition. (Picture Marko Vähä-Erkkilä)

## Conclusion of Ryijytin software Version 1.0

Ryijytin version 1.0 (BEng version build 10.3.2007) was tested by making three pieces set of ryjy. Partial plan in Ryijytin software is shown in Figure 8 and ready-made sample pieces woven by Pirjo Pajusalo are shown in Figure 7. (Vähä-Erkkilä 2007, 62-65)

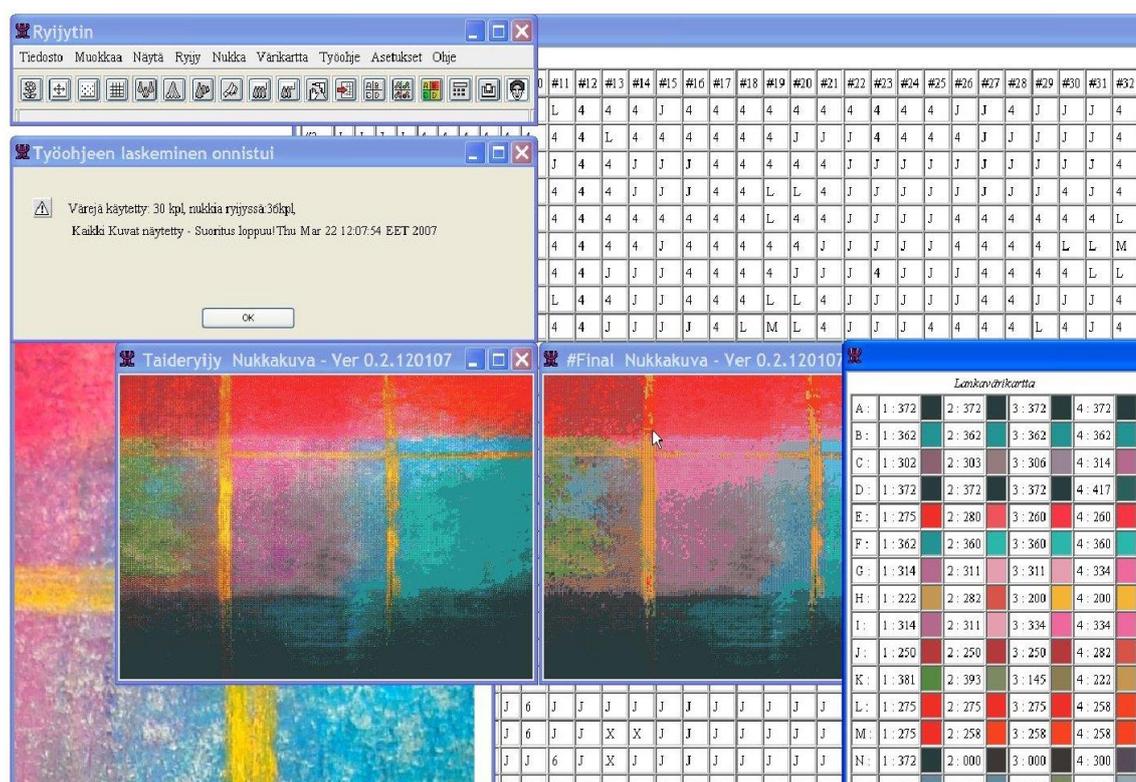


Figure 8. Ryijytin software has finished calculating work charts for test ryjy.

In conclusion, the samples and the work chart done with the Ryijytin 2007 -program did not support the hypothesis that one cannot create a ryjy containing artistic meaning with the help of computer. The samples proved that a computer can, indeed, show a picture of the finished ryjy with different yarn colours and their combinations in the finished product. Colour shades are so accurate in comparison with the original sketch that the human eye cannot achieve a similar accuracy. The whole process took only a few minutes and when compared with the traditional experimenting with yarns and their mixtures this is a significant improvement. Still, there is work to be done to make Ryijytin software a versatile tool for hobbyists and professionals. I would not describe Ryijytin version 1.0 software as visual rug planning software. To be able to visually design ryjy with Ryijytin software more control should be added over design. (Vähä-Erkkilä 2007, 72)

### 4.3 Adding features

After the first version succeeded to prove that the concept was working, started prototyping phase to test new functionalities. Prototyping took me almost a decade because of other work I did simultaneously. Some functionalities of software improved, some left out after the first version, some features were added and some requested features were tested and planned. Not every planned feature was added to Ryijytin software -07 as some features would have needed too much work to be added beforehand. Adding those features were left to Ryijytin software version 2. On this chapter I take a look at what was done after the first version was ready.

### Installation Program

The installation program was created starting 10th September of 2007 and finished during year 2008. Programming was done with NetBeans IDE 6.0.1. Purpose of installation program was to make Ryijytin software easier to set up on Windows based platforms. Installing to computer would have been done by uncompressing archive but installation program includes check for the right to use the software. Installation process was designed to be simple step by step procedure shown in Figure 8 that include 6 steps.



Figure 9. Installation program.

Installation starts with welcoming screen in step 1. In window is included welcome message, info and warnings. The program states that you are about to install Ryijytin software that is copyrighted. The next window in step 2 displays Ryijytin software's end user license agreement. The license agreement is loaded from text file to make it easy to maintain. Only after you click agree tick box you are allowed to click next button on installation program. In next window on step 3 it's asked to input registered name and serial number. License owners name was added to Ryijytin software to be used on copyright notes and to be included on pdf work charts produced by Ryijytin. For example,

to add text to work chart that is like “created by [License owners name] with Ryijytin version 1.0” and to be added into properties of pdf file as well with saved work etc. Product key added some complexity and made installation routine look more professional. Each licensed user is given a unique serial number based on name and some other attributes. At that point, software was not meant to be mass produced as every copy would need to be individually credited. In case of software piracy that would make easy to identify which copy of the software was copied illegally. Only after inputting correct registered name and serial number will you be transferred to the next screen.

In next window on step 4 you select Ryijytin software’s installation location. By clicking browse you open directory dialog in Figure 10 next to step 4. Using directory browser you can choose preferred installation directory. By default Windows Vista and 7 installations default folders are “Documents and settings”, and there “My Documents” folder is preselected and in windows 10 “Documents” folder. The correct folder is returned within Java calling system property user home. Step 5 is final window before Ryijytin software would be installed to hard disc. The selected path and directory for installation are displayed in final windows. Also, the name of license holder is shown so you may check you were really installing your own software.

As of result from running installation Ryijytin software has been installed to computer. Window in step 6 shows the directory and Ryijytin software version that were installed. As of now, none of the Ryijytin software licenses were sold as bundled to CD haven’t been need for installation program. Therefore no other have ever tested installation than me. Installation program is still usable with some alterations with Ryijytin software -20.

### **Saving and Editing Yarn Colour Charts**

Ryijytin version 1.0 (BEng version build 10.3.2007) didn’t have the ability to open and edit previous yarn colour charts. For built-in yarn colour charts, colours were handpicked with Adobe Photoshop colour picker tool and their RGB value were inputted to Ryijytin software version 1.0. After inputting desired colours to yarn colour chart, colours were saved to text file. Then from text file yarn colours were inputted into code to test software capabilities. Ryijytin version 1.0 (BEng version build 10.3.2007) had five different

sample yarn colour charts. Three of them were created from yarns of actual producers of ryijy yarns and the rest were some random colours for testing purposes. From built in yarn colour charts you were able to choose one to use for your own ryijy design. Once yarn colour chart was opened you were able to edit, add new colours to chart by inputting RGB values and delete colours from yarn colour chart, but not save any changes. (Vähä-Erkkilä 2007, 47-48)

Ryijytin software version 1.0.0.0 (build 9.4.2007) had option to choose from built in yarn colour maps or create new and save it to hard disc. And finally the ability to open saved yarn colour charts from hard disc. Ryijytin yarn colour chooser is shown in action in Figure 10. For fine tuning there is handles that you could alter the colour if you aren't fully satisfied to shade which Ryijytin software calculated. To be noticed that Ryijytin software can calculate only one solid colour that is the main colour of yarn. In actual finished ryijy will be more shades caused by light and shadows. Also by brushing the ryijy you could change the places of each yarn and alter the surface and change the look and appearance of a ryijy. (Vähä-Erkkilä 2007, 47-48)

Yarn colours from manufactures are provided as sample yarns tied to carton and named as shown in appendix 2. Depending on manufacturer, some have taken pictures of them and they are available for download from their website. From some manufactures you could borrow or buy actual handmade carton with yarns. For inputting yarn colours to Ryijytin software -07 from yarn chart a tool was built for opening an image and selecting area where yarn colour were picked. The mechanism that was used to open an image was derived from code that was used to open a draft for ryijy.

Into window were added code that allowed user to click a point and drag with mouse a rectangle around chosen area. From chosen area was used to calculate approximately average colour that would be used as yarn colour to be compared to draft used for designing ryijy. From discussions with Vuorisalo (2006-2013) during years 2006 and 2007 I learned that colour on the cut end of yarn is brighter than on outside of yarn and also that as a result of dying technique used to dye yarns make bit variety to shades of yarns. So I fine-tuned Ryijytin software -07 to calculate a bit brighter RGB value representing yarn colour when calculated from picture taken of yarn colour chart.

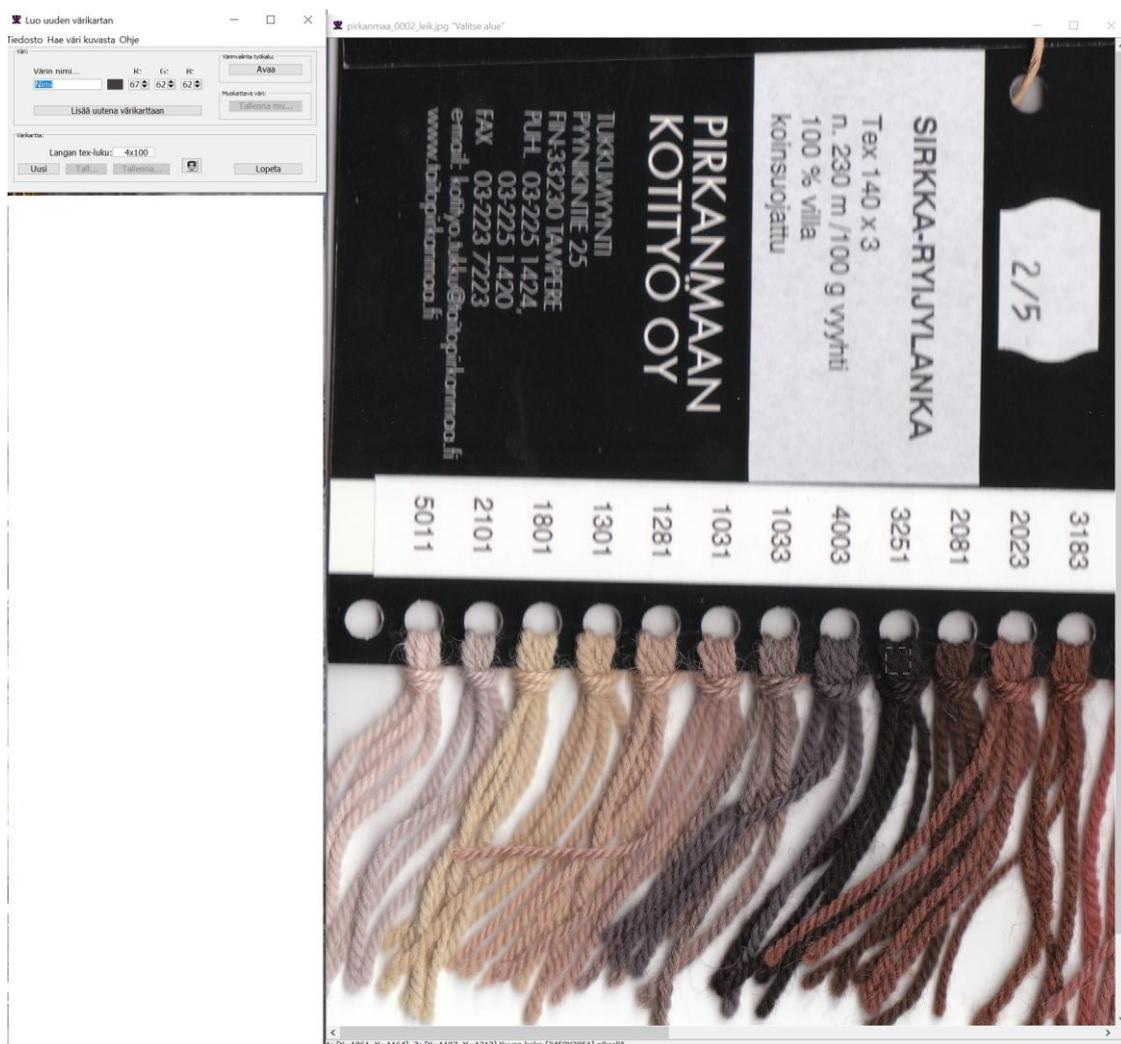


Figure 10. Inputting yarn colour to Ryijytin software.

