

TALLINNA TEHNIKAÜLIKOOL  
Infotehnoloogia teaduskond

Ksenija Ivanova 185300IACB

# **Sudoku lahendaja**

Bakalaureuse töö

Juhendaja: Peeter Ellervee  
PhD

Tallinn 2021

## **Autorideklaratsioon**

Kinnitan, et olen koostanud antud lõputöö iseseisvalt ning seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on töös viidatud.

Autor: Ksenija Ivanova

17.03.21

## **Annotatsioon**

Peamiseks selle töö ülesandeks oli kirjutada Sudoku mängu lahendaja, mille rõhuks on mugav ja arusaadav kasutajaliides, lisaks sellele ka üheks eesmärkidest oli kirjutada seda autonoomselt platvormiüleseks. Praktilise töö kirjutamiseks tuli uurida suur hulk erinevaid võtteid, kuidas Sudokut saab lahendada ja valida nendest ühte, lisaks sellele on töös toodud ka platvormi valiku põhjendused, teiste inimeste poolt kasutajaliidese küsimustiku järgi hinnang, kasutajaliidese vigade ja edu analüüs, valitud algoritmi ja koodi seletus.

Töös on toodud kõik pusle mängu reeglite seletused ja terminid koos seletustega. Edasi on seletusi erinevatest lahendusmeetoditest, millest siin töös on tehtud otsus, millise algoritmi kirjutada.

Lõpptulemuseks sai tehtud töötav tarkvara, mugava ja arusaadava kasutajaliidese, mille mootor ja kasutajaliides on sõltumatud.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 38 leheküljel, 6 peatükki, 36 joonist, 6 tabelit.

# **Abstract**

## **Sudoku solver**

The main objective of this Bachelor's thesis was to write a Sudoku game solver, which focuses on a comfortable and easy-to-understand user interface, which must give hints. In addition, one of the goals was to write it cross-platform compatible. The practical objective was to research the different techniques of how to solve Sudoku and choose one of them, there are the justifications included of why the platform was chosen, the assessment of the user interface inquired by a questionnaire, the analysis of the successes and faults of the user interface and the explanation of the chosen algorithm and code.

In order to select the most fitting algorithm research had to be conducted about the rules of Sudoku and the already existing solving algorithms. The choices were the following: The Backtracking algorithm, the Stochastic method, Knuth's algorithm X and various human techniques of solving Sudoku. During the research, the Backtracking algorithm was chosen.

The programming language of this project was C++ and QML. Qt Creator was chosen as the main IDE because it can work with C++ and QML. It is a cross-platform IDE, that can implement software to computers, mobile phones and embedded systems. The game engine and the user interface were made with this program.

The Backtracking algorithm consists of several steps. The first one is that an empty location must be chosen. Several numbers are tried on that empty location. If no number fits, the algorithm goes back to the previous location.

All objectives of this thesis were solved. An easy-to-understand user interface that gives hints was made. The chosen solving algorithm was implemented and integrated with the user interface.

The thesis is in Estonian and contains 38 pages of text, 6 chapters, 36 figures, 6 tables.

## Lühendite ja mõistete sõnastik

Adobe Color wheels	Veebi rakendus, kus saab teha värvide kombinatsioone
API	<i>Application programming interface</i> , rakendusliides
Boolean	Andmetüüp, mis saab olla ainult tõene või väär
Cross-platform	Mitme platvormi peal töötamise võimalus
Data	Andmed
Engine	Mootor
Field	Mängu välja massiiv koodis
Nishio	Sudoku lahendamise meetod
Num	Muutuja koodis, mis tähendab numbrit
X-Wings	Sudoku lahendamise meetod

## Sisukord

1 Sissejuhatus .....	11
2 Sudoku lahendamise võimalikud meetodid .....	13
2.1 Stohhastiline meetod.....	13
2.2 Tagurdamise meetod.....	13
2.3 Algoritm X.....	14
2.4 Alternatiivsed meetodid vastavalt raskusele.....	15
2.4.1 Lihtsad Sudokud .....	15
2.4.2 Keskmised Sudokud .....	16
2.4.3 Sudokud edasijõudnutele.....	19
2.4.4 Meistrite Sudokud .....	20
2.4.5 Veel raskemad Sudokud .....	26
2.5 Järeldused algoritmi valiku puhul.....	27
3 Platvormisõltumatuse analüüs .....	28
3.1 Qt Creator .....	28
3.2 Töökäik Qt Creatoris .....	28
3.2.1 Kasutajaliides .....	29
4 Kasutajaliidese tagasiside küsitlus ja analüüs .....	33
4.1 Küsimustiku vastuste analüüs.....	33
4.1.1 Kasutajaliidese välimus .....	33
4.1.2 Kasutajaliidese värvi muutumise võimalus .....	34
4.1.3 Kasutajaliidese arusaadavus .....	35
4.1.4 Võrdlus teiste mängu rakendustega.....	36
4.2 Teised tagasisided ja muutused .....	37
5 Programmi arhitektuur ja algoritmi realiseerimine .....	42
5.1 Üldine struktuur.....	42
5.2 Mängu mootor .....	43
5.2.1 <i>StartGame</i> funktsioon.....	43
5.2.2 <i>MakeMove</i> , <i>Legal</i> ja <i>ClearCell</i> funktsioonid .....	46
5.2.3 <i>Solve</i> funktsioon .....	46

5.2.4 <i>Value</i> ja <i>Hint</i> funktsioonid.....	47
6 Kokkuvõte .....	49
Kasutatud kirjandus .....	50
Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks .....	51
Lisa 2 – Tagasiside küsimustik.....	52

## Jooniste loetelu

Joonis 1. Puuduvate numbrite kohad reas [3].....	16
Joonis 2. Näide ainukestest kandidaaditest [3].....	16
Joonis 3. Kui ruutu läheb 4, siis teisi neljade variante võib kustutada veerust [3].....	17
Joonis 4. Variantide mahavõtmine [3].....	17
Joonis 5. Leitakse paare [3] .....	18
Joonis 6. Ülejäänud variandid kustutatakse [3] .....	18
Joonis 7. Keskmise Sudoku kolmanda variandi näide [3].....	18
Joonis 8. Paari leidmine [3] .....	19
Joonis 9. Peidetud paar keskmises ruudus [3] .....	20
Joonis 10. <i>X - Wingsi</i> näide [3].....	21
Joonis 11. Võimalikute käikude valik [3].....	21
Joonis 12. Kustutatud väli [3].....	22
Joonis 13. Numbri jaoks koha valik [3].....	22
Joonis 14. Tekkinud kinnine ahel [3] .....	23
Joonis 15. Esimene vastuste paigaldamise variant [3] .....	23
Joonis 16. Teine vastuste paigaldamise variant [3] .....	24
Joonis 17. Üleliigsete numbrite kustutamine [3] .....	24
Joonis 18. Ruutude paari valik [3].....	25
Joonis 19. Kolmas meetod meistrite Sudokust. Esimene samm [3].....	25
Joonis 20. Kolmas meetod meistrite Sudokust. Teine samm [3] .....	26
Joonis 21. Kolmas meetod meistrite Sudokust. Kolmas samm [3] .....	26
Joonis 22. Värvide valik [6] .....	29
Joonis 23. Esileht.....	30
Joonis 24. Raskustaseme valik .....	30
Joonis 25. Mängu aken koos kõikide nuppudega ja mänguvõimalustega .....	31
Joonis 26. Rekordite aken.....	32
Joonis 27. Helesinine kasutajaliides .....	38
Joonis 28. Roosa kasutajaliides .....	38
Joonis 29. Roheline kasutajaliides.....	39



Joonis 30. Muudetud mänguvälja asukoht .....	40
Joonis 31. Nupu seletus .....	40
Joonis 32. C++ Engine klass .....	43
Joonis 33. Programmi skeem.....	44
Joonis 34. Mängu alustamise funktsioon.....	45
Joonis 35. Sudoku lahendamise algoritm [8] .....	47
Joonis 36. Vihje andmise funktsioon.....	48

## Tabelite loetelu

Tabel 1. Knuth algoritmi tabel.....	14
Tabel 2. Mängu välimus .....	34
Tabel 3. Värvide valiku olulisus.....	34
Tabel 4. Värvide valiku võimalus .....	35
Tabel 5. Mängu arusaadavus .....	36
Tabel 6. Rakenduse võrdlemine teistega .....	37

# 1 Sissejuhatus

Selles töös seletatakse Sudoku mängu lahendajat. Sudoku on loogiline puslemäng, mille põhimõtteks on sisestada üheksa korda üheksa ruudustikku numbreid – üks kuni üheksa ja mängu alguses on vastavalt Sudoku raskusele on ette antud mingi hulk arvu ette. Kuna mängu peamised alad, mida vaadatakse on read, veerud ja kolm korda kolm ruudud, siis ei saa numbrid olla dubleeritud ühes reas, veerus või ruudus.

Käesoleva töö ülesandeks on:

1. Realiseerida C++ ja QML keeltes Sudoku lahendusalgorithm
2. Teha mugav kasutajaliides
3. Teha töötavad rakendust mängimiseks
4. Võimaldada kasutajal küsida programmis vihjet
5. Võimaldada rakendusel lahendada mängu

Lõputöö koosneb neljast osast. Esimeseks on Sudoku lahendamise algoritmide uuringust ja andmete põhjal järelduste tegemisest. Lisaks on ka seletatud Sudoku reeglid, millele töös tuginetakse. Kuna antud töös algoritm ei vaja väga suurt lahenduse kiirust ja põhirõhk on suunatud kasutajaliidese kirjutamisele, oli esimeseks valikuks kõige lihtsam algoritm, mille nimeks on jõurünne või teisiti tagurdamine, mis võib olla aeglasem, aga lihtsam kirjutamiseks.

Teiseks osaks on platvormisõltumatuse analüüs, kus uuritakse, mis see tähendab ja milliste riistade kasutamist siin töös eelistatakse selleks, et teha rakendust võimalikult platvormisõltumatuks. Peatükis on näidatud kõik valitud programmi eelised ja võimalused, mille pärast valik langes just Qt Creator IDE peale. Lisaks on peatükis lühidalt seletatud kasutajaliidese töötamine koos näidetega ja seletustega, kuidas see töötab.

Kolmandaks on kasutajate käest tagasiside küsimine ja vastuste põhjal analüüsi tegemine. Küsitluse eesmärgiks oli võrrelda antud töö kasutajaliidest teise populaarse juba tehtud

mänguga. Küsitlus on tehtud tuttavate vahel ja palutud olla võimalikult objektiivsed vastamisel, kuna see võib tugevasti mõjutada töö lõpptulemust.

Neljandaks osaks on algoritmi detailne seletus ja näited sellest, kuidas on kasutajaliides ja algoritm omavahel seotud. Samuti on lisatud mõned koodi näited ja kogu koodi saab vaadata git-is.

Töö tulemuseks on töötav rakendus ja kõik ülesanded on täidetud. Uuringu käigus on selgitatud töös olevad nõrgad kohad ja lisatud mõned võimalused kasutaja jaoks, mida eelnevalt ei olnud planeeritud.

## **2 Sudoku lahendamise võimalikud meetodid**

Sudokul on olemas mitmeid algoritme, nende seas: tagurdamine, Stohhastiline meetod, Algoritm X, lisaks nendele on olemas erinevaid variante ja teooriat, kuidas lahendada Sudoku. Selles peatükis uuritakse erinevaid meetodeid mängu lahendamiseks ja ka seletatakse, kuidas töötavad juba tuntud algoritmid koodi kirjutamiseks

### **2.1 Stohhastiline meetod**

Sudokut saab lahendada juhusliku meetodiga. See tähendab, et tühjad kohad juhuslikult täidetakse arvudega, peale seda loetakse vigade arv vastavalt reeglitele, kui kõik vead on selgunud, segatakse arve ära niikaua, kui vigade arv on null.

Sellega on mäng lahendatud. Ära segamise võtted võivad sisaldada simuleeritud lõõmutamist, geneetilist algoritmi ja tabu otsingu. Stohhastilise meetodi lahendused on üpris kiired, samas aga võivad olla aeglasemad, kui deduktiivse meetodiga lahendatud Sudokud. [1]

### **2.2 Tagurdamise meetod**

Meetodit nimetatakse jõu kasutuse algoritmiks, sest seda kasutatakse proovimise võttega. Seda meetodit võib nimetada kõige lihtsamaks nii koodi kirjutamise kui ka loogika arusaamise poolest. Algoritmi põhimõte seisneb selles, et antud Sudoku variandis otsitakse esimest tühja kohta, millele üritatakse panna üks, kui see sobib, siis liigutakse järgmise tühja koha juurde, kui aga üks ei sobi, siis üritatakse panna järgmist numbrit ehk kahte, juhul kui ei sobi ükski variant minnakse tagasi eelmise koha juurde. Tagurdamise meetodi eeliseks on selle lihtsus, kuigi sellega võib tulla probleeme, näiteks programmi töötamise aeg võib olla mahukas. Mõnel erilisel juhul võib programm arvutusi teha mitu tundi või isegi päeva. Tavalise Sudokuga aga ei peaks olema probleeme, et programm on aeglane. Niikaua kui Sudoku on lahendatav, siis see meetod garanteerib lahendust. [1]

## 2.3 Algoritm X

See algoritm lahendab diskreetset puslet või teisiti lahendab hulga täpse kattuvuse ülesannet. On olemas hulk  $S$  ning mitu alamhulka  $Y$ . Ülesanne seisneb selles, et kõikide  $Y^*$  kokkupanek tekitab ja kattub hulgaga  $S$  ja  $Y^*$  ei oma paaris olevaid kattuvaid hulki.  
[2]

$S = \{1, 2, 3, 4, 5\}$  ja

$Y = \{ A = \{1, 2\},$

$B = \{1, 3\},$

$C = \{2, 5\},$

$D = \{3, 4\},$

$E = \{5\} \}$

Tabel 1. Knuth algoritmi tabel

$Y_i/S_j$	1	2	3	4	5
A	1	1	0	0	0
B	1	0	1	0	0
C	0	1	0	0	1
D	0	0	1	1	0
E	0	0	0	0	1

Lahenduse algoritm näeb selline välja:

1. Kui hulk  $S$  on tühi, siis hulkade pinu on kaetud.
2. Kui hulk  $Y$  on tühi, siis katmist pole võimalik leida.
3. Hulgast  $S$  otsime  $s$  elementi, mis esineb kõige harvemini hulk  $Y$  alamhulgast  $X$ .

