

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Leonid Maznitsa 206193IACB

SUDOKU LAHENDAJA JAVAS

Bakalaurusetöö

Juhendaja: Peeter Ellervee
PhD

Tallinn 2023

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: Leonid Maznitsa

15.05.2023

Annotatsioon

Käesoleva lõputöö eesmärk oli saada töötav tarkvararakendus, mis töötab Sudoku mänguga. Peamiseks ülesandeks on lahenduse leidmine lühikese aja jooksul, vigaste variantide näitamine, võimalike liikumisvariantide kuvamine.

Arenduse käigus saadi programmlahendus koos kasutajaliidesega, mis võimaldab lahendada kõik vajalikud ülesanded. Kasutati Java programmeerimiskeelt (versioon 17).

Lõputöö sisaldab Sudoku lühikest ajalugu ja reegleid.

Lõputöö on kirjutatud eest keeles ning sisaldab teksti 25 leheküljel, 5 peatükki, 30 joonist, 1 tabeli.

Abstract

Sudoku Solver using Java

The aim of this thesis was to obtain a software application working with Sudoku puzzle. The main task of the solution is to find the solution of the puzzle in a short time, indication of incorrect variants, displaying possible variants of the move.

In the process of development, it was obtained programme solution with user interface, which allows to solve all the required tasks. The programming language used was Java (version 17).

The thesis contains a brief history and rules of Sudoku.

The thesis is in Estonian language and contains 25 pages of text, 5 chapters, 30 figures, 1 table.

Lühendite ja mõistete sõnastik

Toggle Button	Nupp, millel on olekud ja mille vahel saab vahetada (antud juhul 2 olekut ON ja OFF);
Sudoku mänguväli	9x9 lahtri suurune väli, mis koosneb 9 ruuduosast 3x3 alamväljast, millel mäng toimub;
Ruuduosa (Square Subfield)	Osa Sudoku mänguväljast, mis on 3x3 lahtritest koosnev ruut;
Lahter	Mänguvälja minimaalne ühik, mis võib sisaldada numbrilisi väärtusi vahemikus 1 kuni 9 ja võib olla ka tühi;
UI	User Interface, kasutajaliides;
„Sudoku Dashboard”	Sudoku lahendaja programmi nimetus;
PNG	Portable Network Graphics, graafilise teabe salvestamise vorming;
LMB (Left Mouse Button)	Hiire vasakpoolne nupp;
RMB (Right Mouse Button)	Hiire parempoolne nupp;
Kasutaja	Huvitatud inimene, kes suhtleb rakendusega interaktiivselt;
Vihje	Täidetud lahter (kus juba asub mingi number);

Sisukord

1 Sissejuhatus.....	10
2 Sudoku mäng.....	11
2.1 Mängu ajalugu.....	11
2.2 Mängureeglid ja põhinõuded.....	12
2.3 Mängu raskused ja eripärad.....	13
3 Programmfunktsionaal.....	14
3.1 Java valiku selgitus.....	14
3.2 Start Screen.....	15
3.2.1 Sudoku Field.....	16
3.2.2 „Clear” nupp.....	17
3.2.3 „Load” ja „Save” nupud.....	17
3.2.4 „Indicate Variants” nupp.....	19
3.2.5 „Show errors” nupp.....	20
3.2.6 „Check Solvability” nupp.....	22
3.2.7 „Generate” nupp.....	23
3.2.8 „Try solve” nupp.....	24
3.2.9 „Save Image” nupp.....	25
3.2.10 „About” nupp.....	26
4 Programmi funktsionaalsed osad.....	27
4.1 Kasutajaliidese realiseerimine.....	27
4.2 Sudoku mänguvälja mudel klassis SudokuModel.....	28
4.3 Sudoku automaatse lahenduse algoritm.....	31
4.3.1 Ühe miljoni sudoku lahendamise test.....	33
4.3.2 Sudokud kõrge tasemega.....	33
5 Kokkuvõte.....	35
Kasutatud kirjandus.....	36
Lisa 1– Link projektile GitLab’is.....	37