### Saving Project and Work Chart

Ability to save project was essential for continuing from previous work. Starting from first version that calculated work chart for ready-made ryijy the work chart was automatically saved as HTML –files. HTML files were then used for showing resulted plan. Actual save function was not implemented to Ryijytin version 1.0 (BEng version build 10.3.2007). You got ryijy work chart as HTML –files that were automatically named with date and time to separate different files. Files reside on Ryijytin software temp directory. I am skipping version 1.0.0.1 that included only some improvements and corrections to errors and go straight to version 1.0.0.2. Into work chart menu of Ryijytin software version 1.0.0.2 (build 28.4.2007) was added drop down menu for saving work chart, knot chart and yarn consumption to HTML file and select with file dialog name

and path for files. Also Ryijytin software version 1.0.0.2 (build 28.4.2007) included function for saving project and loading previous project. Saving was implemented simply by writing object holding all necessary data to disk using Java file IO stream. Soon I realized that was not optimal, as changing object would cause error when trying to input saved data. Ryijytin software version 1.0.0.5G (build 1.10.2008) introduced project saving version 2 and finally starting from Ryijytin software version 1.0.0.5L5 (build 19.1.2009) both version 1 saved files could be read and converted to version 2 and used. Project saving version 2 was used until version 1.0.3.6C (build 16.9.2018) with some improvements done in different prototypes. Save project version 2 had some improvements made to be more backwards compatible in case of minor modifications to software. As still some restrictions to changing objects applied to version 2 also. From that point Ryijytin software was after loading saved project in the same state as before saving project. All settings, yarn colour chart, windows that were open and calculated work charts were exactly the same way as they were on display at the moment of saving project. User could continue working with same draft and previously inputted settings.

### **Printing Work Charts and Saving to PDF**

Ryijytin software version 1.0.0.5G (build 1.10.2008) introduced first prototype of printing. On first version you could set how many columns and rows per page print would have and select print. Setting rows and columns was crucial because HTML tables were used for printing work charts and there was no way to predict how many rows and columns would fit in one HTML page. Printing was done by saving work chart to HTML file and sending that to operation systems default Internet browser using Java AWT API. Different Internet browsers would then show HTML table on slightly different way. On this first prototype you could only print via your computer's default Internet browser as actual printing was not yet implemented. Ryijytin software version 1.0.0.5L2 (build 26.10.2008) improved functionality by adding “Preview in browser” –button to which the above-mentioned functionality was moved. To actual Print -button was added code for sending the HTML file to operation systems printer using JAVA Print API. Clicking Print –button opened systems print dialog where you could choose printer, page size etc.

Ryijytin software version 1.0.0.5L2 (build 26.10.2008) introduced first prototype of saving work chart to pdf using MPL library iText 2.1.3 by Bruno Lowagie and Paulo Soares. First version was able to output given part of work chart or whole work chart divided to number of pages given to Ryijytin software. Development of saving to PDF was continued on Ryijytin software version 1.0.0.5L6 (build 29.1.2009) and calculating how many columns and rows fit to one A4 page was added. Writing PDF file properties was added so when checking PDF properties via file system would show that Ryijytin software created the file and who was author i.e. owner of the license. Also some other info about file content and copyright etc.

Improved print dialog was added to Ryijytin software 1.2.2009 shown in Figure 11. Print dialog had selection for rows and columns per page for printing directly from Ryijytin software. Print dialog also had selection for parts of work chart that you would choose to be included on print. Printable parts were display image (image created by Ryijytin software based on work chart), work chart, knot chart (chart of yarn colours in different knots), yarn consumption and ryijy info (dimensions, density etc.). Option to save as PDF was also added to dialog.

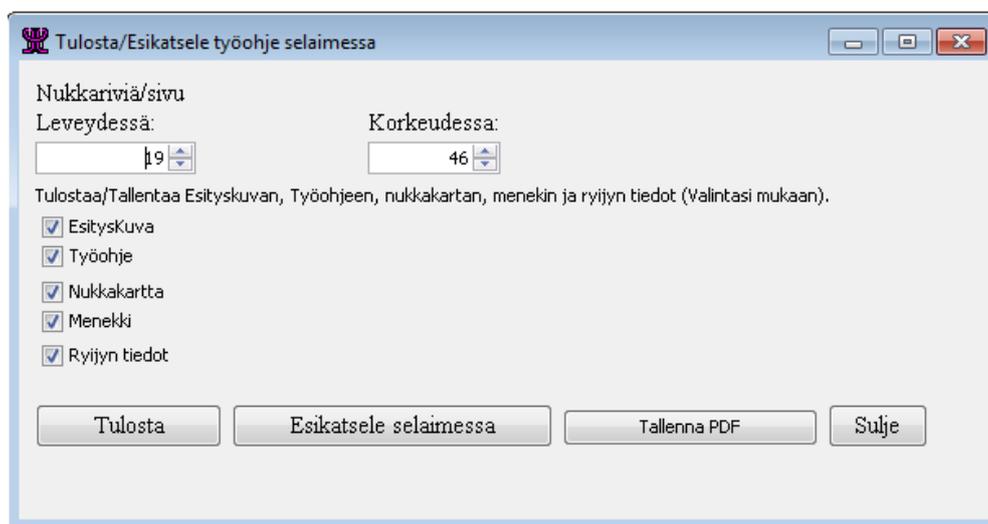


Figure 11. Printing work chart.

YaHP Converter (Yet another Html to Pdf converter) v1.2.1.9 (13/03/2009) was added to Ryijytin software version 1.0.0.6 (build 14.3.2009) to ease converting HTML files to PDF. YaHP 1.2.19 used Flying Saucer library as renderer. On Ryijytin software 1.0.1.1 (build 28.9.2012) YaHP converter was upgraded to version 1.3 (21.11.2011) that uses

third party libraries Flying Saucer (16.4.2011), iText 2.1.7, Apache log4j (1.2.16), JTidy (r938), Shanidom 1.4.17 and Jaxen 1.1.1. YaHP does all that I did manually when converting from HTML to PDF and more that I haven't had time to include to my project. That was my reason to choose over use of free library instead of doing all by myself.

After experimenting with different print methods available by Java or using alternate methods I was convinced that saving to PDF was the only way to be able to define how the work chart would print. Any other file type or printing HTML files with Ryijytin or browser didn't give control how would tables fit into page or how would they show up from printer. Printing directly from Ryijytin software was therefore removed and only option left was to output to PDF.

Starting from Ryijytin software version 1.0.2.9 save as PDF version 2 was introduced. More features and selections were added. Creating editing work chart windows in Figure 12 made possible to save even more precisely formed PDF files using graphics created with Java Swing API's JTable. YAHP was no longer used to form work charts.

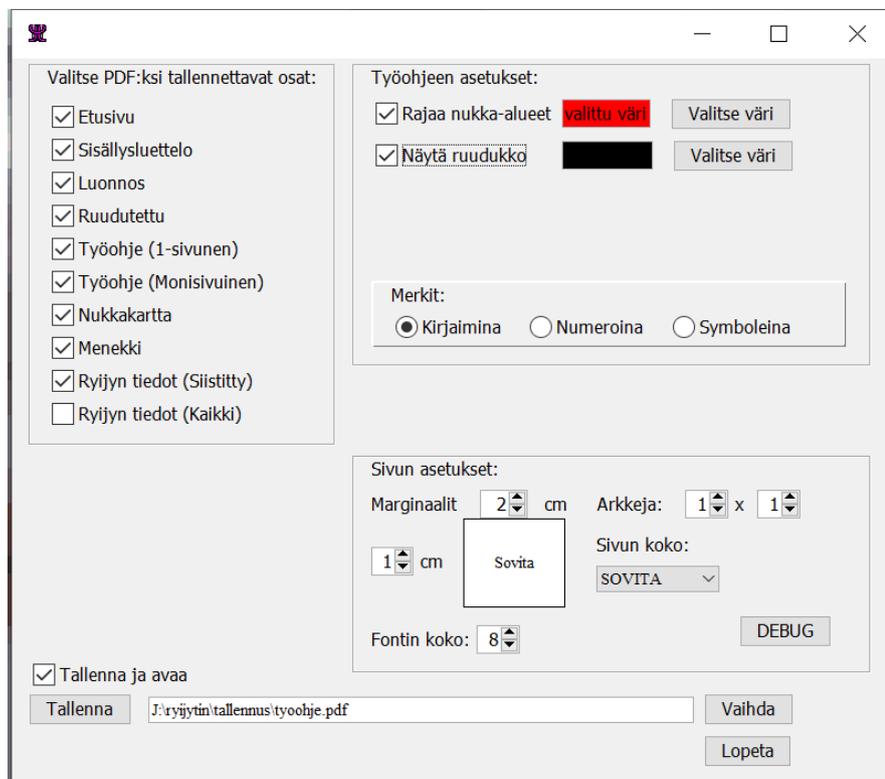


Figure 12. Save work chart as PDF Version 2

As Figure 12 shows you could choose to output PDF with several new elements. Elements included front page, table of contents, draft, checkered draft, work chart on one page or several pages, yarn colour chart for different knots, yarn consumption, clean or raw information of ryijy. To work chart was able to choose to draw border around same knot areas and choose border colour. Ability to choose whether grid was visible or not and changing grid colour was added. Selection for characters shown in work chart was added to choose between alphabet, numbers or symbols. For PDF file was added to choose page size and margins. Also choosing saving directory was made possible and choose whether saved file opens in operation systems default PDF reader. As conclusion I was quite satisfied with save to PDF version 2 and PDF files outputted with it. Output can be seen in chapter 4.4 as pictures of “Karhu” ryijy work chart are formatted with save as PDF version 2.

### **Using XML for configuration**

Ryijytin software version 1.0.1.3 (build 17.10.2012) was prototype for testing reading and saving properties to XML file. Instead of using hardcoded settings using JAVA XML API is possible to read and write settings to XML file. Even without having to know anything of XML syntax as JAVA XML API does formatting when writing XML file. XML therefore could be used store settings, configuration, even objects data. For example, if object has been changed from software’s previous version data still could be read from XML file and stored to object. Problem with writing objects directly to file using Java IO API. As changing the way Ryijytin software -07 saves its project would cause rewriting almost all the code for saving project it was not implemented. For upcoming Ryijytin version 2 was planned to implement saving project data to XML file to make Ryijytin software more transferable.

### **Editing Work Chart**

One of the biggest parts of the project was to enable editing work chart. Feature that would make Ryijytin to be actual visual ryijy planning software. Prototyping was done between May 2013 and November 2013 (versions 1.0.1.8 – 1.0.3.0). Starting point was

to link work chart and calculated yarn colour chart together. Previous versions of Ryijytin software were not able to open designs saved with Ryijytin software and edit, modify or change colours selected for knots. Also editing work chart haven't been possible. Ryijytin software version 1.0.2.0 (build 4.6.2013) introduced first version of work chart window implemented with Java Swing API's JTable class. First version of work chart window was clickable with mouse to select a cell. Clicked cells location and yarn colours were then displayed on screen. To Ryijytin software version 1.0.2.1 (9.6.2013) were added window for calculated yarn colour chart that was similar to inputted yarn colour chart that contains all available colours. Third window was added to work as link between two of these. Functionalities to windows were added until Ryijytin 1.0.3.0 (build 4.11.2013) which edit work chart windows are on Figure 13.

Largest window is the work chart window. Work chart window consists work chart as table with knot id numbers and knot colour as background. Selecting one ID from table could be done by clicking with the mouse. Selecting all the same ID's is done by pressing ALT same time as clicking with the mouse. Free selection can be done by pressing CTRL button and clicking with a mouse. In which case it doesn't matter whether or not cells are holding the same ID. After selection has done right clicking opens editing window.

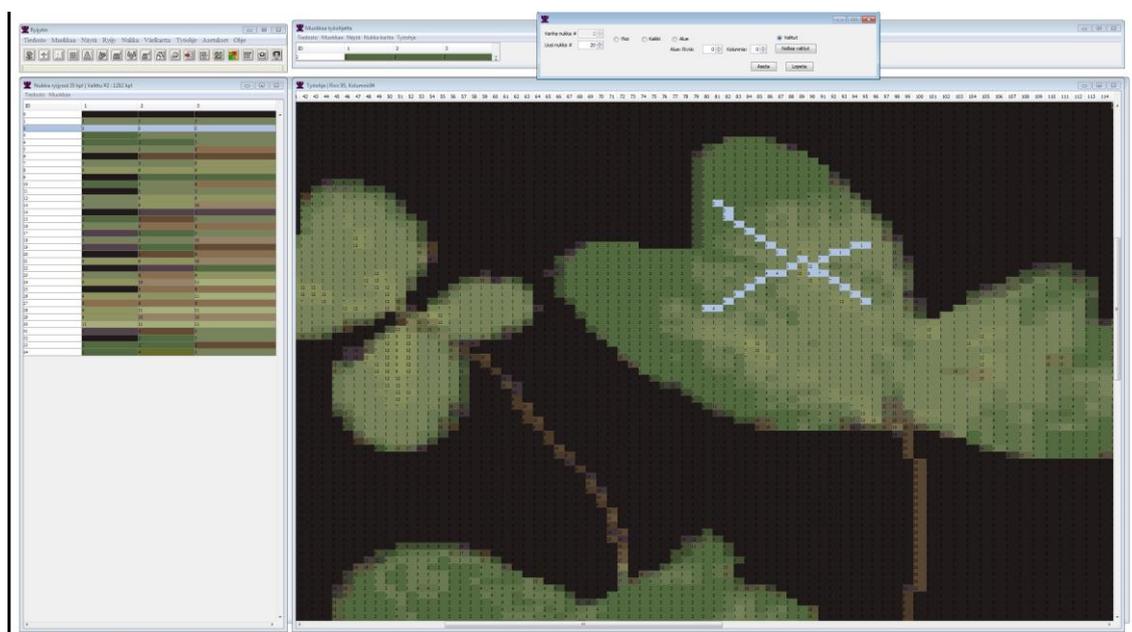


Figure 13. Ryijytin 1.0.3.0 (build 4.11.2013) Edit work chart windows.