4. Otsime kõik read, kus on olemas  $s$ .
  5. Iga hulga jaoks peab läbima järgmised punktid.
  6. Lisame  $X$ -i pinusse kui potentsiaalset katmise osa.
  7. Looime hulgas  $S'$  ja  $Y'$ , kus  $S'$  on  $S$ , kust on kustutatud kõik elemendid, mis kuuluvad hulka  $X$  ja  $Y'$  on hulk  $Y$ -st, mis ei kattu  $X$ -ga.
  8.  $S'$ ,  $Y'$  ja pinu jaoks kutsume  $X$  Algoritmi välja.
  9. Kui punktis kolm on selgunud, et katmine on võimatu, võtame pinu elemendi välja ja liigume järgmise  $X$ -i juurde. Kui lahendus eksisteerib, siis paneme selle tagasi.
  10. Kui lahendust pole mitte ühegi  $X$ -i jaoks, siis selline ülesanne ei ole lahendatav.
- [2]

## 2.4 Alternatiivsed meetodid vastavalt raskusele

Suur osa Sudokudest, mida inimesed igapäevaselt näevad, on lihtsamat tüüpi variandid, kus lahenduse saamine on intuiitiivne, samas on aga olemas variandid, kus inimesel läheks vaja pliiatsit ja loogilist mõtlemist, et Sudoku lahendada. Selles peatükis seletatakse lihtsamatest kuni raskemateni variantideni, milliste meetoditega saab mängu lahendada.

### 2.4.1 Lihtsad Sudokud

Lihtsamat tüüpi mängu saab lahendada kahe meetodiga: üksikpositsiooni ja ainukese kandidaati põhjal.

Esimene nendest seisneb selles, et valitakse kõige rohkem täietud rida,  $3 \times 3$  ruut või veerg ja kontrollitakse, millised numbrid on seal puudu. Vastavalt sellele vaadatakse, mitu varianti on olemas selle lahtri täitmiseks, halvemal juhul on neid kaks või kolm, aga heal juhul võib olla ainult üks variant, mille saab sisestada (näide joonisel 1). [3]

		6		3		7		8
	3							1
2						6		
1			3	5				6
	7	9		4		1	5	
5				1	7			4
		2						7
6							8	
4		7		6		2		

Joonis 1. Puuduvate numbrite kohad reas [3]

Teine meetod on lihtne, kui eelnevalt tehtud töö on ja pliiaatsiga juba kirjutatud, kuhu lahtrisse, missugused väärtused sobivad.

<sup>1</sup> <sub>5</sub>	<sup>3</sup> <sub>5 8 9</sub>	4	<sup>1</sup> <sub>9</sub>	<sup>1 3</sup> <sub>5 9</sub>	6	<sup>2</sup> <sub>7</sub>	2	<sup>5</sup> <sub>7</sub>
<sup>2</sup> <sub>5 8</sub>	<sup>3</sup> <sub>5 6</sub>	7	8	<sup>2 3</sup> <sub>4 5</sub>	<sup>2</sup> <sub>4 5</sub>	9	1	<sup>4 5 6</sup>
<sup>1 2</sup> <sub>5 6 9</sub>	<sup>5 6</sup> <sub>9</sub>	<sup>1 2</sup>	<sup>1 2</sup> <sub>4 9</sub>	<sup>1 2</sup> <sub>4 5 9</sub>	<sup>2</sup> <sub>4 5 7</sub>	3	<sup>4 5</sup> <sub>7</sub>	8
<sup>4 5 6</sup> <sub>7</sub>	1	8	3	<sup>4 5 6</sup>	<sup>4 5</sup>	2	<sup>4 5 9</sup> <sub>7</sub>	<sup>4 5 9</sup> <sub>7</sub>
3	<sup>5 6</sup>	<sup>2</sup>	7	8	9	<sup>4</sup>	<sup>4 5</sup>	1
<sup>2</sup> <sub>4 5 7</sub>	<sup>5</sup> <sub>7</sub>	9	<sup>2</sup> <sub>4 5</sub>	<sup>2</sup> <sub>4 5</sub>	1	<sup>4</sup> <sub>7 8</sub>	6	<sup>4 5 3</sup> <sub>7</sub>
8	<sup>7 9</sup>	3	<sup>1 2</sup> <sub>4 6 9</sub>	<sup>1 2</sup> <sub>4 6 9</sub>	<sup>2</sup> <sub>4 7</sub>	5	<sup>4</sup> <sub>7 9</sub>	<sup>4</sup> <sub>7 9</sub>
<sup>1</sup> <sub>7 9</sub>	4	5	<sup>1 2</sup> <sub>9 7 9</sub>	<sup>1 2</sup> <sub>9 7 9</sub>	3	6	<sup>7 8 9</sup>	<sup>2</sup> <sub>7 9</sub>
<sup>7 9</sup>	2	6	5	<sup>4</sup> <sub>7 9</sub>	<sup>4</sup> <sub>7 8</sub>	1	<sup>4 3</sup> <sub>7 8 9</sub>	<sup>4 3</sup> <sub>7 9</sub>

Joonis 2. Näide ainukestest kandidaaditest [3]

Eelneva uuringuga tuleb välja, et mitmes kohas on jäänud ainult üks kandidaat, et täita lahter (näide joonisel 2). Peale kõikide kandidaatide sisestamist kitseneb ka teistes kohtades olev variantide arv üheni. [3]

## 2.4.2 Keskmised Sudokud

Kuigi suurem arv lahendavatest Sudokudest on lihtsad, võivad mõned variandid minna keerulisemaks, edaspidi tuleb mitu meetodit, mis aitavad vähendada variantide arvu või sisestada õiget arvu keskmise raskusega mängus.



Esimeseks on meetod, mis aitab välja selgitada, missugused variandid ei sobi sisestamiseks. Kui vaadata 3x3 ruutu ja tuleb välja, et seal on joon, kuhu saab paigutada samasuguseid arve, siis on selge, et vahet pole, kuhu tulevikus pannakse neid arve, nad peavad olema sellel joonel (näide joonisel 3). See võib aidata kõrvaldada mõne variandi, mis teeb otsimispiirkonda kitsemaks (näide joonisel 4). [3]

$\begin{smallmatrix} 2 \\ 4 \\ 8 \end{smallmatrix}$	$\begin{smallmatrix} 2 \\ 4 \\ 8 \end{smallmatrix}$	1	9	5	7	$\begin{smallmatrix} 2 \\ 4 \\ 8 \end{smallmatrix}$	6	3
$\begin{smallmatrix} 2 \\ 4 \\ 5 \end{smallmatrix}$	$\begin{smallmatrix} 2 \\ 4 \\ 3 \\ 5 \end{smallmatrix}$	$\begin{smallmatrix} 3 \\ 5 \end{smallmatrix}$	8	$\begin{smallmatrix} 2 \\ 4 \end{smallmatrix}$	6	$\begin{smallmatrix} 1 \\ 2 \\ 4 \\ 9 \end{smallmatrix}$	7	$\begin{smallmatrix} 1 \\ 2 \\ 4 \\ 9 \end{smallmatrix}$
7	6	9	1	3	$\begin{smallmatrix} 2 \\ 4 \end{smallmatrix}$	8	$\begin{smallmatrix} 2 \\ 4 \end{smallmatrix}$	5
$\begin{smallmatrix} 4 \\ 8 \\ 9 \end{smallmatrix}$	$\begin{smallmatrix} 4 \\ 8 \end{smallmatrix}$	7	2	6	1	3	5	$\begin{smallmatrix} 4 \\ 9 \end{smallmatrix}$
3	1	2	4	9	5	7	8	6
$\begin{smallmatrix} 4 \\ 9 \end{smallmatrix}$	5	6	3	7	8	$\begin{smallmatrix} 1 \\ 2 \\ 4 \\ 9 \end{smallmatrix}$	$\begin{smallmatrix} 1 \\ 2 \\ 4 \\ 9 \end{smallmatrix}$	$\begin{smallmatrix} 1 \\ 2 \\ 4 \\ 9 \end{smallmatrix}$
1	$\begin{smallmatrix} 2 \\ 3 \end{smallmatrix}$	8	6	$\begin{smallmatrix} 2 \\ 4 \end{smallmatrix}$	9	5	$\begin{smallmatrix} 2 \\ 3 \\ 4 \end{smallmatrix}$	7
$\begin{smallmatrix} 2 \\ 5 \end{smallmatrix}$	9	$\begin{smallmatrix} 3 \\ 5 \end{smallmatrix}$	7	1	$\begin{smallmatrix} 2 \\ 4 \end{smallmatrix}$	6	$\begin{smallmatrix} 2 \\ 3 \\ 4 \end{smallmatrix}$	8
6	7	4	5	8	3	$\begin{smallmatrix} 1 \\ 2 \\ 9 \end{smallmatrix}$	$\begin{smallmatrix} 1 \\ 2 \\ 9 \end{smallmatrix}$	$\begin{smallmatrix} 1 \\ 2 \\ 9 \end{smallmatrix}$

Joonis 3. Kui ruutu läheb 4, siis teisi neljade variante võib kustutada veerust [3]

$\begin{smallmatrix} 2 \\ 4 \\ 8 \end{smallmatrix}$	$\begin{smallmatrix} 2 \\ 4 \\ 8 \end{smallmatrix}$	1	9	5	7	$\begin{smallmatrix} 2 \\ 4 \\ 8 \end{smallmatrix}$	6	3
$\begin{smallmatrix} 2 \\ 4 \\ 5 \end{smallmatrix}$	$\begin{smallmatrix} 2 \\ 4 \\ 3 \\ 5 \end{smallmatrix}$	$\begin{smallmatrix} 3 \\ 5 \end{smallmatrix}$	8	$\begin{smallmatrix} 2 \\ 4 \end{smallmatrix}$	6	$\begin{smallmatrix} 1 \\ 2 \\ 4 \\ 9 \end{smallmatrix}$	7	$\begin{smallmatrix} 1 \\ 2 \\ 4 \\ 9 \end{smallmatrix}$
7	6	9	1	3	$\begin{smallmatrix} 2 \\ 4 \end{smallmatrix}$	8	$\begin{smallmatrix} 2 \\ 4 \end{smallmatrix}$	5
$\begin{smallmatrix} 4 \\ 8 \\ 9 \end{smallmatrix}$	$\begin{smallmatrix} 4 \\ 8 \end{smallmatrix}$	7	2	6	1	3	5	$\begin{smallmatrix} 4 \\ 9 \end{smallmatrix}$
3	1	2	4	9	5	7	8	6
$\begin{smallmatrix} 4 \\ 9 \end{smallmatrix}$	5	6	3	7	8	$\begin{smallmatrix} 1 \\ 2 \\ 4 \\ 9 \end{smallmatrix}$	$\begin{smallmatrix} 1 \\ 2 \\ 4 \\ 9 \end{smallmatrix}$	$\begin{smallmatrix} 1 \\ 2 \\ 4 \\ 9 \end{smallmatrix}$
1	$\begin{smallmatrix} 2 \\ 3 \end{smallmatrix}$	8	6	$\begin{smallmatrix} 2 \\ 4 \end{smallmatrix}$	9	5	$\begin{smallmatrix} 2 \\ 3 \\ 4 \end{smallmatrix}$	7
$\begin{smallmatrix} 2 \\ 5 \end{smallmatrix}$	9	$\begin{smallmatrix} 3 \\ 5 \end{smallmatrix}$	7	1	$\begin{smallmatrix} 2 \\ 4 \end{smallmatrix}$	6	$\begin{smallmatrix} 2 \\ 3 \\ 4 \end{smallmatrix}$	8
6	7	4	5	8	3	$\begin{smallmatrix} 1 \\ 2 \\ 9 \end{smallmatrix}$	$\begin{smallmatrix} 1 \\ 2 \\ 9 \end{smallmatrix}$	$\begin{smallmatrix} 1 \\ 2 \\ 9 \end{smallmatrix}$

Joonis 4. Variantide mahavõtmine [3]

Teine variant seisneb paaride otsimisel. Selleks on vaja otsida kaks kasti, kus kahes reas või veerus saavad olla samasugused väärtused (näide joonisel 5). Kolmandas kastis selle abil saab kõrvaldada neid kahte joont, sinna see arv juba tuleb igal juhul, seega jääb ainult üks joon, kuhu saab antud arvu paigaldada (näide joonisel 6). [3]