Jooniste loetelu

Joonis 1. „Number Place” mäng (Dell Pencil Puzzles and Word Games 1979).....	12
Joonis 2. Algne sudoku 9x9 mänguväli (sektor on märgitud punasega, rida rohelisega ja veerg sinisega).....	12
Joonis 3. Lahendatud sudoku näidis (antud algsed väärtused on musta värviga, sisestatud- punasega).....	13
Joonis 4. „Sudoku Dashboard” algekraan.....	15
Joonis 5. „Sudoku Dashboard” rakenduse kasutusjuhud.....	16
Joonis 6. „Sudoku Dashboard” mänguväljaku täitmine.....	17
Joonis 7. „Sudoku Dashboard” mänguväljaku täitmine.....	17
Joonis 8. Kuidas salvestatud sudoku näeb .sudoku-failis.....	18
Joonis 9. Failid, mida meie saame avada programmis.....	18
Joonis 10. „Save” nupu vajutamisel tekkiv dialoog.....	19
Joonis 11. „Indicate Variants” nupu vajutamine tühjal mänguväljakul;.....	19
Joonis 12. „Indicate Variants” nupu vajutamine lihtsamal sudokul;.....	20
Joonis 13. „Indicate Variants” nupu vajutamine („The Hardest Sudoku);.....	20
Joonis 14. Sellel juhul, kõik väärtused on õigesti pandud;.....	21
Joonis 15. Sellel juhul, kui väärtused on valesti pandud, näidetakse, milles osas on viga.	22
Joonis 16. „Check Solvability” töö näide.....	23
Joonis 17. Võimalikud genereeritud sudoku tasemed.....	23
Joonis 18. Extremely Easy ja Evil sudoku näided.....	24
Joonis 19. „Try solve” nupu vajutamise tulemus (kui sudoku on õige).....	24
Joonis 20. „Try solve” tulemus, kui sudokul on mitu lahendust.....	25
Joonis 21. Kuidas „Save Image” nupp töötab.....	25
Joonis 22. Mida näidetakse „About” nupu vajutamisel.....	26
Joonis 23. „Sudoku Dashboard” rakenduse objektdiagramm.....	28
Joonis 24. Algne mänguväljaku täitmine arvudega.....	29
Joonis 25. Mänguväljaku transponeerimine.....	30

Joonis 26. Mänguväljaku veergude/reae vahetus ruudus.....	30
Joonis 27. Mänguväljaku ruudude vahetus.....	31
Joonis 28. Võimalikud genereeritud sudoku tasemed.....	33
Joonis 29. „Everest” sudoku – algne mänguväli ja lahendus.....	33
Joonis 30. „Gold Nugget” sudoku – algne mänguväli ja lahendus.....	34

Tabelite loetelu

Tabel 1. Programmi UI osad.....	16
---------------------------------	----

1 Sissejuhatus

Autor on 16 aastat malet mänginud, seega tema huvitub loogikamängudest. Sudoku on üks neist mängudest. Samuti võimaldas see töö autoril uurida graafiliste liideste ehitamist Java keele abil ja selle programmeerimiskeele uusi võimalusi.

Enne kui hakatakse rääkima töö tegemisest, tasub tähele panna seatud algtingimusi: Algoritm ja kasutajaliides Sudoku lahendamiseks, õigsuse kontrolliks ja vihjete andmiseks. Optimeerida lahendaja ja kontrollija töökiirust. Lähtetingimusteks on Sudoku reeglid.

Töö on jaotud mitmeks osadeks:

- Sudoku mängu kirjeldus (lühike ajalugu, mängureeglid, lahendustingimused);
- Sudoku mängu lahendamise meetodi kirjeldus;
- Saadud tarkvaralahenduse kasutusjuhend;
- Saadud tarkvaralahenduse sisemise toimimise kirjeldus;