Window on the left is yarn chart window. Yarn chart window consists of knot ID's and calculated yarn colours for each knot. By clicking any row selects both that row and from work chart window each cell that have the same ID. Third window on the middle is editing window that shows selected knot id and its yarn colours. Editing window bind other two windows together. In editing window you could change one or more colours inside knot. Changes are then instantly shown both in yarn colour chart and work chart. Upper right is dialog that opens by right clicking with mouse if you are changing one or more knot id from work chart to another. Right clicking the work chart tells Ryijytin software that you want to change that id to another and on dialog you can select one ID, or ID's in square area or all ID's. You can select to which ID current ID is to be changed by clicking yarn colour chart. After done you can apply changes and see instantly the change.

Ryijytin 1.0.3.0 (build 4.11.2013) made available changing the way work chart was displayed. You could choose to show background colour or without. Changing between showing ID's as numbers, letters or symbols was made possible. Option to show or hide grid was added and border between different knots could be set visible or hide. Colour for grid and border was set to red as default but that you could also change easily if needed. Different variables are shown in Figure 15 which is combinations of four different screen captures. All above is selectable also when saving work chart to PDF file as previously explained in chapter Printing Work Charts and Saving to PDF and shown in Figure 14.

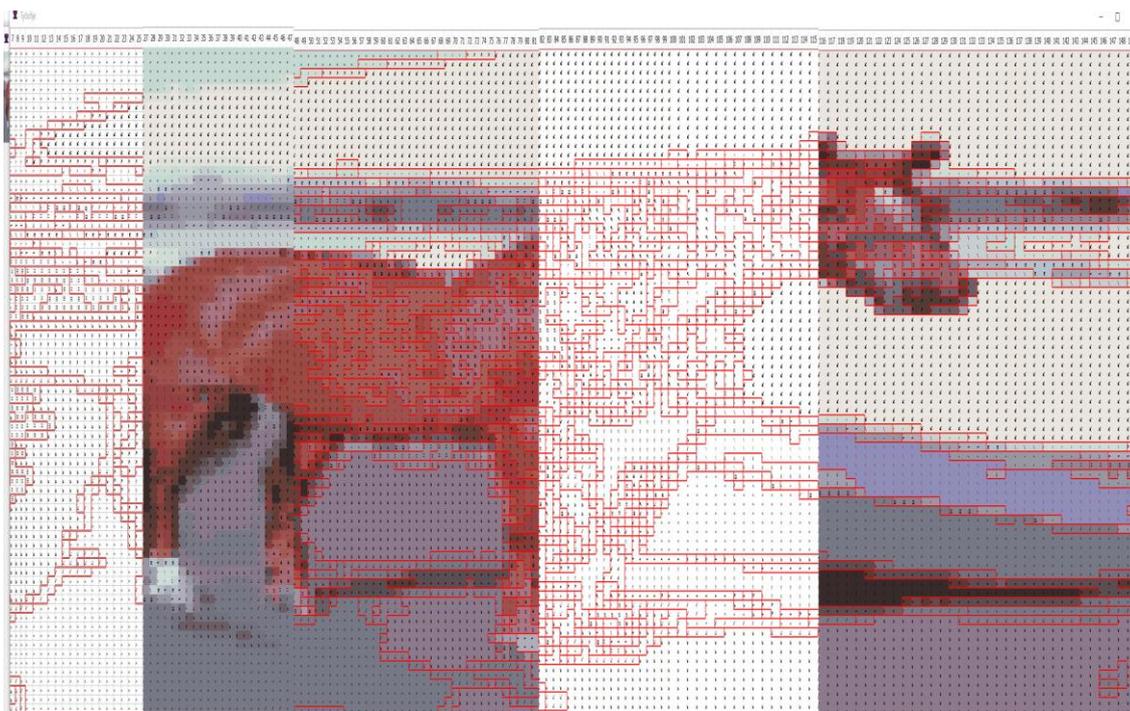


Figure 14. Work chart with different display modes.

#### 4.4 Testing Ryijytin in Action

In this section are several cases where Ryijytin software was tested by planning actual ryijy. With Ryijytin software made work chart was then produced ryijy by third party. First case is “Hevonen” ryijy made with Ryijytin version 1.0.0.5 that is quite early development version. Second case is “Karhu” ryijy designed with Ryijytin 1.0.2.9 that had been developed several years after first test. Third case is “Malta-06” ryijy designed with Ryijytin software 1.0.3.0 that is nearly as final version as development was done with version 1.0 descendants. Only a few flaws in Ryijytin software -07 suggested by Koivisto (2014) were fixed after that and updated some libraries.

##### “Hevonen” ryijy (23.9.2008)

Commercial work done with Ryijytin software were set of two ryijy designs made from customers pictures of horse in Figure 15 and horse head in Figure 20. Ryijytin version 1.0.0.5 (build 9.6.2008) was used for producing work chart for ryijy. Paper pictures used for design were typical photographs taken with film camera and printed to 10X15 size photo paper by Ifolor.



Figure 15. Paper photograph of horse (10 x 15 cm). Scanned with flatbed scanner to jpg. Original picture Silmu Peltonen.

Horse picture in Figure 15 was scanned to computer with flatbed scanner. Image file properties are width 3543 pixels and height 4134 pixels and density 300 dpi. After scanning picture was edited in Adobe Photoshop CS2 and people on background, manure

and other irrelevant details were blurred out of the picture. Edited image shown in Figure 18 was then opened in Ryijytin software. Several different backgrounds were tested before selecting final image for designing ryijy.

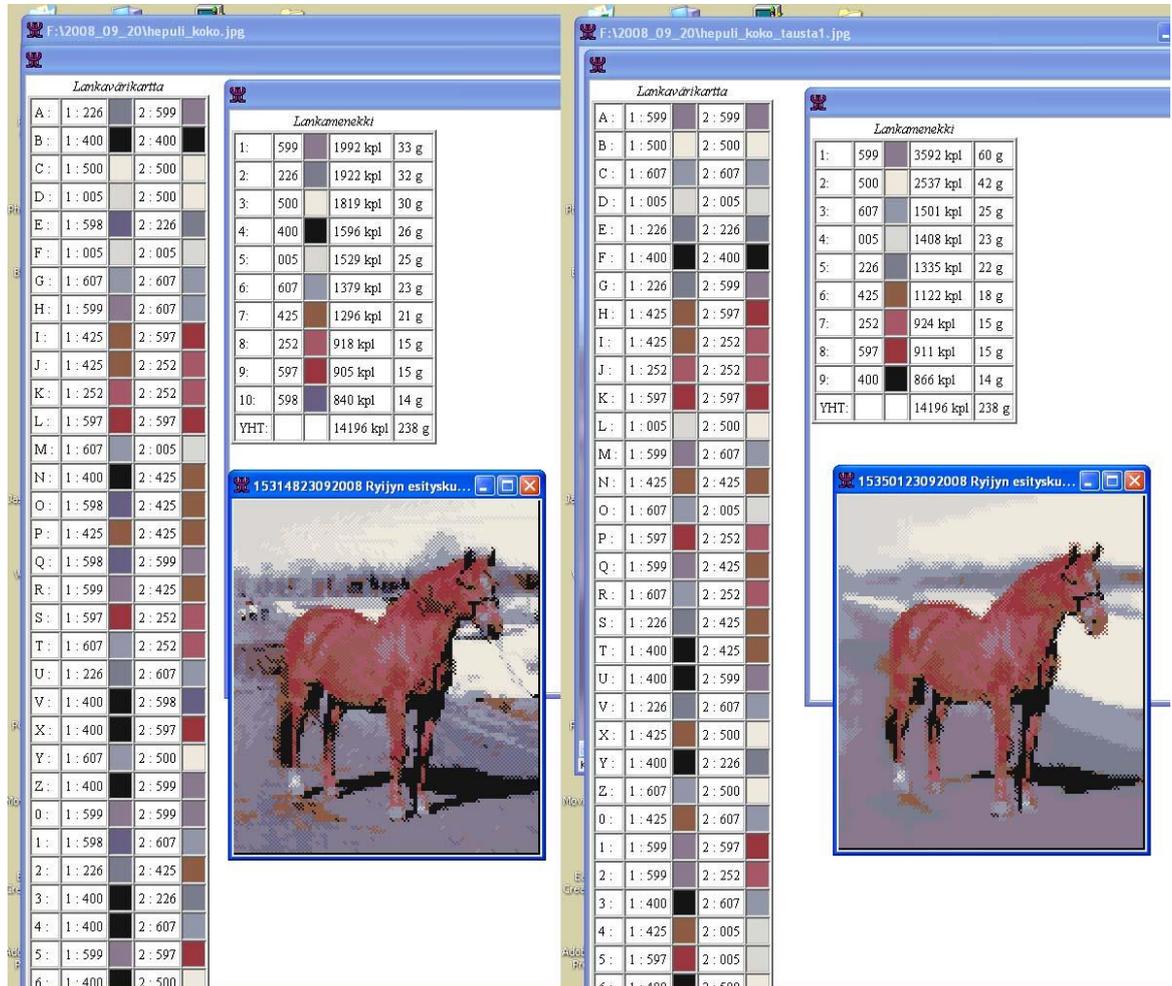


Figure 16. Testing different backgrounds for horse. Left has irrelevant details and right is without them.

Two different test images are shown in Figure 16. Difference can be seen on details that display image contains. More details on background makes work chart more complex and harder to follow when producing ready-made ryijy. Final picture selected for design is shown in Figure 19 compared to Ryijytin made display image. In Figure 18 is partial work chart created with Ryijytin software from final edited picture.

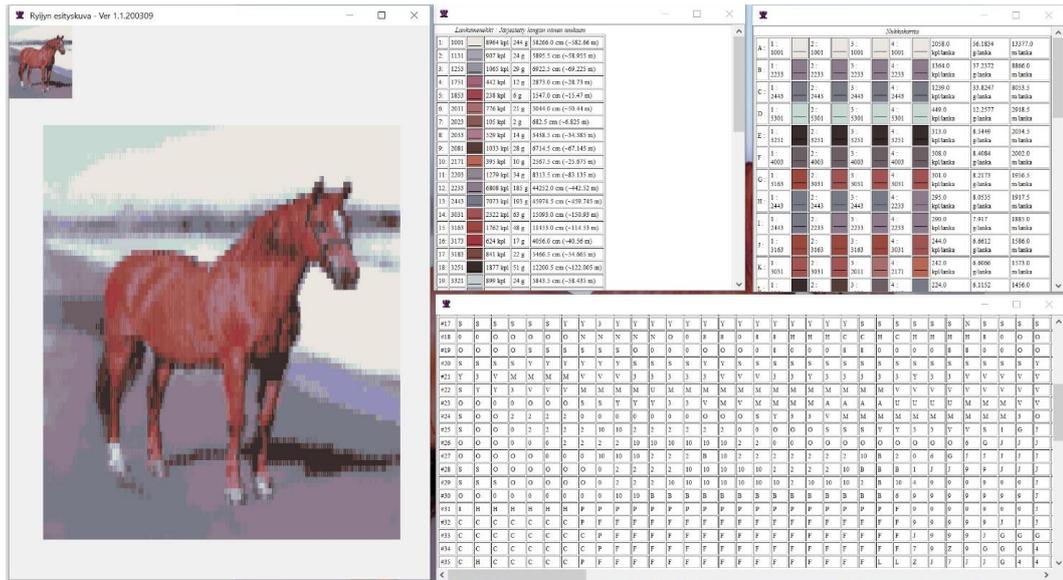


Figure 17. Sample Ryijytin software produced display image of Hevonen ryijy and part of work charts for ryijy.

Work charts for Hevonen ryijy was printed out for customer and as Ryijytin software 1.0.0.5 was incapable of producing full work charts only HTML files were saved with display image to hard disk.

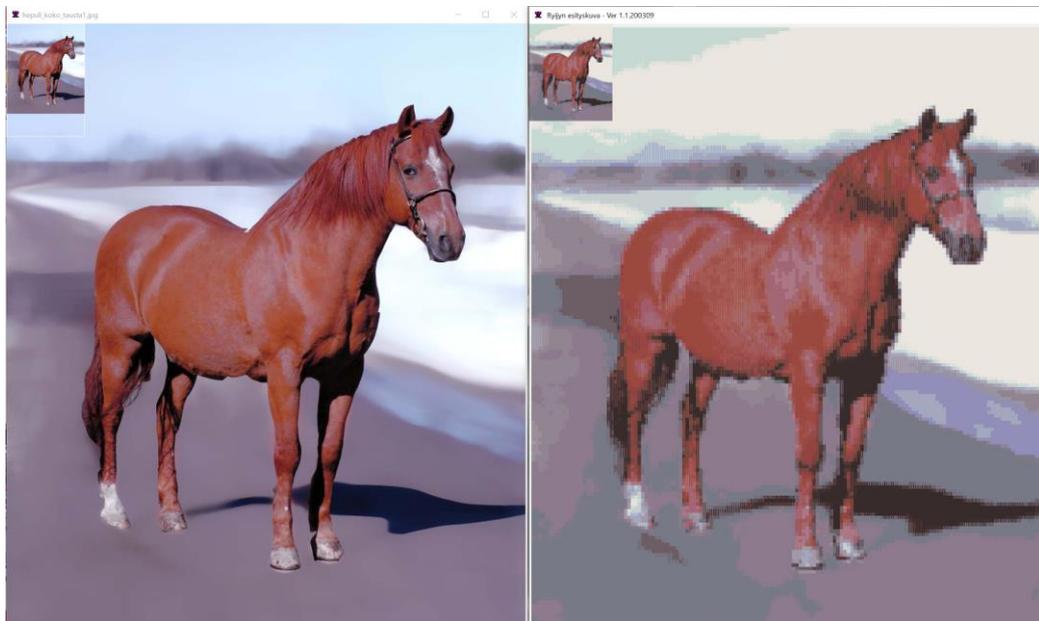


Figure 18. Original picture compared to display image of ryijy created by Ryijytin software.