9	3	4	<sup>1 2</sup> <sub>7</sub> 6	<sup>2</sup> <sub>7</sub> 6	<sup>1</sup> <sub>7 8</sub> 5	<sup>1</sup> <sub>8</sub> 5
<sup>1</sup> <sub>7</sub> 6	<sup>1</sup> <sub>5</sub> 6	<sup>1</sup> <sub>7 8</sub> 6	<sup>1</sup> <sub>7 8</sub> 4	<sup>5</sup> <sub>7 8</sub> 4	<sup>1</sup> <sub>7</sub> 9	<sup>1</sup> <sub>8</sub> 2 3
<sup>1 2</sup> <sub>7</sub> 8	<sup>1 2</sup> <sub>5</sub> 8	<sup>1 2 3</sup> <sub>7</sub> 8	<sup>1</sup> <sub>7</sub> 9	<sup>2</sup> <sub>7 5</sub> 3	<sup>1</sup> <sub>7</sub> 4	<sup>1</sup> <sub>8</sub> 6
<sup>1 2</sup> <sub>9</sub> 8	<sup>1 2 3</sup> <sub>9</sub> 8	<sup>1 2 3</sup> <sub>9</sub> 5	<sup>1</sup> <sub>7</sub> 4	<sup>1</sup> <sub>7</sub> 6	<sup>1</sup> <sub>3</sub> 7	<sup>1</sup> <sub>9</sub> 7
<sup>4</sup> <sub>9</sub> 6	<sup>4</sup> <sub>9</sub> 7	<sup>3</sup> <sub>7 9</sub> 6	<sup>2</sup> <sub>7 8</sub> 1	<sup>2</sup> <sub>7 8</sub> 1	<sup>4</sup> <sub>8</sub> 5	<sup>8 9 5</sup>
<sup>1</sup> <sub>4</sub> 5	<sup>1</sup> <sub>7</sub> 5	<sup>1</sup> <sub>7</sub> 3	<sup>1</sup> <sub>7 8</sub> 9	<sup>1</sup> <sub>7 8</sub> 9	<sup>1</sup> <sub>4</sub> 6	<sup>1</sup> <sub>8</sub> 2
<sup>1</sup> <sub>8</sub> 3	<sup>5</sup> <sub>9</sub> 6	<sup>5</sup> <sub>9</sub> 4	<sup>5</sup> <sub>8</sub> 4	<sup>5</sup> <sub>8</sub> 1	<sup>1</sup> <sub>8</sub> 2	<sup>8 9 7</sup>
<sup>1 2</sup> <sub>9</sub> 4	<sup>2</sup> <sub>9</sub> 7	<sup>2</sup> <sub>9</sub> 6	<sup>2 3</sup> <sub>8</sub> 6	<sup>2 3</sup> <sub>8 9</sub> 5	<sup>1</sup> <sub>8</sub> 5	<sup>1</sup> <sub>8 9</sub> 3
<sup>1 2</sup> <sub>9</sub> 8	<sup>1 2</sup> <sub>5</sub> 8	<sup>2</sup> <sub>7</sub> 8	<sup>2</sup> <sub>7</sub> 6	<sup>2</sup> <sub>7</sub> 5	<sup>2</sup> <sub>7 9</sub> 6	<sup>2</sup> <sub>8</sub> 3 4

Joonis 5. Leitakse paare [3]

9	3	4	<sup>1 2</sup> <sub>7</sub> 6	<sup>2</sup> <sub>7</sub> 6	<sup>1</sup> <sub>7 8</sub> 5	<sup>1</sup> <sub>8</sub> 5
<sup>1</sup> <sub>7</sub> 6	<sup>1</sup> <sub>5</sub> 6	<sup>1</sup> <sub>7 8</sub> 6	<sup>1</sup> <sub>7 8</sub> 4	<sup>5</sup> <sub>7 8</sub> 4	<sup>1</sup> <sub>7</sub> 9	<sup>1</sup> <sub>8</sub> 2 3
<sup>1 2</sup> <sub>7</sub> 8	<sup>1 2</sup> <sub>5</sub> 8	<sup>1 2 3</sup> <sub>7</sub> 8	<sup>1</sup> <sub>7</sub> 9	<sup>2</sup> <sub>7 5</sub> 3	<sup>1</sup> <sub>7</sub> 4	<sup>1</sup> <sub>8</sub> 6
<sup>1 2</sup> <sub>9</sub> 8	<sup>1 2 3</sup> <sub>9</sub> 8	<sup>1 2 3</sup> <sub>9</sub> 5	<sup>1</sup> <sub>7</sub> 4	<sup>1</sup> <sub>7</sub> 6	<sup>1</sup> <sub>3</sub> 7	<sup>1</sup> <sub>9</sub> 7
<sup>4</sup> <sub>9</sub> 6	<sup>4</sup> <sub>9</sub> 7	<sup>3</sup> <sub>7 9</sub> 6	<sup>2</sup> <sub>7 8</sub> 1	<sup>2</sup> <sub>7 8</sub> 1	<sup>4</sup> <sub>8</sub> 5	<sup>8 9 5</sup>
<sup>1</sup> <sub>4</sub> 5	<sup>1</sup> <sub>7</sub> 5	<sup>1</sup> <sub>7</sub> 3	<sup>1</sup> <sub>7 8</sub> 9	<sup>1</sup> <sub>7 8</sub> 9	<sup>1</sup> <sub>4</sub> 6	<sup>1</sup> <sub>8</sub> 2
<sup>1</sup> <sub>8</sub> 3	<sup>5</sup> <sub>9</sub> 6	<sup>5</sup> <sub>9</sub> 4	<sup>5</sup> <sub>8</sub> 4	<sup>5</sup> <sub>8</sub> 1	<sup>1</sup> <sub>8</sub> 2	<sup>8 9 7</sup>
<sup>1 2</sup> <sub>9</sub> 4	<sup>2</sup> <sub>9</sub> 7	<sup>2</sup> <sub>9</sub> 6	<sup>2 3</sup> <sub>8</sub> 6	<sup>2 3</sup> <sub>8 9</sub> 5	<sup>1</sup> <sub>8</sub> 5	<sup>1</sup> <sub>8 9</sub> 3
<sup>1 2</sup> <sub>9</sub> 8	<sup>1 2</sup> <sub>5</sub> 8	<sup>2</sup> <sub>7</sub> 8	<sup>2</sup> <sub>7</sub> 6	<sup>2</sup> <sub>7</sub> 5	<sup>2</sup> <sub>7 9</sub> 6	<sup>2</sup> <sub>8</sub> 3 4

Joonis 6. Ülejäänud variandid kustutatakse [3]

Kolmas variant on sarnane eelmisele, kuigi on raskem.

<sup>2</sup> <sub>4 7 8</sub> 9	<sup>2</sup> <sub>4 5 8</sub> 9	<sup>2</sup> <sub>5</sub> 3	<sup>2</sup> <sub>7</sub> 6	<sup>2</sup> <sub>5</sub> 6	<sup>1 2</sup> <sub>4</sub> 8	<sup>1 4</sup> <sub>7</sub> 9
<sup>2</sup> <sub>7</sub> 3	<sup>2</sup> <sub>5</sub> 6	<sup>2</sup> <sub>7</sub> 1	<sup>2</sup> <sub>5</sub> 4	<sup>2</sup> <sub>7</sub> 8	<sup>2</sup> <sub>7</sub> 9	<sup>2</sup> <sub>8</sub> 9
<sup>1</sup> <sub>4</sub> 1	<sup>2</sup> <sub>4</sub> 8	<sup>4</sup> <sub>7</sub> 8	<sup>4</sup> <sub>7</sub> 6	<sup>4</sup> <sub>7</sub> 9	<sup>2</sup> <sub>4 7</sub> 3	<sup>2</sup> <sub>5</sub> 5
<sup>2 3</sup> <sub>4 5 7</sub> 9	<sup>4 5</sup> <sub>7</sub> 9	<sup>1 2 3</sup> <sub>4 5 7</sub> 8	<sup>1 2 3</sup> <sub>5 7</sub> 6	<sup>1 2 3</sup> <sub>5 7</sub> 8	<sup>2</sup> <sub>4 5 6</sub> 8	<sup>2 3</sup> <sub>4 5 6</sub> 4
<sup>2 3</sup> <sub>4 5 7</sub> 1	<sup>4 5</sup> <sub>7</sub> 1	<sup>2 3</sup> <sub>4 5 7</sub> 8	<sup>2 3</sup> <sub>5 7 8</sub> 6	<sup>2 3</sup> <sub>5 7 8</sub> 9	<sup>2</sup> <sub>4 5</sub> 9	<sup>2 3</sup> <sub>4 6</sub> 9
<sup>2 3</sup> <sub>4 5</sub> 6	<sup>2 3</sup> <sub>4 5</sub> 8	<sup>2 3</sup> <sub>4 5</sub> 9	<sup>2 3</sup> <sub>5</sub> 9	<sup>2 3</sup> <sub>5</sub> 1	<sup>2 3</sup> <sub>4</sub> 7	<sup>2 3</sup> <sub>4</sub> 3
<sup>4 5</sup> <sub>8</sub> 6	<sup>4 5</sup> <sub>8</sub> 1	<sup>5</sup> <sub>7 8</sub> 9	<sup>5</sup> <sub>7 8</sub> 3	<sup>4 5</sup> <sub>7</sub> 4	<sup>4 5</sup> <sub>8</sub> 2	<sup>4 5</sup> <sub>8</sub> 2
<sup>4 5</sup> <sub>8</sub> 9	<sup>4 5</sup> <sub>8</sub> 7	<sup>1</sup> <sub>5</sub> 2	<sup>1</sup> <sub>5</sub> 6	<sup>1</sup> <sub>5</sub> 4	<sup>1</sup> <sub>5</sub> 3	<sup>1</sup> <sub>8</sub> 5
<sup>4 5</sup> <sub>8</sub> 8	<sup>4 5</sup> <sub>8</sub> 3	<sup>1</sup> <sub>7</sub> 5	<sup>1</sup> <sub>7</sub> 2	<sup>1</sup> <sub>7</sub> 9	<sup>1</sup> <sub>4 5 6</sub> 9	<sup>1</sup> <sub>4 7 8</sub> 6

Joonis 7. Keskmise Sudoku kolmanda variandi näide [3]

Raskus seisneb selles, et see töötab samamoodi, aga nüüd kandidaate võib olla mitu ja ka kaste võib olla mitu (näide joonisel 7). Tavaliselt see meetod ei aita paigaldada lahtrisse valmis vastust, aga aitab edaspidiseks Sudoku lahendamiseks. [3]

### 2.4.3 Sudokud edasijõudnutele

Kuna Sudokud lähevad aina raskemaks, siis edasi tulevad enamasti variandid, mis aitavad kõrvaldada kandidaate, mitte aga saada kohe lahendust.

Esimeseks raskemate Sudokude puhul on olemas meetod, mis on suunatud kandidaatide kustutamiseks lahtritest, otsides numbrite paare või kolmikuid. Seda meetodit saab kasutada juba enamasti lahendatud mängu puhul. Kui ühes reas on leitud koht, kus on näiteks olemas ainult kaks samasugust arvu, mis sinna võivad sobida, siis seda nimetatakse numbri paariks (näide joonisel 8). Täpselt ei ole teada, mis kahest sinna sobib, aga on kindel, et mingeid muid arve sinna paigutada ei saaks. Sellega saab järeldada, et neid arve ei saa eksisteerida teistes kohtades selles reas, kuna see paar sobib ainult nendesse kahte lahtrisse. Sama põhimõttega saab leida ka kolmikuid ja eemaldada neid variante teistest lahtritest. [3]

4	<sup>1</sup> <sub>5</sub>	<sup>3</sup> <sub>5</sub>	2	7	<sup>3</sup> <sub>9</sub>	6	<sup>1</sup> <sub>8 9</sub>	<sup>5</sup> <sub>8</sub>
7	9	8	1	5	6	2	3	4
<sup>1</sup> <sub>6</sub>	2	<sup>3</sup> <sub>5 8</sub>	8	4	<sup>3</sup> <sub>9</sub>	<sup>1</sup> <sub>5</sub>	<sup>1</sup> <sub>9</sub>	7
2	3	7	4	6	8	9	5	1
8	4	9	5	3	1	7	2	6
5	6	1	7	9	2	8	4	3
<sup>3</sup> <sub>6</sub>	8	2	<sup>3</sup> <sub>6</sub>	1	5	4	7	9
<sup>1</sup> <sub>6 8</sub>	7	<sup>5</sup> <sub>6</sub>	<sup>6</sup> <sub>9</sub>	2	4	3	<sup>1</sup> <sub>6</sub>	<sup>5</sup> <sub>8</sub>
<sup>1</sup> <sub>6 8</sub>	<sup>1</sup> <sub>5</sub>	4	<sup>3</sup> <sub>6 9</sub>	8	7	<sup>1</sup> <sub>5</sub>	<sup>1</sup> <sub>6</sub>	2

Joonis 8. Paari leidmine [3]

Teiseks variandiks on peidetud paarid ja kolmikud, see variant sobib nendele, kes juba oskavad kasutada esimest, sest peidetud paare ja kolmikuid on leida raskem. Peidetud meetodi puhul samamoodi valitakse mingi joon ja vaadatakse, mis variandid sinna sobivad ning kirjutatakse üles.

2	<sup>3</sup> <sub>6</sub>	<sup>3</sup> <sub>5 6</sub>	<sup>5</sup> <sub>8</sub>	4	<sup>3</sup> <sub>5 6 9</sub>	1	<sup>8</sup> <sub>9</sub>	7	
9	<sup>3</sup> <sub>7 6</sub>	<sup>1 3</sup> <sub>5 6 7</sub>	<sup>1</sup> <sub>7</sub>	<sup>5 6</sup> <sub>8</sub>	<sup>1 3</sup> <sub>5 6</sub>	2	4	<sup>3</sup> <sub>8</sub>	
8	4	<sup>1 3</sup> <sub>7</sub>	2	<sup>3</sup> <sub>7</sub>	<sup>1 3</sup> <sub>9</sub>	5	6	<sup>3</sup> <sub>9</sub>	
7	1	2	4	9	8	3	5	6	
6	<sup>3</sup> <sub>8</sub>	<sup>3</sup> <sub>8</sub>	<sup>1</sup> <sub>7</sub>	<sup>2</sup> <sub>5</sub>	<sup>1 2</sup> <sub>5</sub>	4	<sup>7</sup> <sub>9</sub>	<sup>1</sup> <sub>9</sub>	
5	9	4	6	<sup>3</sup> <sub>7</sub>	<sup>1 3</sup> <sub>7 8</sub>		2	<sup>1</sup> <sub>8</sub>	
4	5	<sup>6</sup> <sub>8</sub>	3	<sup>6</sup> <sub>8</sub>	7	9	1	2	
1	<sup>2</sup> <sub>7 8</sub>	<sup>6</sup> <sub>7 8</sub>	9	<sup>2</sup> <sub>8</sub>	<sup>6</sup> <sub>7 8</sub>	4	<sup>6</sup> <sub>7 8</sub>	3	5
3	<sup>2</sup> <sub>7 8</sub>	9	<sup>5</sup> <sub>8</sub>	1	<sup>2</sup> <sub>5 6</sub>	<sup>6</sup> <sub>7 8</sub>	<sup>6</sup> <sub>7 8</sub>	4	

Joonis 9. Peidetud paar keskmises ruudus [3]

Vahepeal on märgata, et isegi siis, kui ühes lahtris on olemas unikaalne paar, ning teises see sama paar, aga lisa võimaliku väärtusega, saab järeldada, et kolmas sinna ei sobi (näide joonisel 9). See juhtub sellepärast, et kuskil mujal peale neid kahte lahtrit see number ei eksisteeri ja ainukesed kaks variandi nende jaoks on need kaks kohta. See tähendab, et „peidetud“ lahtrist saab uurimise käigus selgunud mitte sobiliku numbri kustutada. [3]

#### 2.4.4 Meistrite Sudokud

Meistri Sudoku jaoks on olemas variandid, mis teevad lahendamist kergemaks ja põhinevad ka nagu eelmised sellele, et eelnevalt on inimene juba teinud märkmeid, kuhu lahtrisse sobivad teatud arvud.