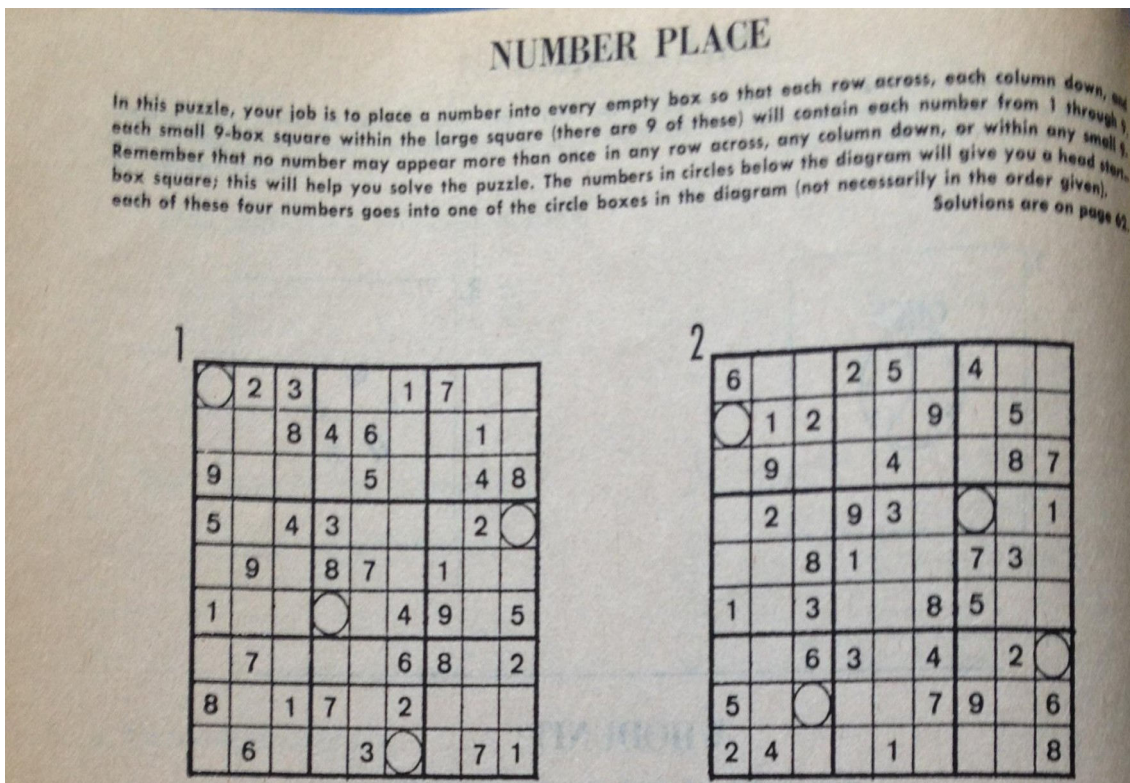
2 Sudoku mäng

Vaatame lühidalt, mida Sudoku mäng endast kujutab.

2.1 Mängu ajalugu

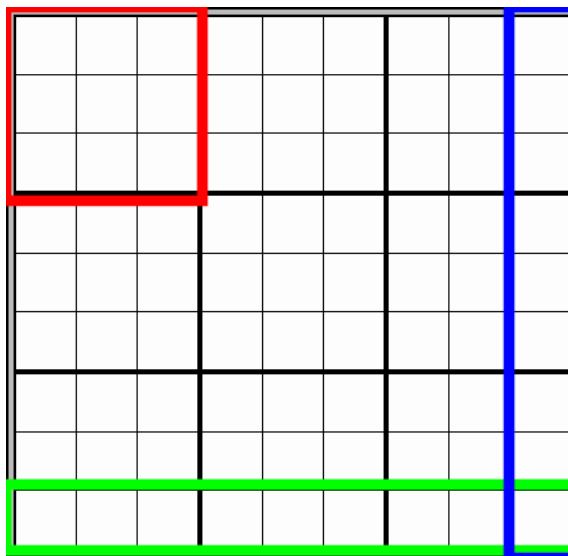
Arvatakse, et mäng sai alguse 18. sajandil, kui matemaatik Leonhard Euler leiutas matemaatilise mõistatuse "Ladina ruudud", mille käigus koostati ruudukujulised maatriksid, milles arvud ei tohi igas reas ja igas veerus korduda. Tema arhiividest leiti kirjeid, mis sisaldasid teavet "maagilise ruudu" kohta, millel oli teatud arv lahtreid (ta pööras erilist tähelepanu 9, 16, 25 ja 36 lahtritele). [1]

Kuni 20. sajandi teise poole keskpaigani ei olnud see mäng väga populaarne, kuid 1979. aastal avaldas ajakiri "Dell Pencil Puzzles and Word Games" mängu nimega "Number Place" (Joonis 1), mida meie teame tänapäeval "Sudoku" nime all. Mõned aastad hiljem rändas mäng Jaapanisse, kus see muutus äärmiselt populaarseks ja sai oma praeguse nime (Sudoku - jaapani keeles lühend "Sūji wa dokushin ni kagiru", mis tähendab, et "numbrid on piiratud ühe esinemisega. Paar aastat hiljem otsustasid jaapanlased lisada mängule uue elemendi: algvälja tuli täita mitte rohkem kui 32 lahtrit võimalikust 81-st, et lisada mängu raskuse element. [2]



Joonis 1. „Number Place” mäng (Dell Pencil Puzzles and Word Games 1979).

2.2 Mängureeglid ja põhinõuded



Joonis 2. Algne sudoku 9x9 mänguväli (sektor on märgitud punasega, rida rohelisega ja veerg sinisega).

Sudoku mängimiseks on vaja 9x9 ruudustikku (Sudoku on saadaval eri suurustes, näiteks 16x16). Väli on jagatud 9 ruuduosaks, millest igaühe mõõtmed on 3x3 lahtrit (Joonis 2).

Mõistatuse lahendamise tingimus on täita kõik lahtrid numbritega 1 kuni 9 järgmistel tingimustel (Joonis 3):

- iga number peab olema reas ainulaadne;
- iga number peab olema veerus unikaalne;
- iga number peab olema unikaalne selles ruuduosas, kuhu ta kuulub;

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

Joonis 3. Lahendatud sudoku näidis (antud algsed väärtused on musta värviga, sisestatud- punasega).

Peale nende reeglite peab korralikult koostatud Sudoku mõistatuses olema täpselt üks lahendus. Sageli peetakse mõistatuse raskusastmeks algsete vihjete arvu, milleks on algselt täidetud lahtrid. Dublini ülikooli üliõpilaste uuring näitas, et minimaalne täidetud lahtrite arv mängu alguses peaks olema vähemalt 17, et anda mõistatusele üks unikaalne lahendus. [3]

2.3 Mängu raskused ja eripärad

Sudokut võib nimetada NP-täielikuks probleemiks (NP-täielik probleem on probleem, mille lahendamiseks ei ole efektiivseid algoritme).