When compared original image and display image of ryijy to ready-made ryijy in Figure 18 resemblance is quite remarkable. Some small details are of course lost and some colour areas doesn't blend well. Taking into account that original picture has millions of colours

and display image produced by Ryijytin software only dozens of. Hand sewed ready-made ryijy by Silmu peltonen is presented in Figure 19. Picture was taken with a compact camera and with flash. Image is edited with Adobe Photoshop as flash severed colours and final appearance depends on lightning and how well ryijy has been brushed.



Figure 19. Horse ryijy. Woven by Silmu Peltonen by hand to ready-made basecloth. Picture Marko Vähä-Erkkilä.

Selected dimensions for ready-made ryijy in Figure 19 were width 60 cm and height 70 cm. By type of base cloth piles per cm at width 2.38 and height 1.13. Amount of piles per width is 79 pieces and in height 143 pieces.

Second part of the work was done using picture of horse head. On Figure 20 at upper part is shown image used in designing ryijy and partial work chart produced by Ryijytin software. Lower part of Figure 20 represents display image produced by Ryijytin software and ready-made hand woven ryijy by Silmu Peltonen.

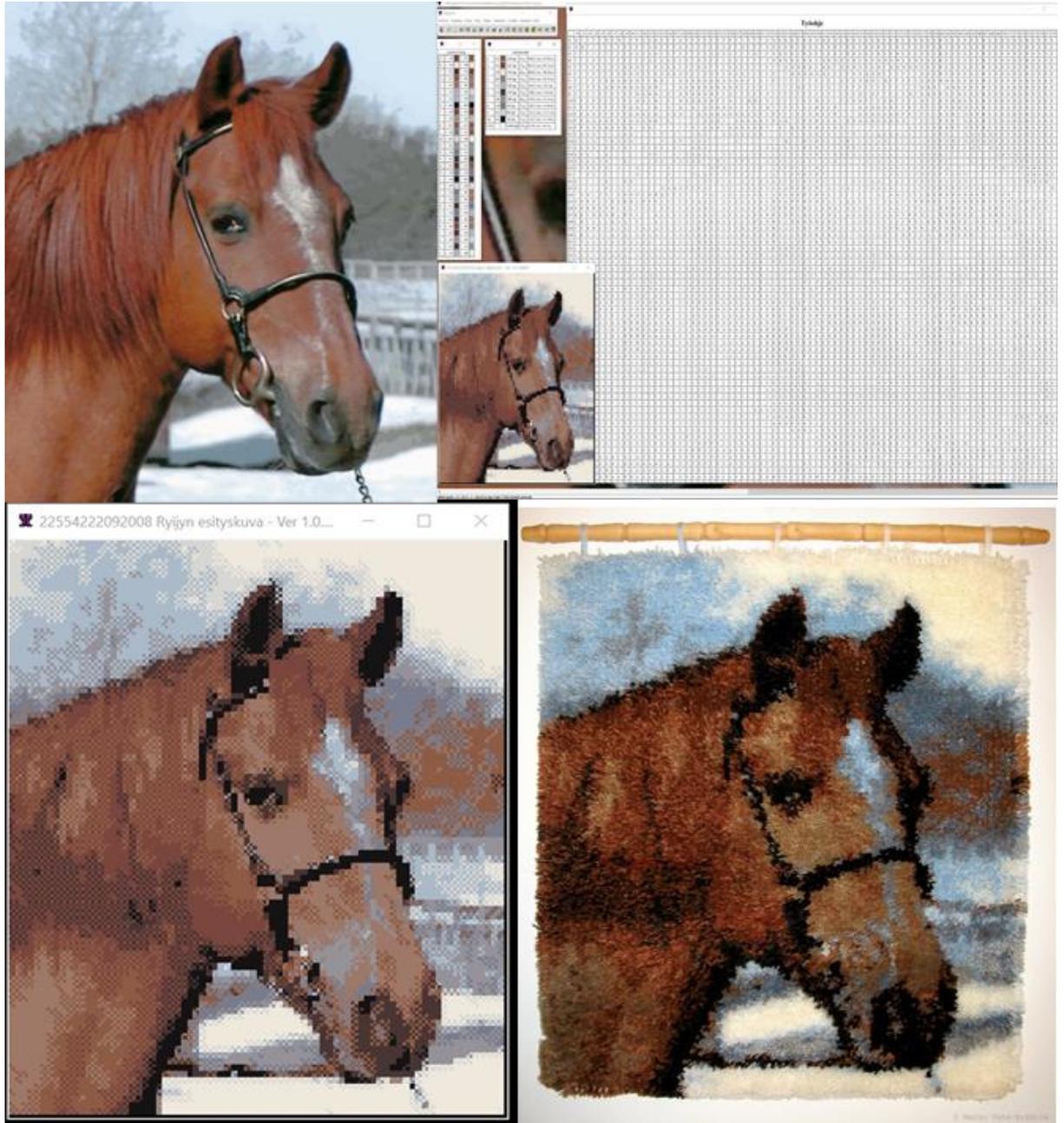


Figure 20 Upper left is image used in design and on the upper right is partial work chart produced by Ryijytin software. On the lower left is display image produced by Ryijytin software and on the lower right picture of final hand woven ryijy.

As can be seen on Figure 20 details from original picture are migrated to display image and final woven ryijy accurately. In this case display image does not fully show the richness of details on final product. As of display image is produced with only one shade of yarn representing of fabric that has more shades caused by light and angle of yarn on ryijy. With good brushing is able to change how the ryijy appears as yarns change places and the end of cut yarn is brighter. Tests showed that Ryijytin software has capabilities to become reliable design software.

### **“Karhu” ryijy (23.10.2013)**

For testing purpose was made ryijy based on well-known logo of Finnish beer brand. I don't have rights for picture and work was done for non-commercial educational purposes. As disclaimer the full work chart developed with Ryijytin software is not publicly available and is not for sale.



Figure 21. Original logo used for design of ryijy.

Original 24 bit jpg image in Figure 21 used for designing a ryijy work chart was downloaded from the internet. Downloaded image had dimensions of 2046 x 2046 pixels and it has a density of 96 dots per inch. Work was done with Ryijytin version 1.0.2.9 (build 24.9.2013). Grid was added to image in Figure 22 that has 106 squares in width and 106 squares in height. Which are amounts of knots in ready-made ryijy. From image with grid could be seen which colour areas are transferred to which knots in work chart of ryijy.



Figure 22. Image of “Karhu” with grid added to image.

One page of work chart in Figure 23 represents page 2 from a total of 18 pages work chart. Page marked as B is area covering bears left ear. Areas of similar knots are bordered with red line. In Figure 24 is shown map for pages. In page marked as A is table with knot

referred as #1/#1 that is upper left corner knot in ready-made ryjy. This is where the producing of ryjy will be started. Ryjy is made row by row from left to right.

B	#20	#21	#22	#23	#24	#25	#26	#27	#28	#29	#30	#31	#32	#33	#34	#35	#36	#37	#38
#106	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
#105	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
#104	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
#103	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
#102	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
#101	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
#100	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
#99	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
#98	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
#97	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
#96	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
#95	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
#94	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
#93	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
#92	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
#91	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
#90	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
#89	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
#88	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
#87	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
#86	A	A	A	A	A	A	A	A	A	A	B	B	B	B	B	A	A	A	A
#85	A	A	A	A	A	A	A	A	A	B	B	B	B	B	B	B	B	A	A
#84	A	A	A	A	A	A	A	B	B	B	A	A	A	A	A	A	B	B	A
#83	A	A	A	A	A	A	B	B	A	A	A	B	B	B	B	A	A	B	B
#82	A	A	A	A	A	B	B	A	A	B	B	B	B	A	A	B	A	A	A
#81	A	A	A	A	A	B	A	A	A	B	B	A	A	A	B	A	B	A	A
#80	A	A	A	A	B	B	A	A	B	A	A	B	A	B	A	A	B	B	A
#79	A	A	A	A	B	B	A	A	B	B	A	A	A	A	B	A	A	B	A
#78	A	A	A	A	B	A	A	B	A	A	A	B	A	A	A	A	A	A	A
#77	A	A	A	B	B	A	A	B	B	A	A	A	A	A	A	A	A	A	B
#76	A	A	A	B	B	A	A	A	A	A	B	A	A	A	A	A	A	B	B
#75	A	A	A	B	B	A	A	B	A	A	A	A	A	A	A	A	B	B	B
#74	A	A	A	B	B	A	A	A	B	A	A	A	A	A	A	B	B	B	B
#73	A	A	A	B	B	A	A	A	A	A	A	A	A	A	B	B	B	B	B
#72	A	A	A	A	B	A	A	B	A	A	A	A	A	B	B	B	B	B	B
#71	A	A	A	A	B	B	A	A	B	A	A	A	B	A	A	B	B	B	B
#70	A	A	A	A	A	B	A	A	A	A	A	B	B	A	A	B	A	A	A
#69	A	A	A	A	A	B	B	A	A	A	B	B	A	A	B	A	A	A	A
#68	A	A	A	A	A	B	B	A	A	A	B	A	A	B	A	A	A	A	A
#67	A	A	A	A	A	A	B	A	A	B	B	A	B	A	A	B	A	A	A
#66	A	A	A	A	A	B	B	A	A	B	A	A	A	B	A	B	A	A	A
#65	A	A	A	A	A	B	B	A	B	A	B	A	B	A	B	B	A	A	B
#64	A	A	A	A	A	B	A	A	B	A	A	B	A	B	A	A	A	A	A

Figure 23. Partial work chart. Page 2 marked as 'B' from 18 pages total.

On Figure 24 is knot colour chart where is presented yarn colours for each knot. In the same chart is given total amount of each colour used in whole ryjy, weight of that yarn and length before cutting it for knots. Continued in Figure 24 is also yarn consumption sorted by name of yarn and sorted by total amount. That is more useful if ryjy has lots of different colours.

Page map for work chart. Notice: Knot #1#1 of Page A is on upper left corner of ryjy.

A	B	C	D	E	F
G	H	I	J	K	L
M	N	O	P	Q	R

Knot color chart

A	1 : : 3251	2 : : 3251	3 : : 3251	4 : : 3251	9757.0 kpl/lanka	266.3661 g/lanka	63420.5 m/lanka
B	1 : : 6421	2 : : 6421	3 : : 6421	4 : : 6421	2015.0 kpl/lanka	55.0095 g/lanka	13097.5 m/lanka
C	1 : : 2261	2 : : 2261	3 : : 2261	4 : : 2261	32.0 kpl/lanka	0.8736 g/lanka	208.0 m/lanka

Yarn consumption sorted by name of yarn.

1:	2261	128 kpl	3 g	832.0 cm (~8.32 m)
2:	3251	39028 kpl	1065 g	253682.0 cm (~2536.82 m)
3:	6421	5788 kpl	158 g	37622.0 cm (~376.22 m)

Yarn consumption sorted descending by total amount.

1:	3251	39028 kpl	1065 g	253682.0 cm (~2536.82 m)
2:	6421	5788 kpl	158 g	37622.0 cm (~376.22 m)
3:	2261	128 kpl	3 g	832.0 cm (~8.32 m)
YHT:		44944.0 kpl	1226 g	292136 cm (~2921 m)

Figure 24. Page map, knot colour chart and consumption combined into PDF report.

Last part saved to the PDF file is projects save folder and image used for draft. Project save file for “Karhu” ryjy was E:\ryjyitin\_2013\_1.02.9\tallennus\KARHU\_VER7.rpt and draft image used was C:\Users\Markon\Documents\Karhu ryjy\Karhu\_VER3.jpg. Into PDF file is also inserted start and end time of calculating work chart. Calculating “Karhu” ryjy started 01:43:07 25.10.2013 and calculating first phase of ryjy was ready at 01:43:08 25.10.2013, work chart had 3 colours 7 different knots after first phase. Calculating second phase was finished at 01:43:08 25.10.2013. Final ryjy has 3 different colours and 4 different knots. Dimensions for ready-made ryjy is width 80 cm and height 80 cm. Knots per width is 106 pieces in row and in height 106 pieces on columns. Knots are tied 1.33 knots per cm in width and 1.33 knots per cm at height. Yarns per knot is 4 pieces each. Yarn length as cut is 6.5 cm. Colour chart used for design was E:\ryjyitin\_2013\_1.02.9\varikartat\karhu. Total amount of yarns in whole ryjy is 44944 pieces. Yarn consumption for complete is 1226.9712000000002 grams that is approximately 1.2 kilograms and 292136.0 centimeters that is approximately 2.92 kilometer.

In Figure 25 is shown whole work chart as one piece with colours and letters in the same picture. Picture is saved as reference and to ease to understand final work chart. When saving work chart to PDF there are several options for saving one page version of work chart. Different options were explained in chapter 4.3 at subsection Printing Work Charts and Saving to PDF with Figure 12 and in chapter 4.3 at subsection Editing Work Chart in Figure 14.



Figure 25. Work chart with colours behind. Detail of right ear zoomed in picture.

As of outcome ryijy shown in Figure 26 was woven to base cloth by sewing knots by hand. Bit of details were left out when sewing for example the line from bottom of figure. The outcome follows the original picture seamlessly.



Figure 26. Ready-made ryijy hanging on the wall.