Esimeseks meetodit *X-Wings* on raskem aru saada, kui eelmisi meetodeid. Selle põhimõtteks on see, et kui on olemas kaks rida (joont), mis omavad samasuguseid väärtusi. Kui on selge, et pole lihtsamaid variante lahendamiseks saab vaadata kahte joont, kuhu mõlemasse saab kahte kohta panna ühte arvu, näiteks 6. Edasi oleks vaja ühele ruudule anda see väärtus ja vaadata, mida see tähendab ühejäänute lahtrite jaoks ja kuidas mäng siis läheb edasi. Seda nimetatakse *X-Wingsiks*, sest selle ülevaatamine toimub ristina (näide joonisel 10). [3]

9	<sup>6</sup> <sub>8</sub>	<sup>2</sup> <sub>4</sub>	<sup>2</sup> <sub>4</sub>	5	1	7	3	<sup>6</sup> <sub>8</sub>	
1	<sup>4</sup> <sub>6</sub>	7	3	9	8	2	<sup>4</sup> <sub>6</sub>	5	
5	<sup>4</sup> <sub>8</sub>	<sup>3</sup> <sub>4</sub>	<sup>2</sup> <sub>3</sub>	<sup>2</sup> <sub>4</sub>	7	6	<sup>4</sup> <sub>8</sub>	9	1
8	1	<sup>6</sup> <sub>8</sub>	7	2	4	3	5	<sup>6</sup> <sub>8</sub>	
2	<sup>4</sup> <sub>3</sub>	<sup>4</sup> <sub>3</sub>	<sup>3</sup> <sub>9</sub>	1	6	5	<sup>4</sup> <sub>8</sub>	<sup>6</sup> <sub>8</sub>	7
<sup>4</sup> <sub>6</sub>	7	5	9	8	3	<sup>4</sup> <sub>6</sub>	1	2	
<sup>4</sup> <sub>6</sub>	2	1	5	3	7	<sup>6</sup> <sub>8</sub>	<sup>4</sup> <sub>6</sub>	<sup>4</sup> <sub>6</sub>	<sup>6</sup> <sub>8</sub>
7	5	8	6	4	9	1	2	3	
3	9	<sup>4</sup> <sub>8</sub>	8	1	2	5	7	<sup>6</sup> <sub>8</sub>	

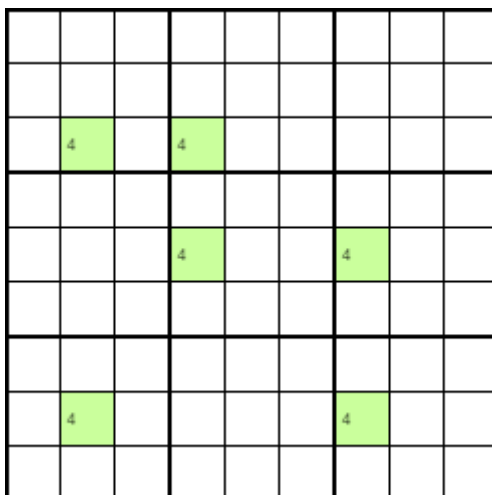
Joonis 10. X - Wingsi näide [3]

Teine variant on sarnane eelmisega. Selle raskus seisneb selles, et peab vaatama andmeid kolmest joonest samaaegselt, mis vajab erilist tähelepanu. Seda meetodit kasutatakse jällegi siis, kui suur osa mängust juba lahendatud ja mingeid teisi variante eelmistest ei saa rakendada.

1	9	5	3	6	7	2	4	8
<sup>2</sup> <sub>4</sub>	7	8	<sup>1</sup> <sub>4</sub>	<sup>2</sup> <sub>5</sub>	<sup>1</sup> <sub>4</sub>	3	6	9
3	<sup>4</sup> <sub>2</sub>	6	<sup>4</sup> <sub>2</sub>	9	8	1	5	7
<sup>2</sup> <sub>6</sub>	<sup>1</sup> <sub>2</sub>	3	7	8	<sup>1</sup> <sub>4</sub>	5	9	<sup>2</sup> <sub>4</sub>
7	<sup>1</sup> <sub>2</sub>	9	<sup>1</sup> <sub>4</sub>	<sup>2</sup> <sub>3</sub>	5	<sup>4</sup> <sub>8</sub>	<sup>3</sup> <sub>8</sub>	6
5	8	4	9	<sup>2</sup> <sub>3</sub>	6	7	1	<sup>2</sup> <sub>3</sub>
8	3	2	5	4	9	6	7	1
9	<sup>4</sup> <sub>6</sub>	7	<sup>6</sup> <sub>8</sub>	1	3	<sup>4</sup> <sub>8</sub>	2	5
<sup>4</sup> <sub>6</sub>	5	1	<sup>6</sup> <sub>8</sub>	7	2	9	<sup>3</sup> <sub>8</sub>	<sup>4</sup> <sub>3</sub>

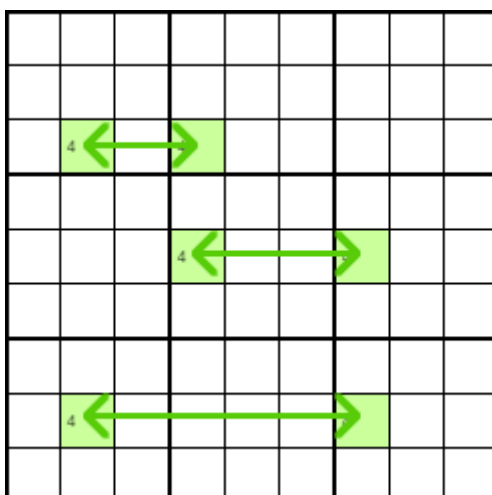
Joonis 11. Võimalikute käikude valik [3]

Sellel variandil olevat meetodit saab kasutada number 4-ga (näide joonisel 11). Esiteks peab eristama kõike kohti, kus on olemas 4 kandidaati. Siin peab samamoodi nagu eelmisel variandil otsima ahelat, mis koosneb neljast väärtusest, kuigi see meetod vajab 6 või enam väärtust ahelaks. [3]



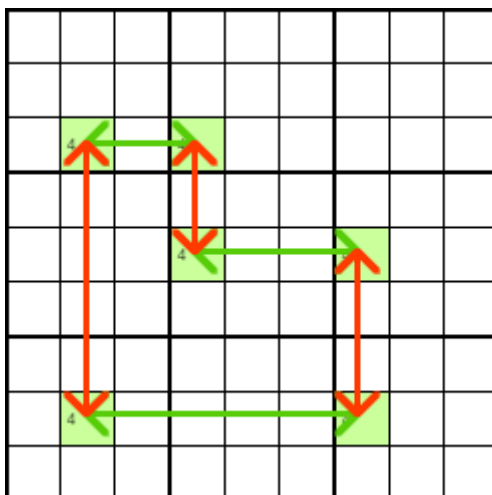
Joonis 12. Kustutatud väli [3]

Siin on näha meetodi rakendamise võimalust kolmes reas (3, 5 ja 8). Mugavuseks on siin ära kustutatud ülejäänud arvud (näide joonisel 12).



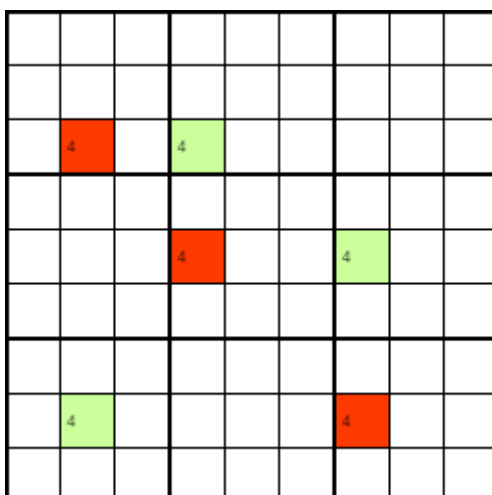
Joonis 13. Numbri jaoks koha valik [3]

Samuti nagu *X-Wingiga* aimatakse, et ühte kohta pannakse number 4 ja vaadatakse, kuidas see mõjutab teisi lahtreid (näide joonisel 13). Siin saab näha, et iga nöörlõpe mingis teise rea veerus (näide joonisel 14).



Joonis 14. Tekkinud kinnine ahel [3]

Tekib kinnine ahel. Selle variandi puhul olemas ainult kaks võimalust, kuidas Sudoku saab olla lahendatud (näide joonisel 15 ja 16).



Joonis 15. Esimene vastuste paigaldamise variant [3]

	4		4					
			4			4		
	4					4		

Joonis 16. Teine vastuste paigaldamise variant [3]

Väärtused on pandud ja sealt saab näha, et read on täidetud vastavalt veergudele. Edasi, kui väärtused on tagasi, saab eristada neid veerge veel kord ja üle vaadata, kust saab nelja ära kustutada (näide joonisel 17). [3]

1	9	5	3	6	7	2	4	8
<sup>4</sup> <sub>2</sub>	7	8	<sup>1 2</sup> <sub>8</sub>	5	<sup>1</sup> <sub>4</sub>	3	6	9
3	<sup>4</sup> <sub>2</sub>	6	<sup>4</sup> <sub>2</sub>	9	8	1	5	7
<sup>2</sup> <sub>6</sub>	<sup>1 2</sup> <sub>6</sub>	3	7	8	<sup>1</sup> <sub>4</sub>	5	9	<sup>4</sup> <sub>2</sub>
7	<sup>1 2</sup> <sub>6</sub>	9	<sup>1</sup> <sub>4</sub>	<sup>2 3</sup> <sub>8</sub>	5	<sup>4</sup> <sub>8</sub>	<sup>3</sup> <sub>8</sub>	6
5	8	4	9	<sup>2 3</sup> <sub>8</sub>	6	7	1	<sup>2 3</sup> <sub>8</sub>
8	3	2	5	4	9	6	7	1
9	<sup>4</sup> <sub>6</sub>	7	<sup>8</sup> <sub>6</sub>	1	3	<sup>4</sup> <sub>8</sub>	2	5
<sup>4</sup> <sub>6</sub>	5	1	<sup>8</sup> <sub>6</sub>	7	2	9	<sup>3</sup> <sub>8</sub>	<sup>4</sup> <sub>3</sub>

Joonis 17. Üleliigsete numbrite kustutamine [3]

Kolmandast meetodist on aru saada lihtsam, kui eelmistest, kuigi see nõuab rohkem aega. See meetod sobib siis, kui eksisteerib palju lahtreid, kus on olemas kaks valikuvarianti. Kui esimesse kohta sisestada üks variantidest, siis teine lahter saab olla ainult üks kahest variantidest. Ahelad võivad olla lühikesed, aga ka suuremad. Siin näiteks tuuakse üks võimalik variant (näide joonisel 18).



<small>4 7 8</small>	<small>6</small>	<small>1 2</small>	<small>6</small>	<small>1 2</small>	5	<small>8 6</small>	<small>2</small>	<small>1</small>	3	9
<small>8</small>	3	9	<small>8 6</small>	4	1	2	5	7		
<small>4</small>	<small>7</small>	<small>1 2</small>	5	<small>2</small>	3	9	6	8	<small>1</small>	<small>4</small>
<small>5</small>	<small>7</small>	9	<small>1</small>	3	2	<small>4</small>	<small>7</small>	<small>1</small>	6	<small>1</small>
<small>1 2 3</small>	<small>1 2</small>	8	9	5	6	7	4	<small>1</small>	<small>3</small>	
<small>5 6</small>	<small>7</small>	<small>6 4</small>	<small>3</small>	<small>4</small>	1	8	9	2	<small>5</small>	<small>3</small>
<small>2</small>	4	6	1	<small>7 8</small>	5	3	<small>7 9</small>	<small>6</small>	<small>2</small>	
<small>2 3</small>	8	7	<small>4 2</small>	<small>6</small>	9	<small>4 2 3</small>	5	1	<small>4 2</small>	<small>6</small>
<small>1 2 3</small>	5	<small>1 2 3</small>	<small>2</small>	<small>4 8</small>	<small>6</small>	<small>4</small>	<small>2 3</small>		<small>4 2</small>	<small>6</small>

Joonis 18. Ruutude paari valik [3]

Pole vahet, missugust väärtust sisestatakse, see mõjutab ahelat nii, et teise lahtrisse läheb viis. Kujutame ette, et ülemisesse lahtrisse läheb 1, siis see jätab variandiks ainult 4 veeru järgmises vabas kohas (näide joonistel 19, 20, 21).