Hinnanguliselt on Sudokul suur hulk võimalikke lahendusi - see arv ulatub 6 670 903 752 021 072 936 960 kombinatsioonini (umbes 10^{21}).[3]

3 Programmifunktsionaal

Mängu töötlemiseks oli kirjutatud spetsiaalne graafilise kasutajaliidesega programm.

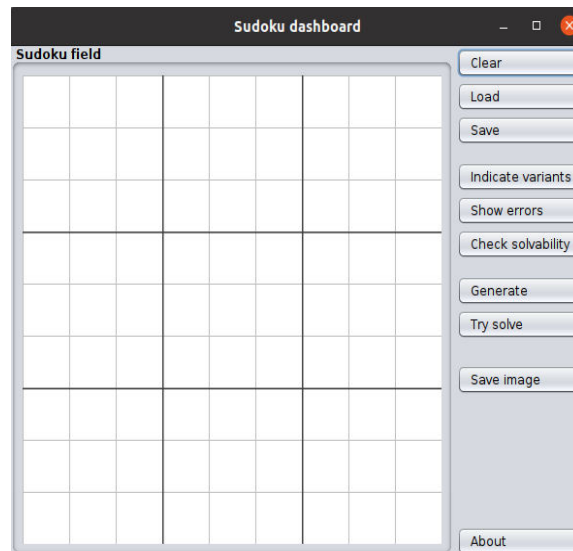
3.1 Java valiku selgitus

Java programmeerimiskeel valiti järgmistel põhjustel:

1. Java on perspektiivne programmeerimiskeel, millel on rikkalik põhiline API loogiliste ja matemaatiliste ülesannete lahendamiseks;
2. Java virtuaalmasin on platvormideülene (crossplatform) ja omab võimalusi peaaegu kõigi operatsioonisüsteemide jaoks;
3. Javal on palju graafilise kasutajaliidesefunktsioone, kasutades sisseehitatud Swingi raamatukogu;
4. Java-platvorm pakub turvalisemat mäluhaldust ja on sellega tegelemisel sallivam programmeerija vigade suhtes;

3.2 Start Screen

Pärast programmi käivitamist avaneb rakenduse graafiline kasutajaliides (Joonis 4).



Joonis 4. „Sudoku Dashboard” algekraan.

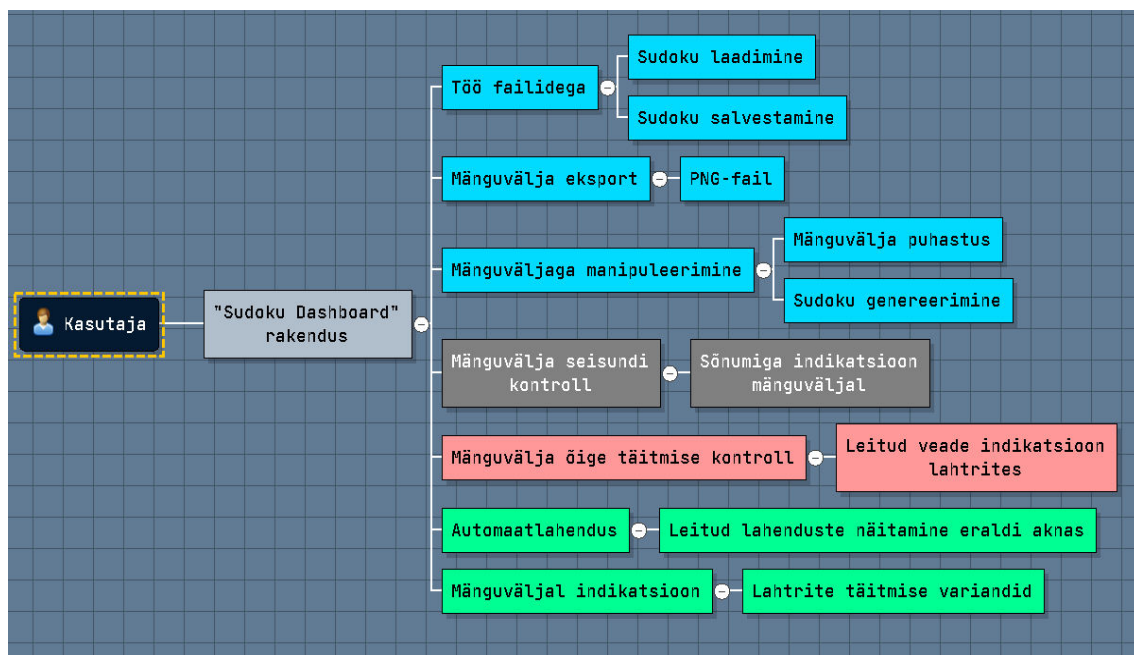
Kasutajaliides sisaldab järgmisi komponente:

Tabel 1. Programmi UI osad

Nimetus	Selgitus
Sudoku Field	9x9 lahtri suurune graafiline ekraan mänguvälja kohta;
„Clear” Button	Nupp, mis puhastab mänguvälja;
„Load” Button	Mänguvälja laadimine välisest .sudoku resolutsiooniga failist;
„Save” Button	Mänguvälja seisu salvestamine välisesse .sudoku resolutsiooniga faili;
„Indicate Variants” Button	Aktiveerib täitmata lahtrite võimaliku sisu kuvamise. Sõltuvalt võimalike variantide arvust valitakse, millise värviga lahter värvitakse;
„Show errors” Button	Mänguväljalt mõistatuste reeglite rikkumisi otsimine ja nende graafiline näitamine;

Nimetus	Selgitus
„Check Solvability” Button	Näitab mänguvälja praegust seisust;
„Generate” Button	Luuu sudoku mänguvälja algse arvu täidetud lahtritega;
„Try Solve” Button	Kontrollida, kas mänguvälja on õige- ja näidata leitud lahendusi eraldi dialoogis;
„Save Image” Button	Salvestada mänguvälja graafilist kujutist välisesse PNG-faili;
„About” Button	Väljastab lühiaandmeid (autori nimi, programmi pealkiri, kirjutamise aasta);

Joonisel 5 on välja toodud kasutusjuhtude (use-case) mõistuskaart, kus on näidetud võimalused kasutajale.



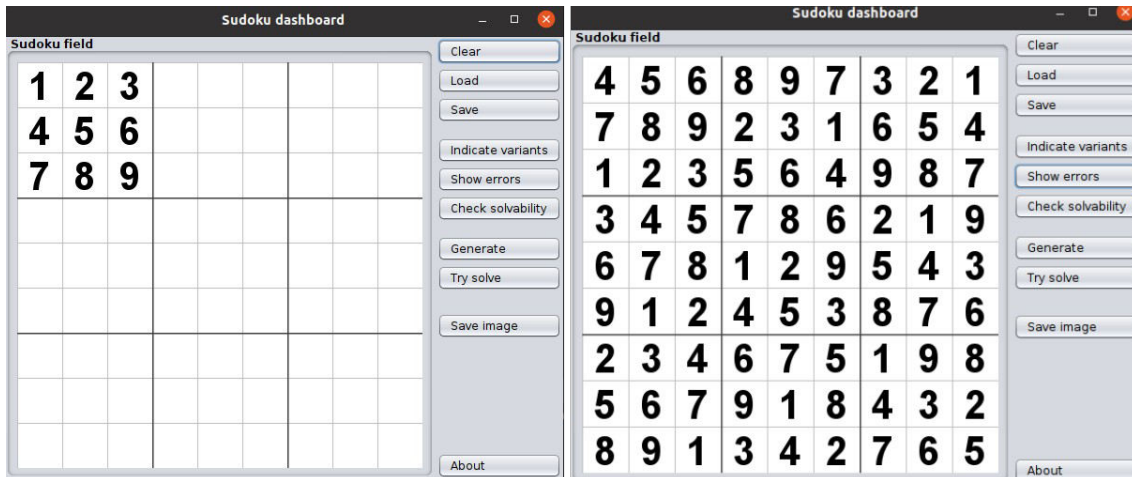
Joonis 5. „Sudoku Dashboard” rakenduse kasutusjuhud.

3.2.1 Sudoku Field

Peamine element kasutaja interaktsioonis on graafiline mänguväljak. Kasutaja saab seda juhtida hiirenuppudega (LKM ja PKM on toetatud). Kui kasutaja klõpsab mänguväljal, suureneb selle väärtus tsükliliselt 1 võrra. Muutuste vahemik on 0 kuni 9. Lahter, mille

väärtus on 0, kuvatakse tühjana. PCM teeb samu toiminguid, kuid vähendab mänguraku väärtust tsükliliselt 1 võrra.

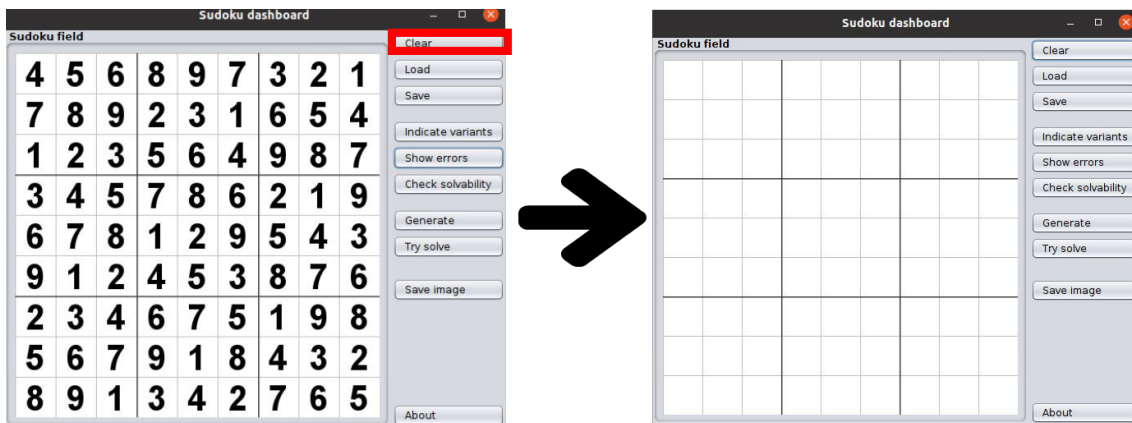
Joonisel 6 on näha kaks näidet sudokukastidest, mille täituvus on erinev:



Joonis 6. „Sudoku Dashboard” mänguväljaku täitmine.