### “Malta -06” ryijy (2014)

Ryijytin software was leased to third party for testing between years 2013 and 2014. In which time development was done accordingly to feedback. During that period design student of Metropolia University Katri Koivisto was experimenting with Ryijytin software. Experimenting was done to test Ryijytin software -07 abilities to design ryijy based on digital photograph. Several digital images were tested and finally one selected to be used in design of ready-made ryijy. Picture was selected based on interviewing target group for interest of different kind of ryijy designs. Target group was specified to be person on age group of 20 to 35 years old. Picture to be transferred to ryijy with Ryijytin software was selected to be something that is not typically seen in ryijy. Guideline was not to let history of ryijy or limitations of thinking what ryijy design could be to specify the design. The image quality was intentionally selected to be mediocre as such pictures were estimated to be the largest part of customer’s photographs. That does not mean weak photography skills, but cheap or average digital pocket cameras that cannot produce excellent image quality. Meaning below average image quality of cheapest compact cameras on the market. Also by interviewing group of 20 – 25 year old women was clear that old ryijy designs were considered to be old-fashioned, outdated and ugly. The image of an interesting ryijy should be self-selected and self-specific. Two out from three would select image from their own holiday, honeymoon, memorable place that is recognizable by them or person of love etc. And the third would select more abstract theme, picture that is zoomed beyond recognition of subject or colours mean more than subject of picture. Most important thing for everyone is that ryijy is personal and unique. Final picture selected for ryijy design was from memorable moment on holiday and it had meaning for writer of bachelor’s thesis. (Koivisto 2014)

Size of ryijy was considered based on place of ryijy in modern home. Apartments nowadays are smaller than in the past and very rarely there is room for big ryijy e.g. 145 X 180 cm. Size of ryijy should although be big enough to use more detailed photo. If the ryijy is too small, it would also feel detached from the interior. Suitable place for ryijy could be over the bed or at the back of the couch. Size and shape of ryijy could be in the same range as horizontal interior tables. The bottom line is that ryijy should link to whole interior. (Koivisto 2014)

Image selected for ryijy design is shown in Figure 27 along with work charts and display image produced by Ryijytin software. First test did not satisfy Koivisto (2014) as exposure of selected picture was wrong. Resulted display image of ryijy was flat and colours did not repeat as intended. Adobe Photoshop was therefore used to correct contrast and colours using automatic tools. The width of finished ryijy was selected to be 110 cm and height 76 cm. Number of different colours in rug is 30 and amount of different knots is 35. (Koivisto 2014)

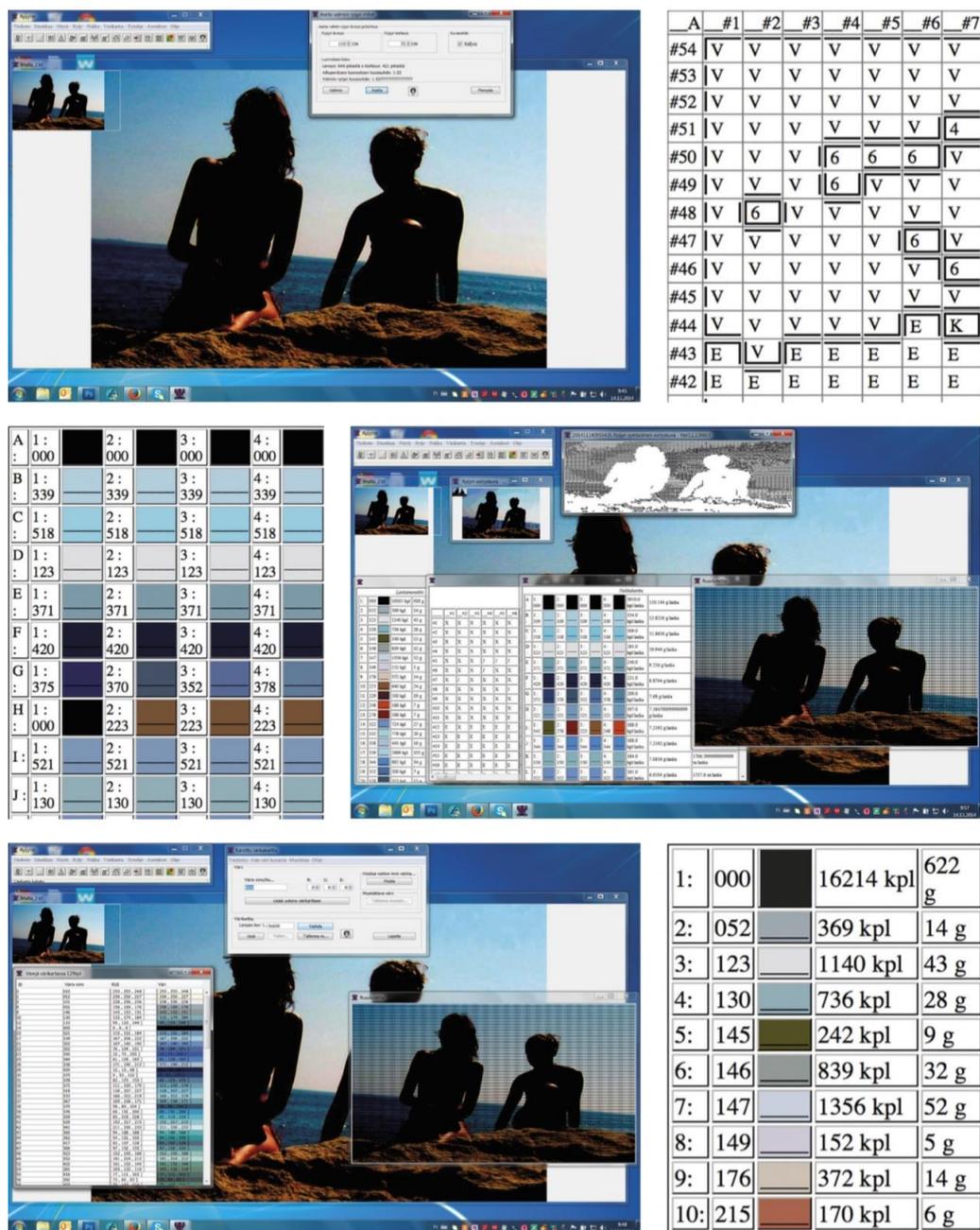


Figure 27. "Malta -06" ryijy design in Ryijytin 1.0.3.5 software. Screen captures Katri Koivisto (Koivisto 2014).

In Figure 28 is ready-made “Malta -06” ryijy. Ryijy was made by hand sewing to base cloth. Decision was made based on fact that nowadays is only in rare occasion to have looms at home. Materials used to produce ryijy was 1404 grams 100% sheep wool yarn and 76 cm of base cloth that comes with width of 110 cm in a roll. Time used to produce ready-made ryijy was 109.75 hours in total. (Koivisto 2014)



Figure 28. Ready-made “Malta -06” ryijy. Picture Katri Koivisto (Koivisto 2014).

Ready-made ryijy came out mainly as wished. Shades of sky would have come out better. On the other hand the outcome looks good and colour areas in the sky distract if the ryijy is considered as independent work and not compare to picture. Posture of figures is represented beautifully in ryijy. (Koivisto 2014)

#### 4.5 Conclusion of Tests

Ryijytin version 1.0.3.6 is easy to use and design process after feeding yarn colours into Ryijytin software is fast. Ryijytin software is easy to use and even photographs with lots of details can be used to design a ryijy. Some flaws and errors could be found from

Ryijytin software but nothing dramatic that prevents designing a ryijy or affects usability. (Koivisto 2014)

In every software development process is crucial to use outsider when considering software usability. Koivisto (2014) took a good look into Ryijytin software and feedback she gave was highly usable and on point. Most flaws I were able to fix instantly as there were some flaws in representing work chart etc. As a designer of the software I knew how it was supposed to be used. From the view of real user there could always be something that the software designer does not see coming. Therefore I was exhilarated that Koivisto (2014) saw Ryijytin software easy to use as it proved that I had successfully designed graphical user interface into purpose of designing of ryijy. And notices that Koivisto (2014) gave about software were enlightening to me.

Decade of incremental development of Ryijytin software -07 provided enough valuable information about guidelines for developing complete new version of Ryijytin software. New version of Ryijytin software is referred as version 2.0 or Ryijytin software -20 to distinguish from current Ryijytin software -07. Information gathered from user experience and interviews by Koivisto (2014) gave me valuable information for my continuing development of Ryijytin software -07. Ryijytin software -07 version 1.0.3.6 architecture is shown in Figure 29. Saving and editing yarn colour charts was added since Ryijytin software -07 version 1.0.0.0 (build 9.4.2007). Writing and reading from project files was added since Ryijytin software -07 version 1.0.0.2 (build 28.4.2007). Starting from Ryijytin software -07 version 1.0.2.9 save as PDF (version 2) was included. Ryijytin 1.0.3.0 (build 4.11.2013) introduced edit work chart windows. In Figure 29 connections are not visible but edit work charts windows inputs calculated yarn colours data and work chart data. Edit work chart windows comes to replace output browser windows in upcoming Ryijytin software version 2.0. And instead of writing and reading HTML files editing work chart module uses data module as input method. In next chapter I will discuss why next generation of Ryijytin software is needed instead of modifying current software.

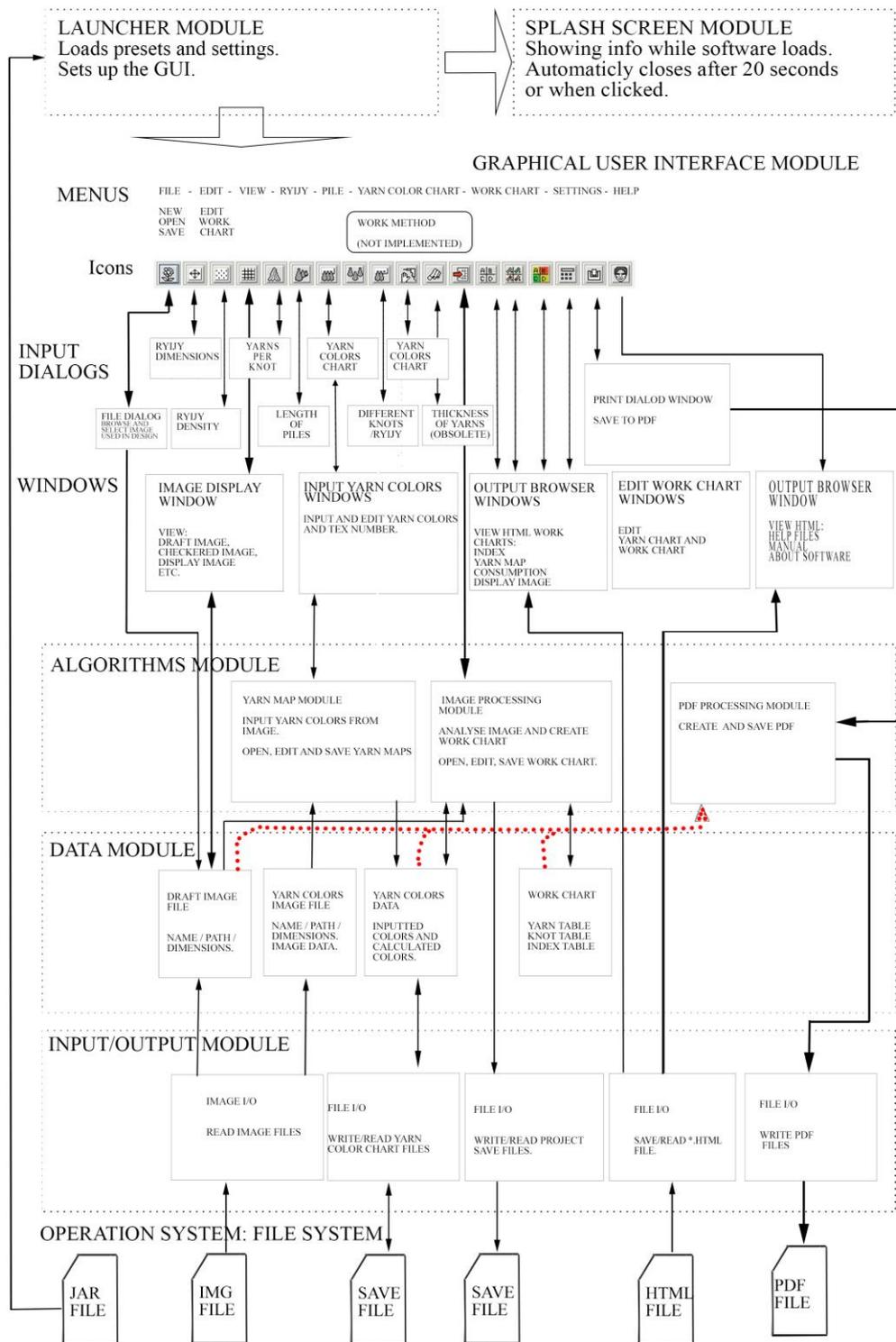


Figure 29. Ryjytin software Version 1.0.3.6 architecture.

#### 4.6 End of Lifetime of Ryijytin 2007

Improvements have been done since the first version, but with over ten years of lifetime Ryijytin 2007 has become both impossible to manage and uses obsolete or deprecated Java libraries. As well Ryijytin version 2007 descendants does not implement all required functionality for business that are impossible to add to current software. First design had some flaws because of misunderstanding basics of ryijy design at first place and furthermore it did not take into account all aspects of needed parameters. With incremental builds were possible to add most features to software but changing one part could lead some other parts to work not as intended.

To get Ryijytin version 1.0.3.6C (build 16.9.2018) build with NetBeans IDE 8.2 (Build 201609300101) that was updated to version NetBeans 8.2 Patch 2 and JDK 1.8 update 192 running on JRE version 8 update 192 b12 on Windows version 10.0.17763.437 I had to include on NetBeans IDE to libraries and copy 42.2 Mb file named rt.jar from my computers Java/jre6/lib folder to my Ryijytin/lib folder. Before doing that trying to run Ryijytin version 1.0.3.6C (build 16.9.2018) gave error message “Exception in thread "main" java.lang.NoClassDefFoundError: org/jdesktop/layout/GroupLayout\$Group”. Message pointed that Swing classes were missing from JRE. Other possible solution would have been building Ryijytin with JDK 1.6 update 18 but that would mean stepping backwards and excluding available improvements that were added to Java in further versions. That would mean rewriting some parts of software that uses Java 7 and 8.