<small>4 7 8</small>	<small>6</small>	<small>1 2</small>	<small>6</small>	<small>1 2</small>	5	<small>8 6</small>	<small>2</small>	<small>1</small>	3	9
<small>8</small>	3	9	<small>8 6</small>	4	1	2	5	7		
<small>4</small>	<small>7</small>	<small>1 2</small>	5	<small>2</small>	3	9	6	8	<small>1</small>	<small>4</small>
<small>5</small>	<small>7</small>	9	<small>1</small>	3	2	<small>4</small>	<small>7</small>	<small>1</small>	6	<small>1</small>
<small>1 2 3</small>	<small>1 2</small>	8	9	5	6	7	4	<small>1</small>	<small>3</small>	
<small>5 6</small>	<small>7</small>	<small>6 4</small>	<small>3</small>	<small>4</small>	1	8	9	2	<small>5</small>	<small>3</small>
<small>2</small>	4	6	1	<small>7 8</small>	5	3	<small>7 9</small>	<small>6</small>	<small>2</small>	
<small>2 3</small>	8	7	<small>4 2</small>	<small>6</small>	9	<small>4 2 3</small>	5	1	<small>4 2</small>	<small>6</small>
<small>1 2 3</small>	5	<small>1 2 3</small>	<small>2</small>	<small>4 8</small>	<small>6</small>	<small>4</small>	<small>2 3</small>		<small>4 2</small>	<small>6</small>

Joonis 19. Kolmas meetod meistrite Sudokust. Esimene samm [3]

<small>4 7 8</small>	<small>1 2 6 7</small>	<small>1</small>	5	<small>6 8 7</small>	<small>2</small>	<small>1 4</small>	3	9
<small>8</small>	3	9	<small>6</small>	4	1	2	5	7
<small>4 7</small>	<small>1 2 7</small>	5	<small>2 7</small>	3	9	6	8	<small>1 4</small>
<small>5 7</small>	9	<small>4</small>	3	2	<small>1 8</small>	6	<small>1 5 8</small>	
<small>1 2 3</small>	<small>1 2</small>	8	9	5	6	7	4	<small>1 3</small>
<small>5 6 7</small>	<small>6 4 3 7</small>	<small>4 7</small>	1	8	9	2	<small>5 3</small>	
<small>2 9</small>	4	6	1	<small>7 8</small>	5	3	<small>7 9 8</small>	<small>2</small>
<small>2 3</small>	8	7	<small>4 2 6</small>	9	<small>4 2 3</small>	5	1	<small>4 2 6</small>
<small>1 2 3 9</small>	5	<small>1 2 3</small>	<small>2 4 8 6 7 8</small>	<small>6 4 7 8</small>	<small>2 3 4</small>	<small>4 8 7 9</small>	<small>4 8 6</small>	<small>2 6</small>

Joonis 20. Kolmas meetod meistrite Sudokust. Teine samm [3]

<small>4 7 8</small>	<small>1 2 6 7</small>	<small>1</small>	5	<small>6 8 7</small>	<small>2</small>	<small>1 4</small>	3	9
<small>8</small>	3	9	<small>6</small>	4	1	2	5	7
<small>4 7</small>	<small>1 2 7</small>	5	<small>2 7</small>	3	9	6	8	<small>1 4</small>
<small>5 7</small>	9	<small>4</small>	3	2	<small>1 8</small>	6	<small>1 5 8</small>	
<small>1 2 3</small>	<small>1 2</small>	8	9	5	6	7	4	<small>1 3</small>
<small>5 6 7</small>	<small>6 4 3 7</small>	<small>4 7</small>	1	8	9	2	<small>5 3</small>	
<small>2 9</small>	4	6	1	<small>7 8</small>	5	3	<small>7 9 8</small>	<small>2</small>
<small>2 3</small>	8	7	<small>4 2 6</small>	9	<small>4 2 3</small>	5	1	<small>4 2 6</small>
<small>1 2 3 9</small>	5	<small>1 2 3</small>	<small>2 4 8 6 7 8</small>	<small>6 4 7 8</small>	<small>2 3 4</small>	<small>4 8 7 9</small>	<small>4 8 6</small>	<small>2 6</small>

Joonis 21. Kolmas meetod meistrite Sudokust. Kolmas samm [3]

Selle ahela abil sai panna teise lahtrisse välistavalt numברי 5. [3]

## 2.4.5 Veel raskemad Sudokud

Vahepeal võib juhtuda, et isegi nendest meetoditest, mis olid enne, ei piisa ja Sudoku ei ole lahendatud. Selle jaoks kõige parem variant on kas kasutada programmi, mis lahendab selle ise ära või proovida aimata, missugune väärtus võiks mingisse kohta sobida. Sellisel juhul see meenutaks tagurdamise algoritmi, mille kohta on eelnevalt juba kirjutatud. Kuigi see juba on peaaegu võimatu inimesel lahendada, on olemas veel üks võte, mis saaks lahendamist aidata.

Selle meetodi nimeks on *Nishio*. Selle põhimõtte seisneb selles, et ühes kohas, kus on olemas jällegi näiteks kaks kandidaati valitakse üks nendest ja lahendatakse Sudoku edasi. Kui veab, siis lahendus saab olla õige ja legalne reeglite järgi, kui aga valitud

number on vale, siis peab minema tagasi ja proovima teist variandi ja lahendada tervet teed uuesti. Peamine probleem siin on see, et vedamise peale mängimine võib olla tüütu ja loogilisi võimalusi mängu lahendamiseks, peale ära arvamise või proovimise teel valida kahest variandist, kahjuks ei ole olemas. Samas alati saab lasta arvutil kiiresti Sudoku ära lahendada huvi korral. [3]

## **2.5 Järeldused algoritmi valiku puhul**

Uurides kõike võimalusi valikuks osunud tagurdamise lahendusalgoritm, mis on sama efektiivne ja tagab võrreldes stohhastilise meetodiga kindlat tulemust, vaatamata Sudoku raskusele, kuigi miinuseks mõnel mitte tüüpilisel variandil võib olla selle kiirus. Antud töö eesmärgiks ei ole algoritmi kiirus, vaid mugav ja aursaadav kasutajaliides ning platvormisõltumatus. Samas aga lõpptulemuse saavutamisel saab algoritmi paremaks teha ja ümber kirjutada näiteks juba antud Knuthi X algoritmiga, sest see lahendab Sudoku paari millisekundiga, kuna töö on kirjutatud sõltumatult kasutajaliideselt. Mootorit saab ümber kirjutada viisil tulevikus paremal. Samas minu valiku puhul oluline ka see, et algoritm lahendaks ka inimese jaoks võimatuid mängu variante, mis on tehtud just tagurdamise meetodi jaoks, mis ei ole millegi muu jaoks tehtud.

### 3 Platvormisõltumatuse analüüs

*Cross-platform software* ehk mitme platvormi peal võimalik töötamine. See kujutab endast ette programmi, mis on võimeline töötama võimalikult paljudel platvormidel ja erinevate operatsioonisüsteemidega, mis saavutatakse tänu kõrgkeeltele ja API-dele, mis suudavad kompileerida koodi vastavalt erinevatele platvormidele.

*Cross-platformiliseks* saab nimetada suurema osa kõrgkeeltes, nagu näiteks C ja C++, mida kasutatakse selle lõputöö praktilise osa kirjutamiseks. Samuti oluline on selle puhul standardiseeritud teegid. Suurimad *Cross-platformilised* teegid on näiteks Qt, Boost, OpenGL, OpenAL jne.

#### 3.1 Qt Creator

QT Creator on siin töös valitud *cross-platform* IDE rakendustarkvara arendamiseks, mis on üks mugavamatest platvormidest, millega saab teha tööd C++ ja QML-ga. See töötab Windowsi, Linuxi ja MacOS-i operatsioonisüsteemidel, ehk seda saab kasutada peaaegu igas arvutis maailmas. Lisaks see IDE võimaldab rakendada tarkvara arvutite, mobiilide ja sardsüsteemide jaoks. Lisaks juba kirjutatud keeltele, võimaldab rakendus kirjutada koodi Pythoni, JavaScripti ja teistes keeltes. Sellel on olemas võimalus kasutada Git-i. [4]

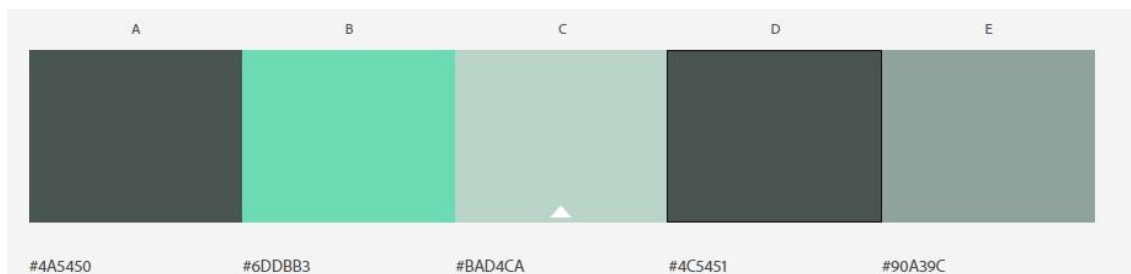
Lisaks eelneval on programmi tegijate veebil on olemas Qt Wiki, mugav dokumentatsioon ja arusaadavad õpetused ja seletused QML-ist ja Qt-st. Olemas seletus C++ ja QML klassidest, Qt Creatorist, kuidas alustada tööd, näited, lisavõimalused jne. [5]

#### 3.2 Töökäik Qt Creatoris

Üheks peamistest põhimõtetest peale platvormisõltumatuse töös oli see, et kasutajaliides oleks mootorist sõltumatu. Selle saab saavutada läbi eraldi kasutajaliidese ja loogika kirjutamise ja kasutades ühte koodiga lisafaili ühendada C++ ja QML-i failid. Sellega saab iga üks pärast optimeerida tehtud algoritmi ilma, et virtuaalne osa saaks kannatada.

### 3.2.1 Kasutajaliides

Kasutajaliidese värvi valik langes online värvi genereerimiseks, mis sai tehtud *Adobe Color wheel-is*, kus saab valida meeldivaid värvitoone ning sellega sobivaid värve (näide joonisel 22). [6]



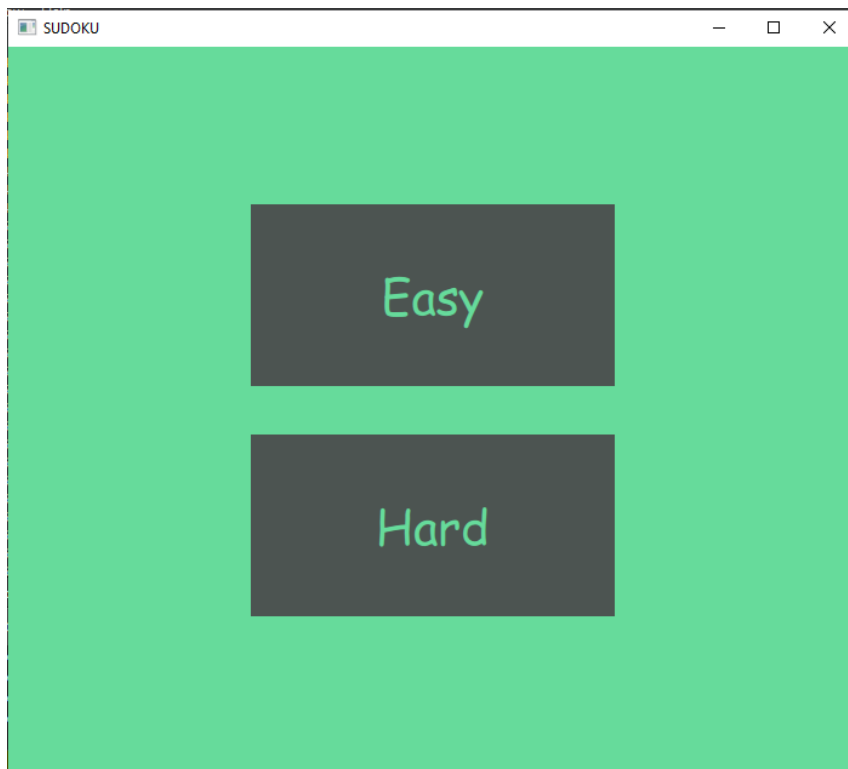
Joonis 22. Värvide valik [6]

Sellised rahulikud hallikad pastellivärvid peaksid olema mugavad inimese silma jaoks ja kasutaja ei peaks silmi pingesse panema. Samas aga on olemas ka üks värv, mis on rohkem pigmenteerunud, et kasutajaliides ei oleks ainult hall ja igav.

Mängu alguses näeb kasutaja stardilehte, kus on olemas valik kolmest variandist. Esimene on uus mäng, millele vajutades saab valida mängu raskust. Valikus on olemas „lihtne“ ja „raske“ raskusetasemed. Teiseks variandiks on nupp rekordid, kus mängija saab vaadata oma eelmisi mängu ja võrrelda oma ajalisi tulemusi ja eelnevalt valitud raskustasemeid (näide joonisel 24). Kolmas nupp on „välju“, millele vajutades saab mängu kinni panna (näide joonisel 23).