3.2.2 „Clear” nupp

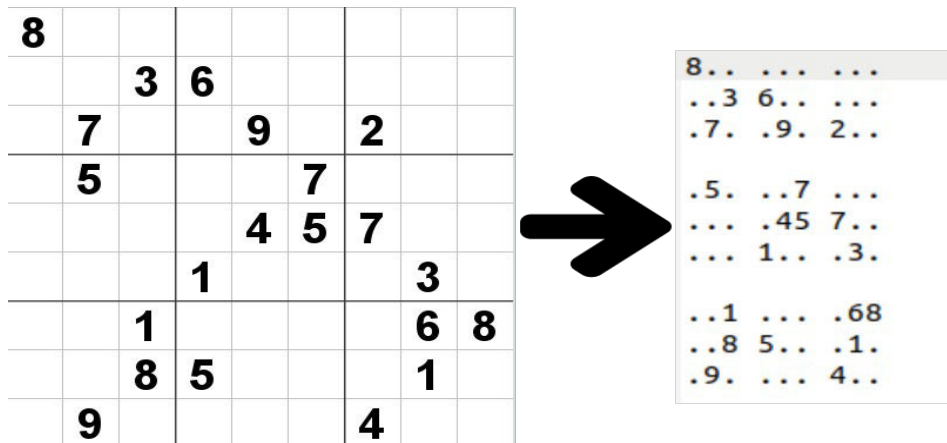
Selle nupu vajutamine puhastab mänguvälja täielikult (ilma võimalust seda operatsiooni tühistada) (Joonis 7).



Joonis 7. „Sudoku Dashboard” mänguväljaku täitmine.

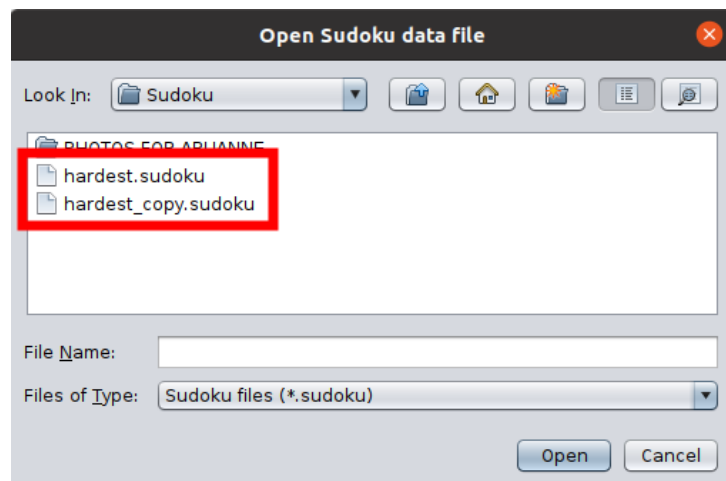
3.2.3 „Load” ja „Save” nupud

Rakendus võimaldab salvestada mänguvälja seisu välisesse tekstifaili laiendiga .sudoku (Joonis 8).



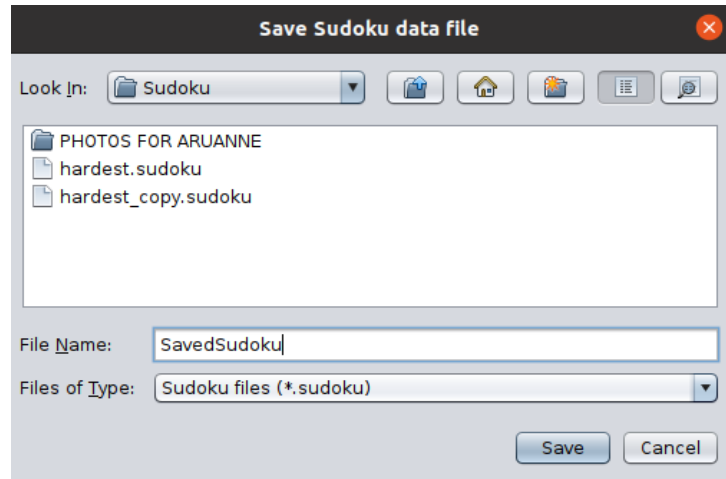
Joonis 8. Kuidas salvestatud sudoku näeb .sudoku-failis.

Nupp "Load" võimaldab aktiveerida dialoogi "Load from file" (Joonis 9).



Joonis 9. Failid, mida meie saame avada programmis.