As I continued searching for reason what causes the problem turned out that some classes were deprecated after Java 6. Therefore org.jdesktop.layout was any longer present and hence not available within JRE for use trying running software causes error. In newer JRE javax.swing.GroupLayout replaced org.jdesktop.layout. As it turned out by testing with NetBeans IDE defining Grouplayout class resides in guarded block generated by NetBeans IDE and therefore changing it would need forking within files. Inside NetBeans IDE fixing problem seemed impossible as code was protected against changes. There could have been workaround but I could not use any more time for seeking it. As conclusion developing Ryijytin 2007 version 1 software was ceased and designing Ryijytin version 2 begun.

## 5 Ryijytin Software Version 2.0

From the start of Ryijytin software -07 development it was clear that in some point developing comes to point where there is not possible to continue. Therefore intention for developing Ryijytin software version 2.0 in future was already done in the beginning. Purpose for developing Ryijytin software version 1.0 was to test different techniques and collect information about designing of ryijy. Final decision for developing Ryijytin software version 2 was done during year 2014. Next software version was planned to be available in the beginning of 2015. Due to technical difficulties with my PC and upcoming news about future of Java delayed development. The first difficulties were caused by my computer that started malfunctioning. After a period of installing and trying different drivers it came clear that I had some fault parts on my computer. Fixing it completely took almost two and a half years as the problem was complicated and hard to reproduce. Replacing part by a time and testing took place and sometimes looked like I got it sorted out but new problems emerged. Finally I had changed most of parts and installed all my software several times. As problem sorted out I had replaced most parts under warranty to new ones and lost one hard disc full of data. As always backing up constantly saved my data. In the end all I had lost was precious time and arrangement of files. Restoring files from backup seemed to take eons and sorting all data was massive project.

From prototyping and experimenting with different techniques were collected enough information for developing next completely new version of Ryijytin software. The need for new version was imminent because Ryijytin software -07 had become both impossible to manage and Ryijytin software -07 uses obsolete or deprecated Java libraries. Long period of developing software have collected garbage to project. Garbage that are removed lines of code and unused files left on project path that had been replaced with different files etc. After so many years of improving, fixing and developing it is not easy to follow what has been done and what was the original purpose of some test files included in project. Therefore in 2014 I started designing new user interface for Ryijytin software 2.0, as in same time as I did some optimizing and testing with Ryijytin version 1.0.3.6. Ryijytin software version 1.0.3.6 was derived to subversions A, B and C in which were updated some libraries to newer version and which were mainly maintenance update versions. When designing next version I benefitted from material and user feedback that

had been gathered since year 2007. Version 2.0 development was supposed to start earlier and was planned to be released at year 2015 and later set to year 2017. Before I could proceed with next version I had to wait to NetBeans IDE to switch from Oracle to Apache and mature until it became usable. In same time Java 8 was released (in March 2014) and soon rumors started to circulate about Oracle changing its policy in use for commercial projects as need to pay license fees. Java 7 support was going to end during year 2015. That all set uncertainty to my project. As well as until starting of 2010 decade Java runtimes were preinstalled to every computer. After some severe security vulnerabilities affecting Java runtime releases Java runtimes were left out from most operating systems browser. Availability of HTML 5 browsers made Java obsolete in most web applications and thus Java runtimes were any longer pre installed in different operating systems.

In chapter 5.1 is discussed requirements for software. Requirements are still in broad outlines as in the beginning of the project. Some features are dismissed and some new introduced during developing previous version of software. As the project matured and requirements were clarified within several use cases explained in chapter 4.4 and suggestions were collected from which new ideas were gathered. Development tools are discussed as selected Java programming language and commercial nature of projects sets the range of tools that could be used for development. To be able to use Java commercially without paying fees to Oracle software developer can choose OpenJDK that is open source version of Java development tool kit. Apache NetBeans IDE is free development environment for Java and it is continuously developed. Continued with discussion of language and settings requirements for software and how it will be implemented. Best practice for developing software is to be able to easily change the language used to text and labels inside software without rewriting code every time language changes.

Chapter 5.2 sets guidelines for creating graphical user interface for Ryijytin software version 2.0. As how it should be designed to be functional and easy to use for all. Software should be designed to be used without knowledge of computing or special skills. Graphical user interface should always be intuitive to use and as instructional as possible for beginner users.

## 5.1 Version 2.0 Requirements

Requirements for version 2.0 are as functionality of software are almost the same as for first explained in my bachelor's thesis and also at section 2.2 and section 4.1. Some new ideas and requests were added to the original plan according to functionality of software. Main function of Ryijytin software would be to assist in the design of Finnish ryijy. Considering the target group for the software, the goal will be to make designing ryijy easier and faster. For that purpose software that aid on designing of ryijy would be created. Users of software should consider to be beginners or without any previous knowledge of designing ryijy. As in her thesis Koivisto (2014) mentioned history or traditional methods should not limit the design of ryijy with Ryijytin software.

### Development tools

Tools for software development were chosen according to needs. Because I have had no funding for my project I have preferred open source tools with limitations that use should not be required to open source code. And even better if tools are free to use commercially as long as with software are linked where to find source for libraries used. Software piracy was not an option because of commercial nature of my project. I could have used the tools provided for educational purposes for my thesis but not in my further planned of selling software.

### OpenJDK

From starting of beginning of Year 2019 Oracle started collecting license fees for using commercially JDK owned by Oracle. In future could be possible that even using JDK by Oracle in commercial project would cost something and stop working in the worst case scenario. As my project of Ryijytin software could be considered as commercial and I didn't want to buy any license as revenue for the past several years has been zero. Therefore I studied other possibilities. Resolution for that problem is to use OpenJDK that is open source version of the same JDK but without fees and does not have optimizations, examples etc. added by Oracle. Current version by the time of writing is OpenJDK 12.0.1. (JDK 12.0.1 General-Availability Release)

## **Apache NetBeans (incubating)**

Oracle announced in 2016 contributing the NetBeans IDE as an open-source project (Oracle Proposes NetBeans as Apache Incubator Project). The NetBeans was by the time of writing in transition to Apache and it was not fully implemented until December 2018. NetBeans IDE 9 was not able to open my previous NetBeans projects and needed some tweaking to get it to run on my PC. Version I was trying was Apache NetBeans IDE 9.0 with JDK 11.0.1. Apache NetBeans (incubating) 10.0 announced on the 27th of December, 2018 (Wielenga, G. 2018b). Version 10 seemed to be working perfectly when I tried it. Newest version on time of writing is Apache NetBeans (incubating) 11.0 that was released on April the 4th, 2019 (Wielenga, G. 2019).

## **Language and Settings**

Java has excellent built in tools for internalization as described in chapter 3.1. Best practice is to start software development with language files that are text files containing all available translation strings for software. As changing languages in middle would be challenging or impossible. All tags, phrases, error messages and labels should be written into code as references to selected language. Software should never rely only on computers locale setting as user's understandable language could be different. Software should never assume anything and let the user themselves decide what they want.

Software always have some average default settings written into but everything should be interchangeable and able to save as the new default. Therefore settings should be read from file and write to file as well. When installing the software to computer installation program ask preferred language. Preferred language is then written into setting file. Every time software loads checks it first for setting file and set language and settings accordingly. Language files should be made with Java resource bundle so translating to new languages is easy. For other settings I will use XML files because they are not dependent on any software version.

## 5.2 Software Design

Software design begun from graphical user interface. Proposed software architecture is presented in Figure 30. As software loads the launcher module first sets up splash screen to show info about Ryijytin software, its version and owner and so on. While showing splash screen launcher tries to load setting file. If setting file could not be found on system for any reason, launcher sets up error message asking to reinstall software and exits to the operating system. If everything is in order launcher sets up GUI and sets it visible. Splash screen closes after preset time or when clicked. For Ryijytin software version 2.0 I designed selection bar that preferably have five different buttons as in Figure 30. For the rest as much as possible will be used from an old project. Parts of code has to be rewritten but most algorithms are transferable to the new project.

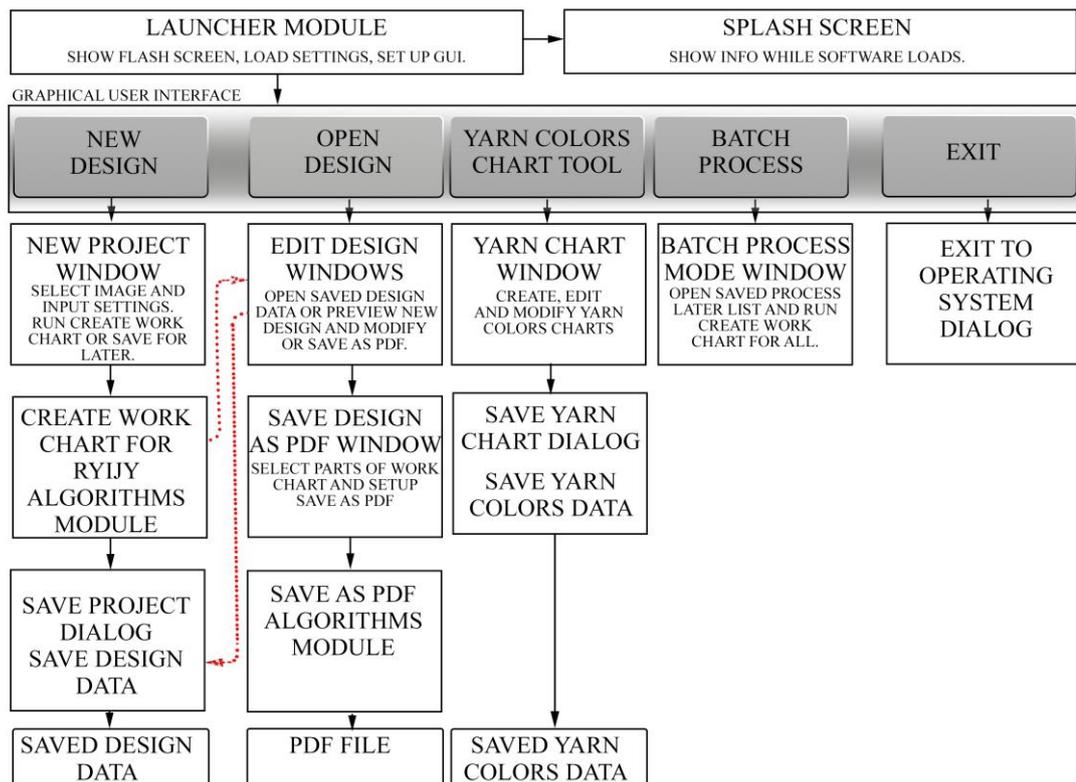


Figure 30. Ryijytin software version 2.0 architecture.

As the first version of Ryijytin software was little bit confusing with all opening and closing settings dialogs, second version is designed to be simpler and straighter forward.

By clicking “New Design” button user starts new project from scratch and setting window in Figure 31 opens. The New design window has a collection of all settings needed for designing a ryjy. In the setting window is missing setting for brightness and contrast that would be included in production version of Ryijytin software -20. Most photographs needs some tuning in their exposure and contrast before used for designing a ryjy. Automated tuning based on image histogram for them is set to be included. Also user can set the tone and warmth of image before processing. For batch mode preferably automated function to automatically set contrast and brightness of image to optimal for Ryijytin software -20. Best result is acquired with an image that is bright, colorful and has contrast.

Design Preview [NewDesignWindow]

File name for design:

Write description of design. Used in work chart about section:

Image used for design:

Image preview:

Settings for Ryjy:

Dimension:  
 Height:  CM X Width:  CM

Density (knots per CM at):  
 Height:  /CM X Width:  /CM

Yarns per knot:  
 pieces

Length of yarn as cut for knots:  
 CM

Limit colors of ryjy to:  
 different colors in ryjy (Zero = No limits).

Limit different knots of ryjy to:  
 pieces of different knots in ryjy (Zero = No limits).

Minimum Grams of yarn in Ryjy.  
 grams (Zero = No limits).

Select method how knots are in ryjy:  
 Knots below another  Knots overlapping each other.

Figure 31. Ryijytin software version 2.0 New Design window.

After user has done setting up properties and is ready to calculate work chart for ryijy that is done by clicking “Calculate and save design” button. Before continuing any further Ryijytin software version 2.0 shows window that collects all data together, does some calculations if numbers are realistic and informs user of ready-made ryijy parameters. In case there is still something to be changed user can go back to make needed changes.

Also third option that first prototype does not have is to add for batch process for later mass calculate work charts. In case calculate now is selected Ryijytin software then goes through calculating work chart for ryijy. After calculating is done and design has been saved automatically Ryijytin software goes to edit design phase. Same can be done by clicking “Open Design” button that opens first dialog to select saved design file. Saved design then opens into edit view based windows shown in Figure 13. Editing functions as described in chapter 4.3 Editing Work Chart and further I will add some new features as more drawing tools. When user is satisfied with the design, it could be saved as PDF. Save as PDF will open window based on Figure 12. Saving as PDF will work as described in chapter 4.3 Printing Work Charts and Saving as PDF. In Ryijytin software version 2.0 PDF will be created using Apache FOP.

Clicking “Yarn Colours Chart Tool” button will open window based on Figure 10. Creating new yarn colour chart or modifying will work as described in chapter 4.3 Saving and Editing Yarn Colour Charts. Yarn colours are read from image of actual yarns.