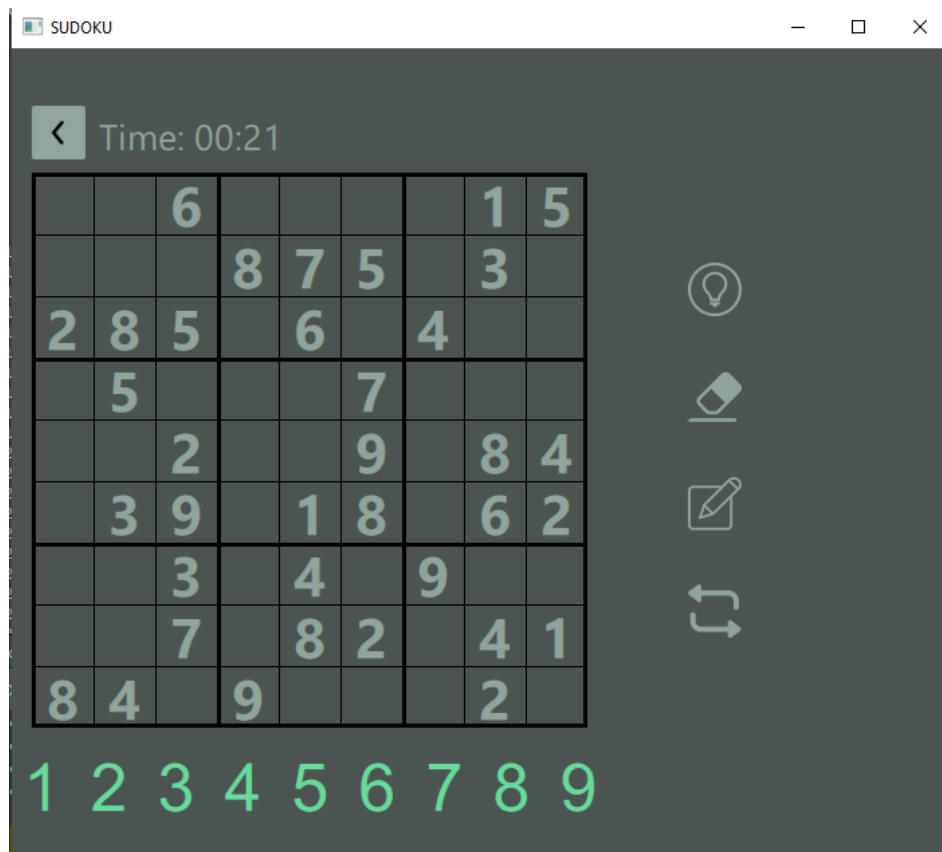


Joonis 23. Esileht



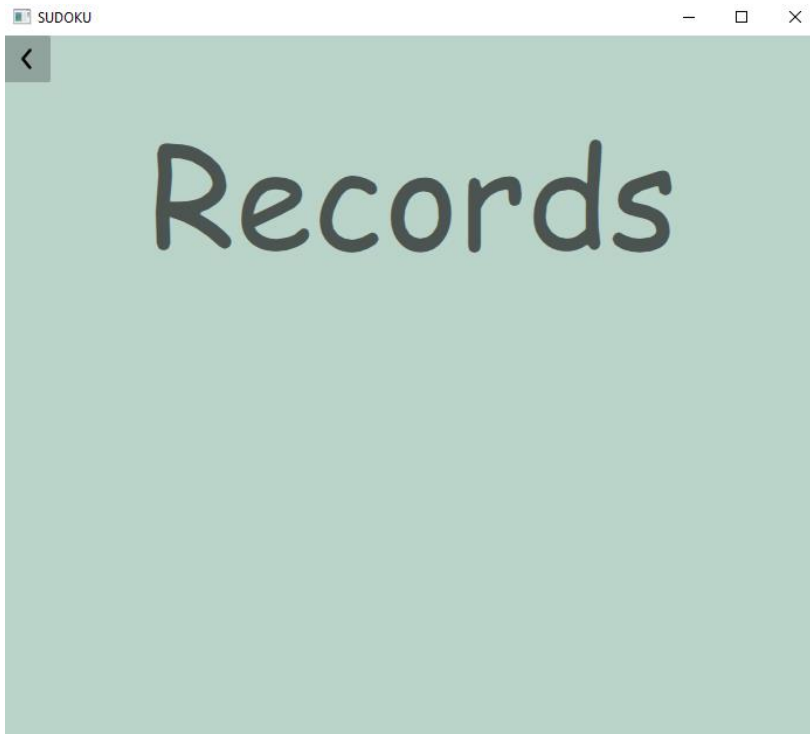
Joonis 24. Raskustaseme valik

Peale raskustaseme valikut tuleb juba leht mänguga, kus on mäng ise (Sudoku), taimer, vihje andmise võimalus, mängu uuesti alustamise võimalus, numברי kustutamise võimalus ja täieliku arvuti abil lahendamise võimalus (näide joonisel 25). Lisaks on mugavuse mõttes alla lisatud numbrid, mis annab võimaluse mängida ka ilma klaviatuuri kasutamisetä.



Joonis 25. Mängu aken koos kõikide nuppudega ja mänguvõimalustega.

Lisaks toon näiteks ka rekordite akna (näide joonisel 26).



Joonis 26. Rekordite aken



## **4 Kasutajaliidese tagasiside küsitlus ja analüüs**

Ilus ja mugav kasutajaliides on üks peamistest eesmärkidest selle töö puhul. Autori jaoks on alati tema tehtud töö parim, selle pärast on objektiivsuse mõttes hea, kui kasutaja saab mängu proovida ja anda tagasisidet. Selle eesmärgiga oli tehtud kasutajaliidese tagasiside küsimustik, kus küsiti mängu üldise välimuse, värvi valiku võimaluse ja kasutajaliidese arusaamise kohta. Lisaks uuritakse, kas teiste mängu variantide seast kasutaja eelistaks seda, mida ta on enne mänginud, kas ta ei ole üldse mänginud või kas ta valib selle lõputöö raames tehtud rakendust.

### **4.1 Küsimustiku vastuste analüüs**

Tagasiside on kokku andnud 13 inimest ja tagasiside tuli väga positiivne. Ainult mõned vastused on olnud negatiivsed ja nende kohta on inimesed saatnud lisa sõnumit, kus seletasid, mida oleks võinud teha paremini või mis nendele ei meeldinud, samuti on tulnud sõnumitesse ka positiivset tagasisidet. Kõike vastuseid uuritakse edasi ja otsustatakse, kas on mõistlik midagi muuta olemasolevas rakenduses.

#### **4.1.1 Kasutajaliidese välimus**

Küsimus oli kasutajaliidese üldise välimuse kohta ja kas inimesele, kes seda hindavad, see meeldib. Suuremas osas tulid vastused „Jah“. Ehk 11 inimest arvas, et praegune kasutajaliides on atraktiivne ja 2 vastajatest leidsid selle mitte nii ilusaks, nagu tahaksid (tabel 2). Üks tagasisidest kirjalikus vormis oli, et vastajale ei meeldinud, et kasutajaliides on liiga lame. Vastaja arvates oleks võinud nappud olla selgelt eristatud tagaplaanist. Üldise värvivalikuga olid rahul kõik, aga veel üks soovitus oli, et stardilehe tagaplaanil oleks mängukäigu pilt.

Uurides seda tagasisidet, tehti otsus, et ainukese tagasiside pärast ei vaja kasutajaliides muutumist. Teistest vastajatest ei ole tulnud muresid stardilehe kohta. Sellest võib järeldada, et tendents on inimeste seas pigem positiivne praegu. Ei ole teada, kas tulenevate muutustega ei oleks muutunud ka teiste vastajate arvamus negatiivsemate vastuste poole, sest pakutud muutused võivad teha kasutajaliidest vähem minimalistlikuks ja teha disaini üle täidetuks. Selle tõttu edasises analüüsis seda tagasiside kohta enam ei arutleta.

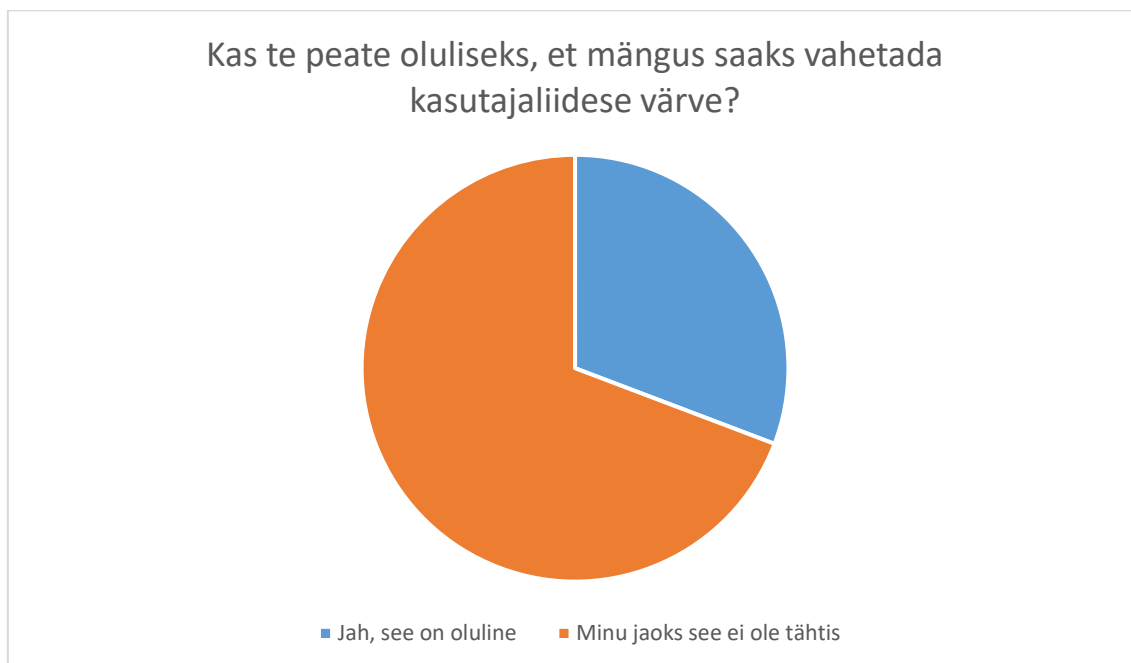
Tabel 2. Mängu välimus



#### 4.1.2 Kasutajaliidese värvi muutumise võimalus

Esimene küsimus oli selle kohta, kui palju inimesi on üldse huvitatud selles, et mängus oleks võimalus muuta värvi. Vastajatest 4 inimest ehk 30,8% vastasid, et nende jaoks on oluline värvi valik, ülejäänud 9 ehk 69,2% on vastanud, et nende jaoks värvi valiku võimalus mängu puhul ei ole tähtis (tabel 3).

Tabel 3. Värvi valiku olulisus



Teise küsimusega uuriti, kas kasutajale üldiselt meeldib, et antud mängus saab valida kasutajaliidese värvi ja siin on tulnud ootamatu tulemus, sest kõik 13 inimest ehk 100% on vastanud, et nendele meeldib, et mängus saab valida värvi (tabel 4). Sellest võib järeldada, et see osa on oluline mängude puhul, isegi siis, kui mäng ise ei ole kunagi väga esitatud ilusate disainidega, kui selline valik on olemas, siis kasutajale see meeldib. See on hea, kui valiku võimalus on ikka olemas. Seega, peale seda küsitlust, on mängu lisatud võimalus valida värvi kolmest valikust.

Tabel 4. Värvivaliku võimalus

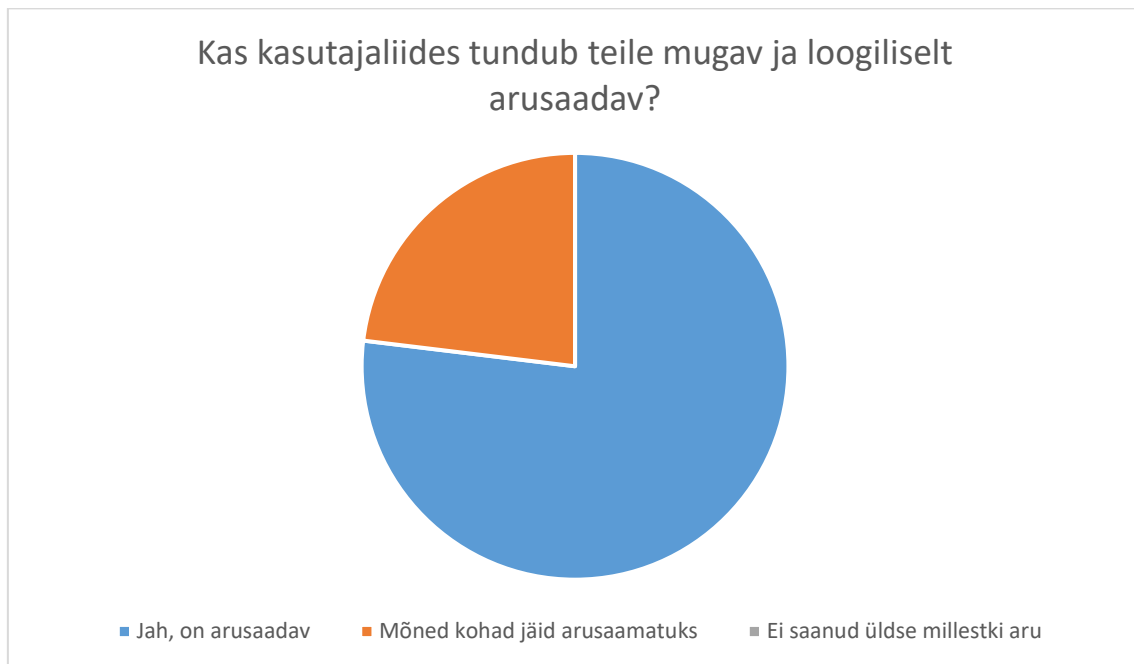


#### 4.1.3 Kasutajaliidese arusaadavus

Selle küsimusega küsiti vastajatelt, kas kasutajaliides on nende jaoks arusaadav. Üldine statistika tuli hea, 10 vastajat ehk 76,9% arvavad, et kasutajaliides on täielikult arusaadav, 3 ehk 23,1% aga vastasid, et mõned kohad jäid arusaamatuks (tabel 5).

Selle küsimuse kohta on ka tulnud isiklik tagasiside, miks üks vastajatest on pannud vastuseks, et mõni koht jäi arusaamatuks. Vastaja arvates on mängu välja juures olevad nupud küll intuiitselt ning arusaadavad, aga oleks hea, kui nende juures oleks väike teksti seletus või lihtsalt seletav tegusõna. Kuna see on tõesti koht, kus kasutajal võivad tulla arusaamatused nuppude tähenduse kohta, siis on võetud tagasisidet arvesse ja parandatud edaspidises töös.