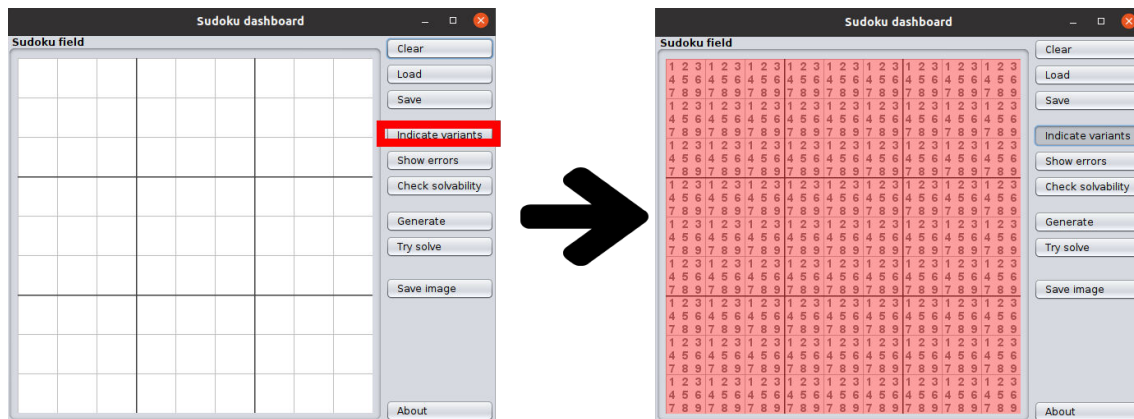
Nupp "Save" aktiveerib salvestusdialoogi (Joonis 10).



Joonis 10. „Save” nupu vajutamisel tekkiv dialoog.

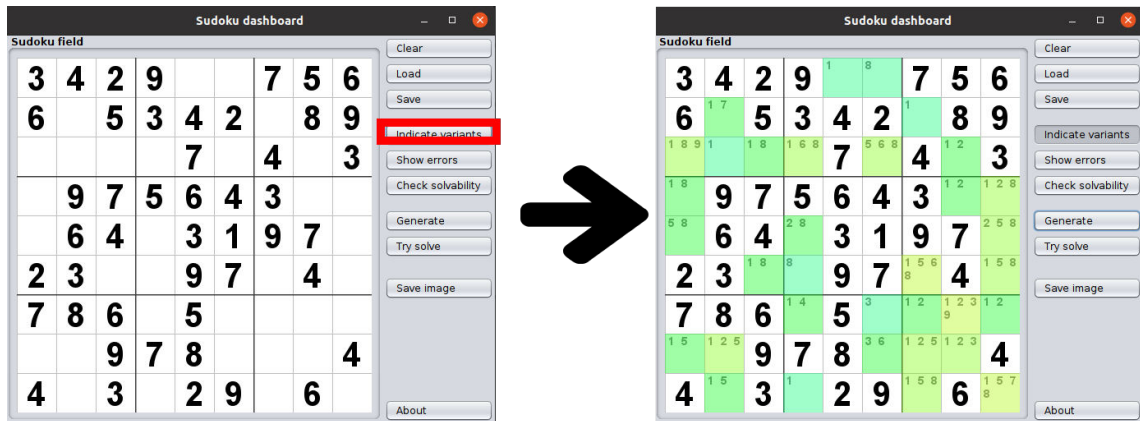
Töödeldakse ainult .sudoku laiendiga faile. Pärast allalaadimist sisaldab mänguvälja failis leitud teavet (numbrid vastavad numbritele, punktid vastavad tühjadele lahtritele).

3.2.4 „Indicate Variants” nupp

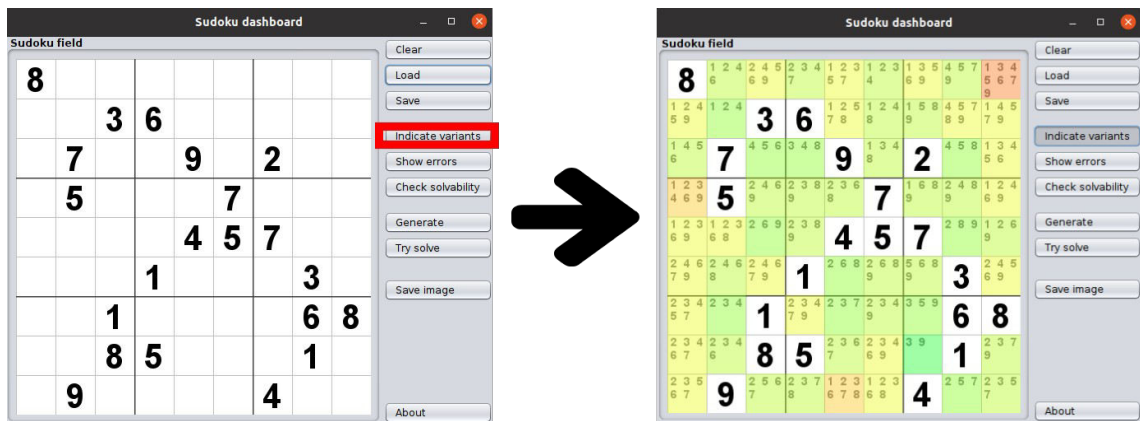


Joonis 11. „Indicate Variants” nupu vajutamine tühjal mänguväljakul;

Variantide kuvamine tähendab tühjade lahtrite arvvaartuste lubatud variantide näitamist. Sõltuvalt variantide arvust varieeruvad lahtrite värvid (rohelisest minimaalse arvu variantide puhul kuni punase maksimaalse arvu puhul). Variandid kuvatakse graafilise numbrite loeteluna lahtri sees (Joonis 11, Joonis 12 ja Joonis 13).