Batch process is a new feature which allows to process several designs in row using previously saved settings. Feature was added into plan to be able to take orders via the internet. Feature is described in chapter 6.3. Customer inputs preferred properties for ryijy and uploads photograph to server via web page. Input form then saves properties into XML file for later processing. Data from XML and images are then combined and processed one by one until all have been gone through. Ryijytin software version 2.0 creates save file and PDF for each design.

## 6 Discussion and Conclusions

Ryijytin software version 1.0 (BEng version build 10.3.2007) managed to prove that the concept of computer aided design of ryijy was possible to realize. Managing to develop software as an independent entrepreneur and without the support of any company was challenging. Meanwhile I worked as an employee in a retail store and as a part-time janitor. Developing my own software had to be done at spare time. A lack of financial support for the project caused that I could not afford to have any extra vacations. And harsh times at work periodically made it impossible to continue working with Ryijytin software -07. The timeline for developing Ryijytin software version -07 can be seen in Appendix 3. Empty spaces in the development timeline of new Ryijytin software version does not necessarily mean that I did nothing. I was researching and studying different periods of time and got new versions out in bursts. Most of the software development was done between years 2007 to 2009 and 2012 to 2014. Between 2009 and 2012 I did less development and more research with other projects. After 2014 I started collecting data for Ryijytin software version 2.0. As working alone for myself computer troubles took my time between 2015 and 2017, as I could not afford to buy a completely new computer to replace the malfunctioning one. Inspecting the reason took all my spare time and eventually my computer seemed to be working perfectly for a while. A first suspect was a bigger Windows update but eventually it started to unfold so that I had several faulty components. Having my computer back to a working state but had lost one hard disc I finally was at a point where I could continue my work. Digging all project files from several backups although took long time as everything had to be sorted again.

The future is newer predictable and forthcoming success of Ryijytin software -20 cannot be predicted for sure. Past decade showed challenges in selling the idea of a computer aided design for a ryijy as most elderly people carried worry about using a computer with crafts or generally resisted use of computers. Inventions as Google AI creating images on its own logic has although lowered the bar for a computerized art-like content. Nevertheless by shifting the aim from artistic to more industrial design I believe that Ryijytin software have possibilities to be successful. With the aid of Ryijytin software design may not be artistic but something that has not seen before as ryijy. Bringing new kind of motifs into ryijy and uniquely personalized content.

In this chapter I will discuss about marketing and service concept for Ryijytin software. Selling something completely new to a generation that does not know or are interested in past could be challenging. Most of the younger population sees ryijy something that is from the past and ancient. Something that were hanged in their grandparents houses. (Koivisto 2014) Future of Ryijytin software depends on as discussed in chapter 6.1 how well marketing of ryijy as modern interior decoration element manages to pull up to everyone's awareness. Chapter 6.2 discusses different marketing methods and delivering instructions to users. Chapter 6.3 discusses possible service concept built in Internet.

## **6.1 Marketing**

Marketing has changed dramatically in the past decade. More content have shifted to the Internet and readers of printed media or television and radio have decreased. Marketing to masses is arranged easiest by a well-done company web site and with the help of Google marketing or search engine optimization in general. The aftermarket has also gone to digital media. In most cases when one buys a new product one gets a leaflet with quick instructions and information where to download an updated operation manual or software with the most current instruction manual.

### **Social media**

Nowadays marketing has gone into social media. Platforms such as Facebook and Instagram are full of companies trying to sell themselves. Social media has become an affordable way to market to the masses. With a couple of dollars one can buy exposure. Facebook and Instagram have algorithms that calculates target groups that could have interested on your product. For a few dollars a company gets targeted marketing. Rest depends on how well companies have placed content to social media accounts.

### **Instruction material**

Printed manuals are past time. Nowadays when buying a new product one gets a leaflet or quick instructions guide including a printed a link and QR code to a website where one can download up to date manual and possible drivers etc. A typical way to help user is to

set up a YouTube channel where to upload instructions and how-to videos. Watching a video on how to use software is in most cases the easiest and most convenient way to understand basics. Even more complicated instructions are easier to show than to write in an understandable way.

## 6.2 Service concept for Internet

Koivisto (2014) designed a service concept for Ryijytin software in her thesis. The concept is designed to fulfill the needs of regular people that are interested in buying a kit for self-made ryijy or ready-made ryijy from their own photograph. Price range for such a kit would be starting from 250 euros and ready-made ryijy several thousand euros. Price includes a design made with Ryijytin software -07 and bundling materials with a work chart. As suggestion a small fee would have to be paid in advance to get preview of ryijy plan. After a customer accepts the ryijy design, a fee is reduced from the total price. A fee is used to avoid misuse of the service. (Koivisto, 2014)

Service design refers to the development, design and creation of services by design, where the service user is at the center of design. The designer's work is therefore no longer just a product-oriented design, but has expanded to the design of comprehensive experiences, different processes and services. Services has to be easy to use, desirable and appealing to use, and generate value to the customer. This can be achieved by acquiring understanding what customer needs and customer service experience are. The service experience is tailored for each customer individually and every time the customer experiences the service. The core of service design is to understand the values and the formation of the target group or customer. Value refers to the relationship between benefits and price, in which the price does not always mean more monetary value, but also the investments and sacrifices made by the customer before the acquisition. Therefore, in addition to the monetary value, the effort that the acquisition has required is also taken into account. So the value is the customer's experience that is in relation to the client's experience and what he / she appreciates. Value is relative and there is no absolute value. (Koivisto, 2014)

From the service concept based on bachelor's thesis by Koivisto (2014) that I saw possible to be implemented was to set up a web page selling work charts for ryijy. A customer would be able to select from few different widths of base-cloths, preferably from two sizes available on roll. Other needed ryijy properties are fed into a form and their own image is uploaded to a server. Saved ryijy properties and image would then be processed with Ryijytin software -20. A design made with Ryijytin software -20 would then be sent to customers email for approval. (Koivisto 2014) To make the process possible and easy I will have to collaborate with a manufacturer or a reseller of base-cloths and yarns for ryijy. For now that is not yet realized and no arrangements have been done. An another possibility would be to continue selling work charts made from customer's images. In this case a customer would buy materials separately. That I have been doing as a part-time entrepreneur under trade name Koodikas since 2007.

### **Selling designs for ryijy**

The Ryijytin is proprietary software owned by Marko Vähä-Erkkilä. Trade name Koodikas was acquired in May 2007 to be able to offer a ryijy design service. The ryijy design service is run as a sole trader. (Yrityksen tiedot 2015). Without a proper marketing there have been only a few customers until now. With a proper marketing and after I have developed Ryijytin software -20, it is possible to gain some more customers. A ryijy is something that one doesn't change often and ryijy could well be purchased once of a lifetime.

### **6.3 Future development and updating software**

Ryijytin software version -07 development has ended and there is no further updates coming. Developing Ryijytin software version 2.0 continues and release is planned to be available in 2020. Updating of Ryijytin version 2.0 will continue for least 5 years. Future will show if the development continues longer than that or will there be Ryijytin software version 3.0. In case project continues as a part-time work, I don't see any fast development. On the other hand, if Ryijytin software 2.0 successes to raise funds for development, nothing would stop me doing fulltime work with the development.

## References

### Articles, Journals:

Jalava, I. 1931. Kotiliesi, Omatekoinen ryijyni.

### Books:

Asplund, K and Kumpulainen, K. 1991. Kuhmoisten ryijy kautta aikojen.

Kuhmoinen:Kuhmoisten kunta

Cadenhead, R. 1999. Java 2 Trainer. Helsinki: IT Press

Cadenhead, R. 2012. Sams Teach Yourself Java™ in 24 Hours, Sixth Edition. United States of America: Sam's Publishing.

Koivisto, K. 2014. Valokuvasta Ryijyksi. Palvelukonseptin suunnittelu Suomen Käsiyön Ystävät Oy:lle. Bachelor's Thesis.

Louhio, A. 1970. Taideryijyjä – Modern finnish rugs. First Edition. Helsinki: Suomalaisen kirjallisuuden kirjapaino.

Lowagie, B. 2007. iText in Action: Creating And Manipulating PDF. First Edition. United States of America: Manning Publications Co.

McLaughlin, B. 2000. Java and XML, First Edition. Sebastopol, CA: O'Reilly.

Norton, P and Stanek, W. 1996. Peter Norton's Guide to Java Programming. United States of America: Sams.

Schade, D. 2015. Image Similarity Using Colour Histograms. Bachelor Thesis

Sun Microsystems, 1999. Programming in Java™ Advanced Imaging. United States of America: Sun Microsystems

Rosenstock, D. 2013. Finding Ground – Threads between here and there. Master of Arts Thesis. Assessed 28.11.2018. Available in [https://aaltodoc.aalto.fi/bitstream/handle/123456789/10364/optika\\_id\\_825\\_rosenstock\\_dorothea\\_2013.pdf](https://aaltodoc.aalto.fi/bitstream/handle/123456789/10364/optika_id_825_rosenstock_dorothea_2013.pdf)

Pylkkänen, R. 1967. Ryijyperinteitä 1500- ja 1600-luvulta. First Edition. Helsinki: Suomen Muinaismuistoyhdistys

Schildt, H. 2007. Java™: The Complete Reference, Seventh Edition. United States of America: The McGraw-Hill Companies.

Vuorisalo, M. 2007a. Ryijytin – Innovatiivinen Yhteistyö Suunnittelussa. Master of Arts Thesis.

Vähä-Erkkilä, M. 2007. Visuaalinen ryijynsuunnitteluohjelma – Ryijytin (Visual Rug Planning Software Ryijytin). Bachelor's Thesis.

### **Discussions:**

Vuorisalo, M. 2006 – 2013.

### **WWW:**

A Brief History of NetBeans. Accessed 11 December 2018. Retrieved from <https://netbeans.org/about/history.htm>

iText PDF: Your JAVA-PDF library - History. Accessed 13 December 2009. Retrieved from <http://itextpdf.com/history>

ITEXT 2.1.7 (JULY 7, 2009). Accessed 13 December 2009. Retrieved from <http://itextpdf.com/changelog/217>

JDK 12.0.1 General-Availability Release. Accessed 12.1.2019. Retrieved from <https://jdk.java.net/12/>

Kowal, G. 2017. Launch4j Cross-platform Java executable wrapper. Accessed 11 November 2018. Retrieved from <http://launch4j.sourceforge.net/>

Oracle, 2016. Oracle Proposes NetBeans As Apache Incubator Project. Accessed 18 November 2018. Retrieved from <https://netbeans.org/community/apache-incubator.html>

Oracle, 2019a. Java™ Internationalization Support. Accessed 18 November 2018. Retrieved from <https://docs.oracle.com/javase/8/docs/technotes/guides/intl/index.html>

Oracle, 2019b. Oracle Java SE Support Roadmap. Accessed 12.5.2019. Retrieved from <https://www.oracle.com/technetwork/java/java-se-support-roadmap.html>

Sinervo, P. 2016. Traditional methods of making ryijy. Accessed 12.11.2018. Retrieved from <https://punomo.fi/tekniikat/ryijyt/traditional-methods-of-making-ryijy/>

Sinervo, P. 2017. Lankojen numerointijärjestelmät. Accessed 12.11.2018. Retrieved from <https://punomo.fi/kuluttaja-ja-materiaalitieto-2/muu-kuluttajatieto/lankojen-numerointijarjestelmat/>

Singh, C. 2015. Java AWT tutorial for beginners. BeginnersBook. Accessed 12 March 2019. Retrieved from <https://beginnersbook.com/2015/06/java-awt-tutorial/>

The Apache™ FOP Project, 2016. Accessed 23.11.2018 Retrieved from <https://xmlgraphics.apache.org/fop/>

Vuorisalo, M. 2007b. MoA Master. Accessed 11 December 2018. Retrieved from [http://www.mastersofarts.fi/moa2007/index.php?option=com\\_moa\\_masters&task=view&memid=90&Itemid=32](http://www.mastersofarts.fi/moa2007/index.php?option=com_moa_masters&task=view&memid=90&Itemid=32)

Wielenga, G. 2018a. Apache NetBeans 9.0. Accessed 7.4.2019. Retrieved from <https://cwiki.apache.org/confluence/display/NETBEANS/Apache+NetBeans+9.0>

Wielenga, G. 2018b. Apache NetBeans 10. Accessed 7.4.2019. Retrieved from <https://cwiki.apache.org/confluence/display/NETBEANS/Apache+NetBeans+10/>

Wielenga, G. 2019. Apache NetBeans 11.0. Accessed 7.4.2019. Retrieved from <https://cwiki.apache.org/confluence/display/NETBEANS/Apache+NetBeans+11.0>

Yrityksen tiedot. 2015, Patentti ja rekisterihallitus, Accessed 11.4.2015. retrieved from [https://virre.prh.fi/novus/companySearch?\\_flowExecutionKey=\\_cD76E229D-EAF9-92C8-FEB4-9A8C228E4855\\_k9DB60285-8D5A-F168-5110-2E0B8CD0D524](https://virre.prh.fi/novus/companySearch?_flowExecutionKey=_cD76E229D-EAF9-92C8-FEB4-9A8C228E4855_k9DB60285-8D5A-F168-5110-2E0B8CD0D524)

## Ryijy instructions at Kotiliesi magazine from year 1931

300 — KOTILIESI 1931

## OMATEKOINEN RYIJYNI.

Kotiliedelle piirtänyt ja kirjoittanut *Ilona Jalava*.

Tähän aikaan, jolloin ryijyjä kudotaan ja ommellaan vähän joka talossa ja ostetaan vielä sen lisäksi, on syytä huomauttaa, että ryijy on todella arvokas esine, mutta sen arvo vaatii, että se tehdään hyvin ja kunnollisista aineista.