Tabel 5. Mängu arusaadavus

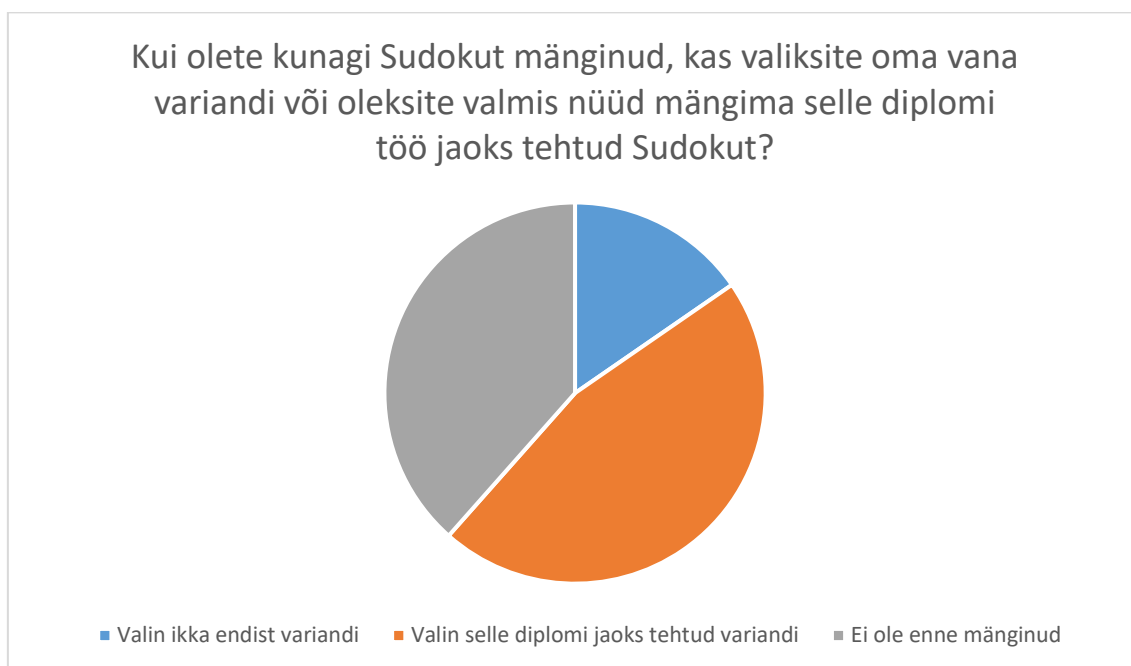


#### 4.1.4 Võrdlus teiste mängu rakendustega

Kuna mängus on olemas analoogseid lahendusi, on selles küsimuses vastajale pakutud võrrelda selle lõputöö jaoks tehtud mängu mänguga, mida nad on enne mänginud. Vastajatest 2 inimest ehk 15,4% valiks mängimiseks endist rakendust, mida nad on enne kasutanud, 5 inimest ehk 38,5% ei ole enne kunagi Sudokut mänginud ja 6 ehk 46,2% vastasid, et nad kasutaks seda rakendust (tabel 6).

Isiklikutest tagasisidetest on tulnud, et üks vastajatest on mänginud ainult LuxExpressi bussides ja tema arvates on selle töö jaoks tehtud lahendus mugavam, sest see ei lase teha viga, võrreldes bussis oleva rakendusega, mis ei näita viga, kui ruutu saab panna mitu varianti ja näidetakse viga ainult numbrite kokkulangevuse puhul. Teine vastaja mainis, et ta on proovinud laadida endale 5-6 erinevat rakendust, aga ükski ei sobinud ja selle töö jaoks tehtud rakendus tundub talle kõikidest parim.

Tabel 6. Rakenduse võrdlemine teistega



## 4.2 Teised tagasisided ja muutused

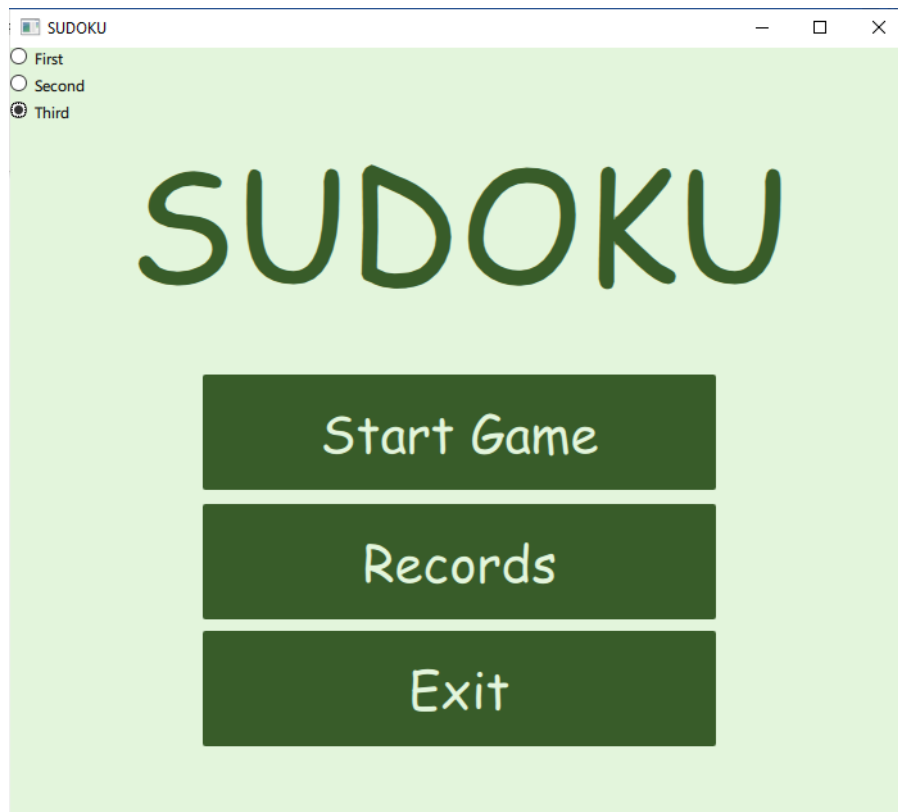
Peale uuringu läbiviimist tuli välja, et inimeste jaoks ei ole oluline, kas mängus on olemas värvivahetuse võimalus või ei ole, aga samuti tuli ka see, et inimestele ikka meeldib, kui mängudes selline võimalus olemas. Tuginedes sellele informatsioonile on otsustatud teha mängu uus võimaluse juurde, kus kasutaja saab valida kolme erineva värvi vahel, kas helesinise, roosaka või rohelise. Samuti esimene värviteema on muudetud rahulikumateks värvideks (näide pildidel 27, 28, 29).



Joonis 27. Helesinine kasutajaliides



Joonis 28. Roosa kasutajaliides



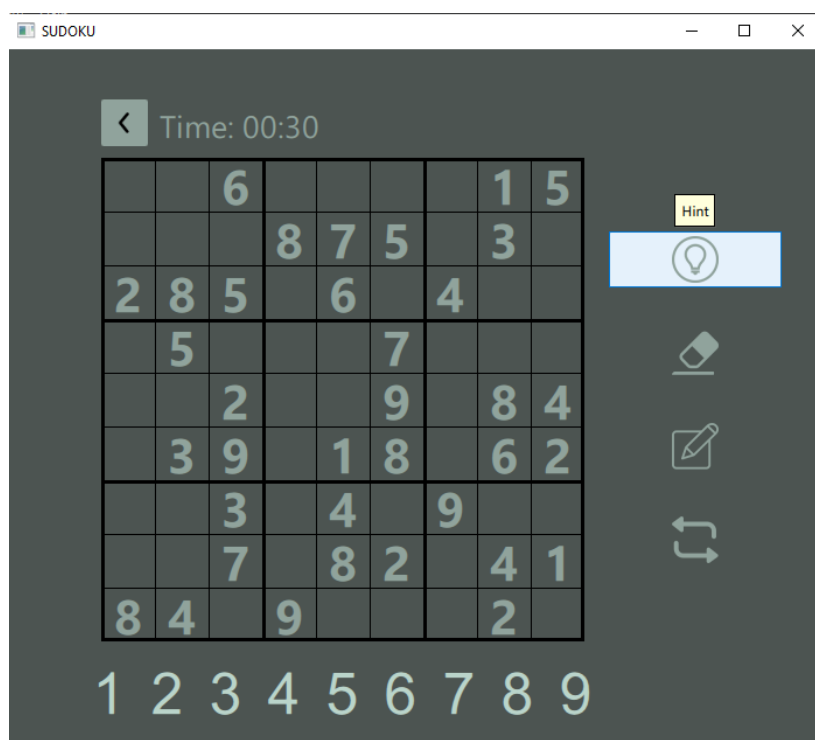
Joonis 29. Roheline kasutajaliides

Peale kommentaare, mis puudutasid küsitluse küsimusi on tulnud ka teist tagasisidet, mida võiks parandada. Üks nendest puudutas mängu akent. Enne küsitlust on olnud mängu väli üpris lähedal vasakule poolele ja sellele juhiti tähelepanu, et mängu väli võiks olla rohkem keskel. Selle soovitusel sai muudetud välja asukoht ja see otsus oli autori arvates õige, kuna nüüd see näeb välja loogilisem ja lakoonilisem. Samuti on olnud stardilehel erinevad teksti stiilid, millele samuti pöörati tähelepanu, see viga on ka nüüd parandatud (näide joonisel 30).



Joonis 30. Muudetud mänguvälja asukoht

Veel üheks oluliseks kommentaariks oli see, mis on juba enne mainitud, et inimestel võivad olla nupud arusaamatud. Isegi kui autori arvates on nad intuiitiivselt arusaadavad, ei pruugi aga teised aru saada, mida täpselt tähendavad nupud mängu ajal.



Joonis 31. Nupu seletus



Isegi arvestades seda, et kõike nuppu saab kasutaja ise läbi katsetada, on selle töö üks põhilistest eesmärkidest teha sellise rakendus, mis oleks lihtne ja arusaadav igale inimesele, kes kunagi soovib seda mängida. Selle kommentaari põhjal on mängu lisatud nupu seletused, kui nende peale hoida hiirt (näide joonisel 31).

## 5 Programmi arhitektuur ja algoritmi realiseerimine

Antud töös üheks eesmärgiks oli luua lahendus, kus kasutajaliides ja lahendusalgoritm on selgelt eristatavad. Kui keegi otsustab koodi muuta, kasvõi kirjutada see ümber teise algoritmina, ei saa kasutajaliides kannatada. Need osad on tehtud eraldi ja kindla võttega kokku ühendatud QtCreatoris. Terve selle töö kood asub git-is<sup>1</sup>.

### 5.1 Üldine struktuur

Kood koosneb neljast QML failist, kolmest C++ failist ja kolmest *headerist* ehk päisest. Failides `engine.cpp` ja `engine.h` on kirjutatud ja määratud kogu mängu loogika, Sudoku failist lugemine ja mängu välja genereerimine, käigu tegemine, kontroll, kas käik on legaalne või mitte, lahtrist vastuse kustutamine, terve mängu lahendus, vihje andmine ja paar funktsiooni, mis on tehtud selleks, et edaspidi oleks lihtne QML-i ja C++ ühendada.

Failid `sudoku.cpp` ja `sudoku.h` on QtCreatori pool automaatselt genereeritud failid, kuhu kirjutatakse funktsioone, mida edaspidi saab kasutada QML koodis. Peab olema mingi objekt, mille sees hoitakse kogu infot mängust, milleks on loodud klass `Sudoku`, kus tehakse objekt *game* ja registreeritakse see, et see oleks nähtav QML-i mootoris. Meetodi eelis on see, et tehakse üks objekt ja see saab olla ainult ühes eksemplaris. On olemas ka teised meetodid, kus objekt viiakse QML-i kontekstisse (mis on ilma asjata raske) või registreerida klass `Sudoku` QML mootori jaoks, et teha nii palju koopiaid kui vaja, aga kõike optimaalsem oli meetod ühe objektiga. Nimelt nendesse failidesse lisatakse ligipääsu mootorile kus on kõik mängu andmed ja luuakse mudel, millel peab olema ligipääs mängu mootorile, et oleks võimalik lugeda andmed, mis lähevad mängu väljasse kasutajaliideses.

Failides `sudokufield.cpp` ja `sudokufield.h` on samuti automaatselt genereeritud ja täidetud failid, mis vajasis lisakoodi. *SudokuField* ehk `Sudoku` väli on tabelikujuline mudel, ehk sellel on olemas read ja veerud. Selles failis on loodud atribuudid QML koodi jaoks, mis ei ole ette kirjutatud. Ehk mudeli jaoks ja sellega töötamise jaoks on vaja kirjutada need omadused, mis on vajalikud mudeli kasutamiseks. Selles programmis on loodud ainult

---

<sup>1</sup> <https://github.com/ksyyta/Sudoku>

üks atribuut, nimega *num*. Tavaliselt tabelitel on rohkem erinevaid veerge ja andmeid, kuigi Sudoku puhul on olemas ridades ja veergudes ainult numbrid. Funktsioonis *data* võetakse mängu mootorist funktsioonist informatsiooni numbrite ja nende koodinaatide kohta.