Loimilangoiksi on kalastajalanka parasta (18-säikeinen) ja kudolangoiksi, so. välikuteeksi, kotimainen sukkalanka. Välikude on lyötävä tiukkaan ja nukitus tehtävä huolellisesti, niin että ryijyä sietää tarkastaa sekä oikealta että nurjalta puolelta. — Mallit vaihtuvat muotien ja makusuuntien mukaan, värit muuttaa aika ja aurinko, mutta hyvin tehty työ pitää sadankin vuoden päästä arvonsa.

Kudottu ryijy on aina asetettava etusijalle — se on se »oikea». Mutta ommellunkaan olemassaoloa ei voi estää, on vain katsottava, että se vastaa mahdollisimman hyvin asemaansa sen »oikean» sijaisena ja varjona. Itse työ on niin hauskaa, että ihmiset tekevät sitä jo vain senkin vuoksi. Ja virkanainenkaan ei laske lomaaikansa rahaksi, vaan uskoo ryijyn tulevan halvemmaksi itse ommellen.

Valmiiksi kankaalle piirretty ryijy on verraten helposti tehty. Mutta ruutumallin mukaan om-

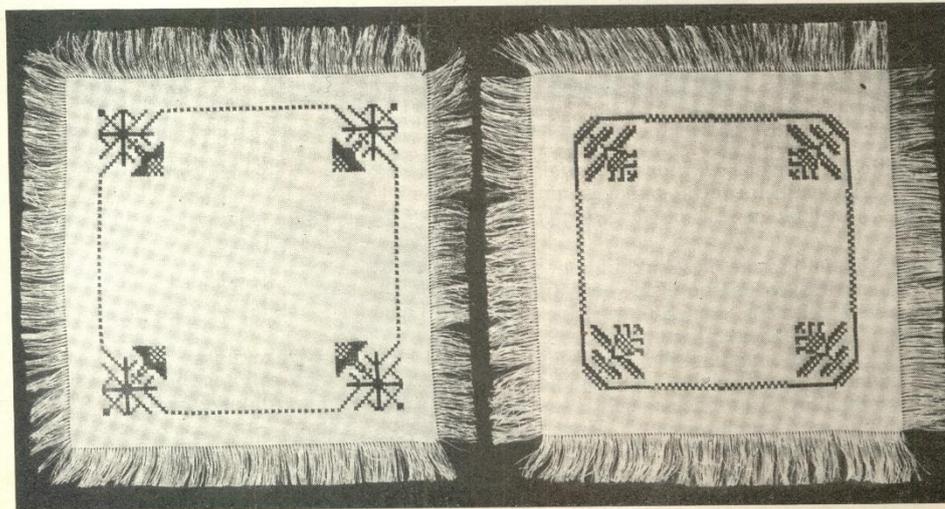
meltu muistuttaa kudottua sittenkin paremmin ja siihen saa kudotun täsmällisyyden varmemmin. Ommeltava kangas on huolellisesti valittava, sen pitää olla kestävä, se ei saa »syödä» nukkalankoja ja sen pitää olla siksi tiheätä, että nukat pysyvät siinä tomuja siitä piiskattaesakin. Ja nukat on välttämättä ommeltava lankasuoraan. Kun vinoon ommeltu ryijy pannaan seinälle, alkaa kangas vähitellen vetäytyä omien lakiensa pakosta suoraksi — ja silloin ilmiantaa ryijy auttamattomasti huolimattoman tekijänsä. Sellainen on ryijyn arvon häpäisemistä — ja sellainen kosta aina itsensä. — Ruutumallin mukaan ompeleminen voi ehkä ihan alussa tuntua hermostuneesta ihmisestä hieman hankalalta, mutta se hankaluus on pian voitettu ja vaivan paljitsee se iloinen varmuus, että kaikki käy matemaattisella tarkkuudella ja työn tulos kiittää tekijäänsä.

Mutta kuitenkin se, jonka »kauneuden kaipuu» vaatii ympäristöönsä todellakin arvokasta, se keksii myös säästää menoistään sen summan, jolla ostaa oikean ryijyn iltainsa iloksi.

Kuvamme esittämä ryijy »Takkavalkea» on suuruudeltaan 100 × 110 cm. Ryijyn väritys on pu-

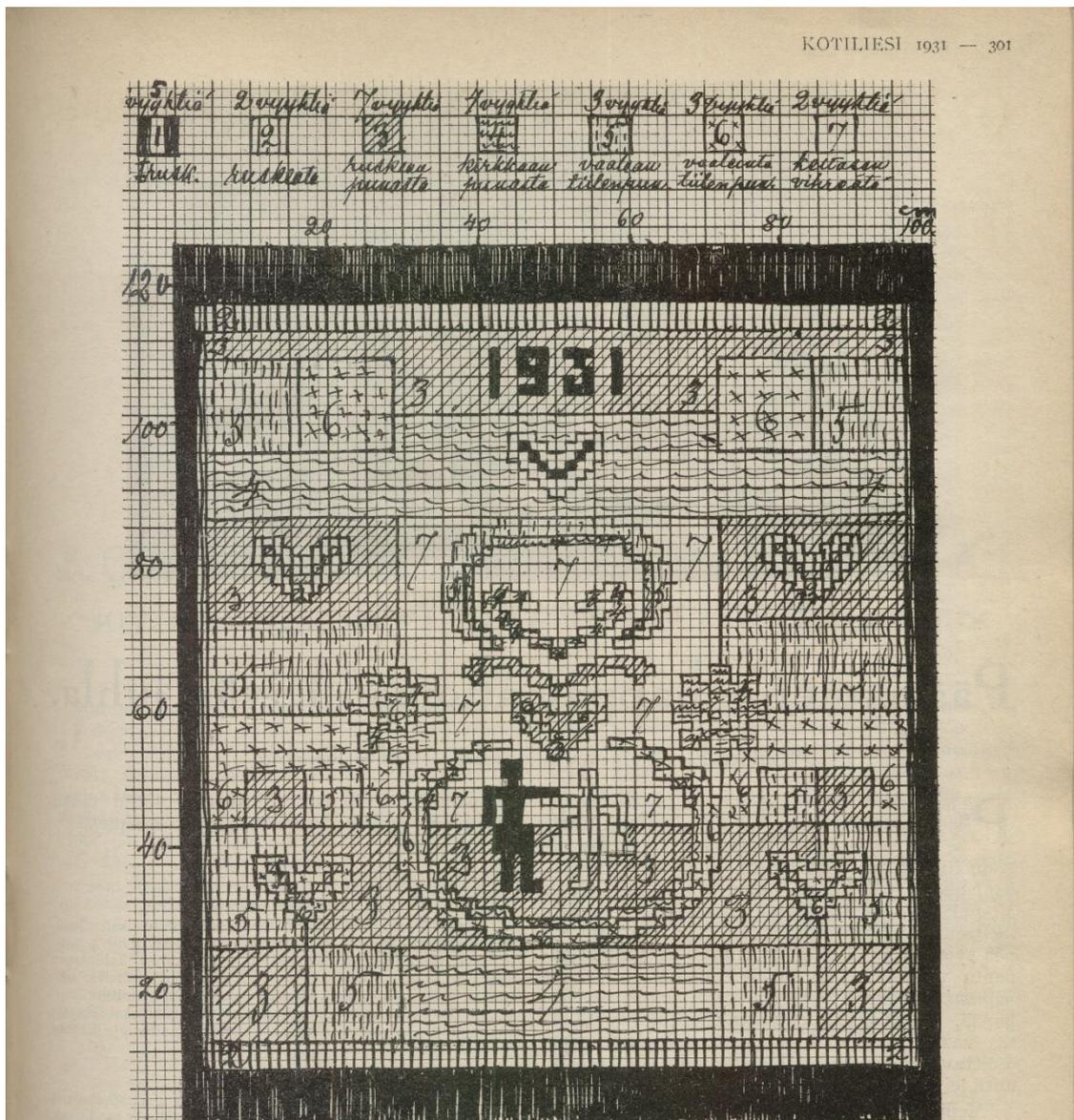


»Takkavalkea» *Ilona Jalava*.  
Malli seuraavalla sivulla.



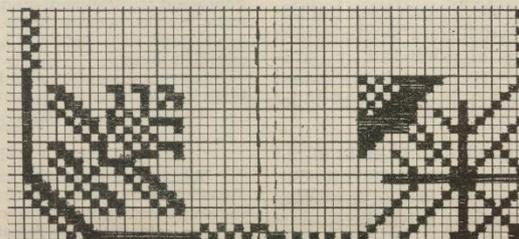
Kaksi sievää ristipistoliinaa. Mallit *Ilona Jalavan*. Ruutumalli viereisellä sivulla. Liinat ommellaan pellavaiselle kanavakankaalle punaisin tai sinisin nimikoimislangoin. — Mitä hauskin käsityö, joka ei lisäselityksiä kaivanne.

## Ryijy instructions at Kotiliesi magazine from year 1931



Ryijy »Takkavalkea», suuruus 100 × 110 cm. — Piirtänyt Ilona Jalava.

naisen puhuva ja värit on valittava niin, että ne erottuvat toisistaan, mutta kuitenkin kauniisti sulautuvat toisiinsa. Ryijyn ulkoreuna on tummanruskeaa ja sen vieressä vähän vaaleamman ruskeaa raita. 3:lla merkityt kohdat ovat ruskean punaista, 4:illä



Ristipistoliinujen ruutumallit.

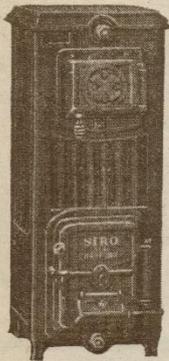
kirkkaan punaista (keltaiseen vivahtavaa), 5:illä merkityt vaalean tiilen punaista, 6:lla vielä siitakin vaaleampaa ja 7:illä merkitty pohja keskellä ryijyä on vihertävän keltaista. Keskellä ryijyä olevista lapsista on poika tummanruskeaa ja tyttö ruskeaa.

(Jatkuu siv. 316.)

## Ryijy instructions at Kotiliesi magazine from year 1931

316 — KOTILIESI 1931

# Suosikaa kotimaista teollisuutta



SIRO N:o 4  
(tulipinta 1,4 m<sup>2</sup>)

## SIRO-

kattila huone-  
ustojen keskus-  
lämmitystä varten

Valmiina ja saatavana varastosta.

HÖGFORSIN TEHDAS OY.

Väärentämättömillä mitoilla ammennettu kauneus tuottaa kotiin viihtyisyyttä ja kohottaa kodin hyvää henkeä.

Kauneus tarjoutuu avuksi, kun aletaan haikailla, mitä olisi tehtävä kotien suojelemiseksi rappeutumasta.

Emanuel Kanajärven avara maailmankatsoisuus ja savupirtin naisen harras perintöesineen kunnioittaminen ovat rinnastettavissa harvinaiseen maljaan ja ihanaan solkeen, jotka vuosituhansien kuluttua ovat kaikki, mitä on jäänyt tallelle muinaisesta sotien runtelemasta sivistyskansasta.

Ajatusteni paine lientyy.

Kiitos vastauksestasi, Emanuel Kanajärvi, veljeni kumpuin yössä.

### Kun meillä on mekkipäivät.

(Jatkoa siv. 304).

mansikkahilloa tms. Hyytelö kaadetaan sokereileivillä tai ohuilla kääretortunviipaleilla sisustettuun vuokaan tai pyöreään kulppiin, annetaan hyytyä ja kaadetaan tarjoiluvadille. Yksinkertaisia jälkiruokia, kuten esim. marjakiiseleitä kermavaahdon kera, keitetyjä hedelmiä, hedelmäsalaattia jne. on helppo valmistaa; mutta jos on juhlapäivällisistä kysymys, valmistettakoon mieluummin mehusta hyytelöä kuin marjakiiseleitä.

Heti päivällisten päätyttyä tarjotaan hyvää, tarpeeksi väkevää kahvia, tietenkin ilman leipää. Myöhemmin tarjoillaan välillä hedelmiä ja makeisia sekä vichy- tai meluvettä. Hedelmiä ei tarvitse välttämättömästi olla, mutta makeismaljan annetaan tavallisesti kiertää. Myöhemmin illalla noin yhdentoista aikaan tarjoillaan pieniä voileipiä. Voileivät järjestetään sievästi tarjottimelle, jolle on levitetty lautasliina tai torttupaperi, ja voileipiä varten varataan asetteja. Toiselta tarjottimelta tarjoillaan valmiiksi kaadetut kaljalasit. Laseja ei yleensä pidä täyttää enemmän kuin puolilleen. Ken haluaa, voi samalla kertaa tarjoilla teetä leivän kera, mutta rehelliselle voileivälle annettaneen kuitenkin enemmän arvoa.

Kun on päästy näin pitkälle, on emäntä täysin huolehtinut vieraittensa ruumiillisesta hyvinvoinnista. *Kaikkien*, niinhyvin vieraiden kuin isäntäväen velvollisuus on ylläpitää viihtyisyyttä ja hyvää tunnelmaa.

### Omatekoinen ryijyni.

(Jatkoa siv. 301.)

Langat ryijyyn on laskettu sen mukaan, että kudotaan 2 lankaa nukkaan. Kauniimmaksi ja tuuheammaksi tulee se tietysti, jos panee 3 lankaa — mutta tietysti  $\frac{1}{3}$  kalliimmaksikin. Kaide n:o 30 on sopivin.

Osto-osaston kautta tilattuina maksavat »Takkavalkean» langat (2 lankaa nukassa) Smk. 195: — + postimaksu. Jos ryijyyn pannaan 3 lankaa nukkaan, tulee hintaan  $\frac{1}{3}$  lisää ja maksavat siis langat Smk. 260: — + postimaksu.

Pirkanmaa Sirkka Ryijy Yarn Colour Chart



## Ryijytin Software Versions Timeline