## 5.2 Mängu mootor

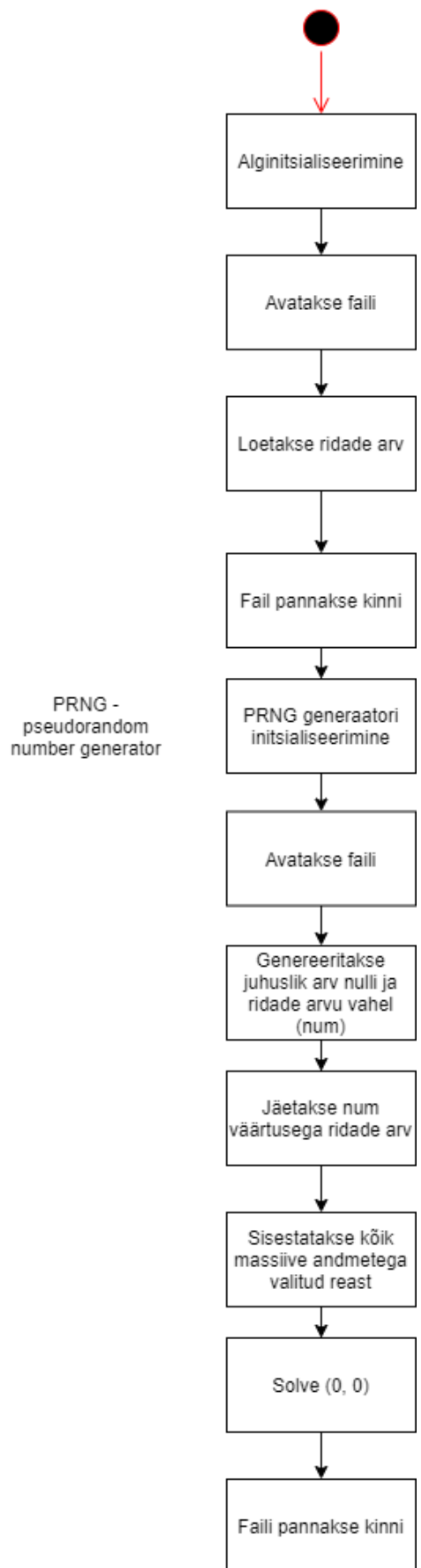
Mängu mootori osa koosneb seitsmest funktsioonist, mis kontrollivad kogu mänguprotsessi. Selleks päisi failis tehtud klassi nimega *Engine* ja defineeritud kõik olulised asjad:

```
class Engine
{
int Field[81];
int SolvedField[81];
int FreshField[81];
int ErrorCounter;
public:
Engine();
int Value(int x, int y);
bool MakeMove(int x, int y, int num);
bool Legal(int x, int y, int num);
void ClearCell(int x, int y);
bool Solve(int x, int y);
void StartGame();
void Print();
void Solve();
void Hint();
void Refresh();
};
#endif // ENGINE_H
```

Joonis 32. C++ Engine klass

### 5.2.1 *StartGame* funktsioon

*StartGame* ehk alusta mängu funktsioon on tehtud selleks, et alustada mängu. Seal pannakse paika näiteks vigade loendur, mis on mõeldud selleks, et kasutaja jaoks teha mängu raskemaks ja huvitavamaks, sest kolme veega mäng lõpeb. Programmi skeem joonisel 33.



Joonis 33. Programmi skeem

Peamine ülesanne on siin lugeda failist juba ette kirjutatud Sudokud ja laadida neid mängu väljadele. Antud töös on tehtud suvaline valik failis olevate Sudokude vahel [7]. Nullid massiivis tähendavad, et lahter on tühi. Alguses loetakse ridu, valitakse nendest üks ja siis sealt kirjutatakse rida arvu massiivi.

```
void Engine::StartGame()
{
    memset(Field, 0, 81); //massiivide nullimine
    memset(FreshField, 0, 81);
    memset(SolvedField, 0, 81);
    ErrorCounter = 0;

    int num = 0;
    int numLines = 0;

    std::ifstream inFile; //faili avamine
    std::string str;
    inFile.open("tables.txt");
    while (std::getline(inFile, str))//ridade arvutamine
        ++numLines;
    inFile.close();

    //juhuslikute arvude genereerimise mootor
    std::mt19937 gen(time(NULL));
    std::uniform_int_distribution<> distr(0, numLines);

    numLines = distr(gen);
    inFile.open("tables.txt");

    //jäetakse ridu vahele
    for (int i = 0; i < numLines; ++i)
        std::getline(inFile, str);

    //sisestatakse kõik massiive valitud Sudokudega
    for (int i = 0; i < 81; ++i) {
        inFile >> num;
        if (num >= 0 && num <= 9) {

            Field[i] = num;
            FreshField[i] = num;
            SolvedField[i] = num;
        }
    }
    Solve(0,0);

    inFile.close();
}
```

Joonis 34. Mängu alustamise funktsioon

### 5.2.2 *MakeMove*, *Legal* ja *ClearCell* funktsioonid

Siin mõlemad funktsioonid on tehtud selleks, et aru saada, kas käik saab olla tehtud ja kas kõik tingimused selleks on täidetud. *MakeMove* ehk tee käik on väga lühike funktsioon, *bool* tüüpi, mis tähendab, et ta tagastab kaks varianti, kui käik on legaalne tagastatakse tulemuseks on *true* ning kui käik ei ole legaalne, sest ei ole kõik reeglid täidetud, siis funktsioon tagastab *false*.

Funktsioonis *Legal* kontrollitakse Sudoku reegleid. Esiteks arv ei saa olla väiksem kui üks ja suurem kui üheksa, sest see mäng ei luba suuremaid väärtusi. Järgmisema kontrollitakse, kas samas reas ei ole juba sama arvu, kui on siis käik ei ole legaalne ja seda ei saa teha. Kolmas kontroll vastutab veeru kontrollimise üle, mis on sama põhimõttega nagu rea kontroll. Viimaseks on ühe ruudu üle vaatamine, kus samuti ei saa olla samasuguseid arve.

Samuti on olemas *ClearCell* ehk puhasta väli funktsioon, mis on tehtud kasutaja mugavuseks. Kui kasutaja on pannud vale vastuse, siis saab ta seda parandada, kustutades eelmise vastuse.

### 5.2.3 *Solve* funktsioon

Kõige olulisem funktsioon siin on *Solve* ehk lahenda, mis lahendab terve mängu ära. Algoritm ise on võrdlemisi lühike, seal vaadatakse, kas lahendaja ei lähe väljast välja, et see õigel ajal läheks teisele veerule ja kas selles kohas on juba olemas mingi number, kui jah, siis see koht jäetakse vahele. Kui kõik kontrollid on tehtud, siis tsükliga massiivi abil proovitakse arve ühest üheksani, kui nad on legaalsed, siis number pannakse sisse ja liigutakse massiivis järgmise koha juurde, kui ei ole, siis proovitakse edasi teise arvuga. *Solve* funktsioon proovib lahendada mängu rekursiivselt. Sudokut ise loetakse failist, kuhu on eelnevalt trükitud mõned mängu variandid, peale mida algoritm lahendab neid vastavalt sellele, milline arvude kombinatsioon on valitud.

```

bool Engine::Solve(int x, int y)
{
    if (x + y*9 >= 81) return true;//lõppu kontroll

    //kui veerg suurem kui 9, siis järgisele reale
    if (x >= 9){
        ++y;
        x = 0;
    }

    //rekursiivne funktsiooni kutsumine järgmise ruuduga
    if (SolvedField[x + y*9] > 0) return Solve(x+1, y);

    //Ruutu arvu proovimine
    for (int num = 1; num <= 9; ++num)
    {
        if (Legal(x, y, num)){

            SolvedField[x + y*9] = num;

            if(Solve(x+1, y)) return true;
        }

        SolvedField[x + y*9] = 0;
    }
    return false;
}

```

Joonis 35. Sudoku lahendamise algoritm [8]

#### 5.2.4 Value ja Hint funktsioonid

*Value* on abifunktsioon, mida kasutatakse *SudokuFieldis* ja see tagastab massiivi *Field* antud koordinaadil. *Hint* ehk vihje funktsioon aga tehtud kasutaja mugavuse jaoks ja võimaldab teha vihjeid, kui on raskusi lahendamisel. Selleks, et mäng ei oleks liiga lihtne, antakse vihjet juhuslikult ja kasutaja ei saa valida, kuhu täpselt tuleb vihje.

```

void Engine::Hint()
{
    int temp[81];
    int count = 0;

    //ajutiselt salvestab tühjade ruutude koordinaate
    massiivi
    for (size_t i = 0; i < SIZE(Field); ++i) {
        if (Field[i]) continue;
        temp [count ++] = i;
    }

    std::mt19937 gen(time(NULL));
    std::uniform_int_distribution<> distr(0, count - 1);

    int square_num = temp [distr(gen)];
    Field[square_num] = SolvedField[square_num]
}

```

Joonis 36. Vihje andmise funktsioon



## 6 Kokkuvõte

Selle töö eesmärgiks oli luua töötav rakendus, millel oleks mugav ja arusaadav kasutajaliides, mis oleks kirjutatud C++ ja QML-i abil ja samuti teha platvormi sõltuvamatuse analüüs ning kasutada vastavat platvormi. Rakenduse põhimõte oli teha töötav Sudoku mäng, kus oleks võimalus kasutajal ise mängida, saada vihjeid ja omada võimalust, et programm lahendaks ka isemängu. Kood töö jaoks oli kirjutatud plaanipäraselt Qt Creatoris.

Töö käigus on uuritud üldised Sudoku reeglid, mängu lahendamise võimalused ja teooria, ja koodi kirjutamiseks erinevad võimalikud algoritmid. Valmis rakendus kasutab tagurdamise algoritmi, mis võimaldab lahendada ka kõige raskemaid mängu, kui nendel on olemas reaalne lahendus. Samuti on loodud kasutajaliides ja tehtud uuring 13 inimeste seast, kus selgusid esialgses variandis tekkinud mitmed vead, nagu erinevad teksti stiilid ja kujundite paigaldus, mis on lõplikus variandis parandatud. Uuringu käigus tuli välja, et kasutajale ei ole oluline kasutajaliidese värvi valiku võimalus, samas aga kõikidele vastajatele meeldiks selline võimalus. Sellepärast on hiljem mängu oli lisatud värvi valiku võimalus.

Lõpptulemus on tänu uuringule ning analüüsile parem, kui oli oodatud, sest lisaks eesmärkidele on ka teisi asju rakendusse toodud. Kõik töös olevad eesmärgid on täidetud ja programm töötab korralikult ja täidab kõike oma ülesandeid korrektselt. Edaspidiseks jätkamiseks selle programmiga võiks proovida teisi algoritme, mis on uuritud teooria osas või viia üle rakendust telefonidele ja tahvelarvutitele.

## Kasutatud kirjandus

- [1] „Wikipedia,“ 14 aprill 2021. [Võrgumaterjal]. Available: <https://en.wikipedia.org/wiki/Sudoku>. [Kasutatud 7 mai 2021].
- [2] p53, „Habr,“ 19 september 2013. [Võrgumaterjal]. Available: <https://habr.com/ru/post/194410/>. [Kasutatud 3 märts 2021].
- [3] „Sudoku of the day,“ [Võrgumaterjal]. Available: <https://www.sudokuoftheday.com/>. [Kasutatud 18 märts 2021].
- [4] „Qt,“ The Qt Company, 2020. [Võrgumaterjal]. Available: [qt.io](https://qt.io). [Kasutatud 18 märts 2021].
- [5] „Qt,“ The Qt Company, 2021. [Võrgumaterjal]. Available: <https://doc.qt.io/qt-5/qml-qtquick-rectangle.html>. [Kasutatud 1 mai 2021].
- [6] „Adobe Color,“ Adobe, 2021. [Võrgumaterjal]. Available: <https://color.adobe.com/ru/create/color-wheel>. [Kasutatud 15 aprill 2021].
- [7] „Sudoku,“ *Sudoku*, p. 62, 2018.
- [8] „GeeksForGeeks,“ GeeksForGeeks, 24 veebruar 2021. [Võrgumaterjal]. Available: <https://www.geeksforgeeks.org/sudoku-backtracking-7/>. [Kasutatud 4 aprill 2021].

# Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks<sup>1</sup>

Mina, Ksenija Ivanova

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose „Sudoku lahendaja“, mille juhendaja on Peeter Ellervee
  - 1.1. reprodutseerimiseks lõputöö säilitamise ja elektroonse avaldamise eesmärgil, sh Tallinna Tehnikaülikooli raamatukogu digikogusse lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;
  - 1.2. üldsusele kättesaadavaks tegemiseks Tallinna Tehnikaülikooli veebikeskkonna kaudu, sealhulgas Tallinna Tehnikaülikooli raamatukogu digikogu kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.
2. Olen teadlik, et käesoleva lihtlitsentsi punktis 1 nimetatud õigused jäävad alles ka autorile.
3. Kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest ning muudest õigusaktidest tulenevaid õigusi.

07.05.21

---

<sup>1</sup> Lihtlitsents ei kehti juurdepääsupiirangu kehtivuse ajal vastavalt üliõpilase taotlusele lõputööle juurdepääsupiirangu kehtestamiseks, mis on allkirjastatud teaduskonna dekaani poolt, välja arvatud ülikooli õigus lõputööd reprodutseerida üksnes säilitamise eesmärgil. Kui lõputöö on loonud kaks või enam isikut oma ühise loomingu tegevusega ning lõputöö kaas- või ühisautor(id) ei ole andnud lõputööd kaitsvale üliõpilasele kindlaksmääratud tähtajaks nõusolekut lõputöö reprodutseerimiseks ja avalikustamiseks vastavalt lihtlitsentsi punktidele 1.1. ja 1.2, siis lihtlitsents nimetatud tähtaja jooksul ei kehti.

## Lisa 2 – Tagasiside küsimustik

Kas teile meeldib mängu üldine välimus?

- Jah
- Ei

Kas te peate oluliseks, et mängus saaks vahetada kasutajaliidese värve?

- Jah, see on oluline
- Minu jaoks see ei ole tähtis

Kas teile meeldib, et mängus on võimalus kasutajaliidese värve valida?

- Jah
- Ei
- Vahet pole

Kas kasutajaliides tundub teile mugav ja loogiliselt arusaadav?

- Jah, on arusaadav
- Mõned kohad jäid arusaamatuks
- Ei saanud üldse midagi aru

Kui olete kunagi Sudokut mänginud, kas valiksite oma vana variandi või oleksite valmis nüüd mängima selle diplomi töö jaoks tehtud Sudokut?

- Valin ikka endist variandi
- Valin selle diplomi jaoks tehtud variandi
- Ei ole enne mänginud