Joonis 12. „Indicate Variants” nupu vajutamine lihtsamal sudokul;

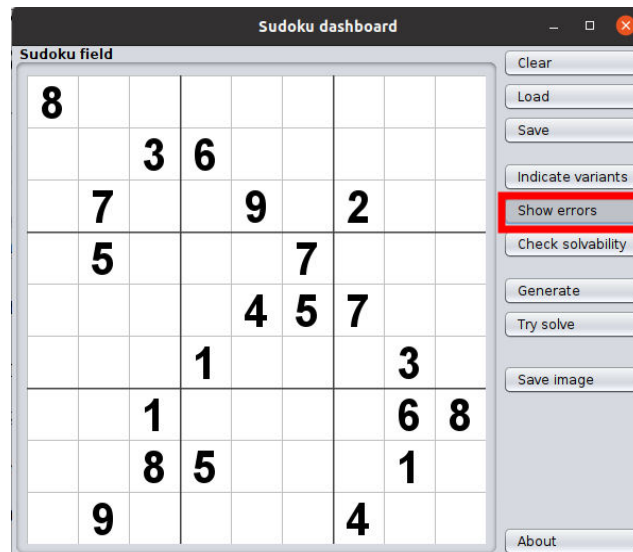


Joonis 13. „Indicate Variants” nupu vajutamine („The Hardest Sudoku”);

3.2.5 „Show errors” nupp

Selle valiku aktiveerimine põhjustab tarkvara pidevat mänguvälja analüüsi ja graafiliselt leitud vigade kuvamist. Kontroll toimub pärast iga kasutaja ja mänguvälja vahelist suhtlemist.

Kui vigu ei ole, siis ei ole mingit indikatsiooni (Joonis 14).



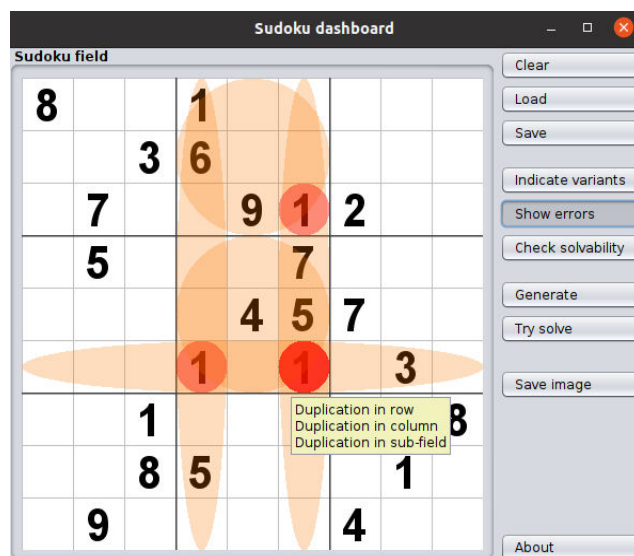
Joonis 14. Sellel juhul, kõik väärtused on õigesti pandud;

Kui leitakse vigu, kuvatakse graafiliselt probleemne lahter ja probleemne ala, et kasutaja saaks seda hinnata ja teha otsuse. Veakontroll tehakse pärast iga kasutaja ja mänguvälja vahelist suhtlemist.

Käsitletakse nelja tüüpi vigu:

- Duplication in row (kordumine reas). Programm leidis samas reas sama väärtusega lahtri;
- Duplication in column (kordumine veerus). Programm leidis samas veerus sama väärtusega lahtri;
- Duplication in sub-field (kordumine ruuduosas). Programm leidis samas ruuduosas sama väärtusega lahtri;
- There is no any variant for this cell (tühja lahtri jaoks puuduvad valikud). Antud varianti ei saa lahendada, sest ühe lahtri jaoks puuduvad valikud.

Joonisel 15 saab näha, näidet, kuidas see veade näitamine näeb välja.



Joonis 15. Sellel juhul, kui väärtused on valesti pandud, näidetakse, milles osas on viga.

3.2.6 „Check Solvability” nupp

Selle nupu vajutamine aktiveerib mänguvälja lahendatavuse kontrolli ja toimub tekstibänneri graafilise kuvamise mänguväljakul. Mänguvälja analüüsitakse vigade ja võimalike lahenduste arvu suhtes. Teave kuvatakse tekstisõnumina otse mänguväljakul.

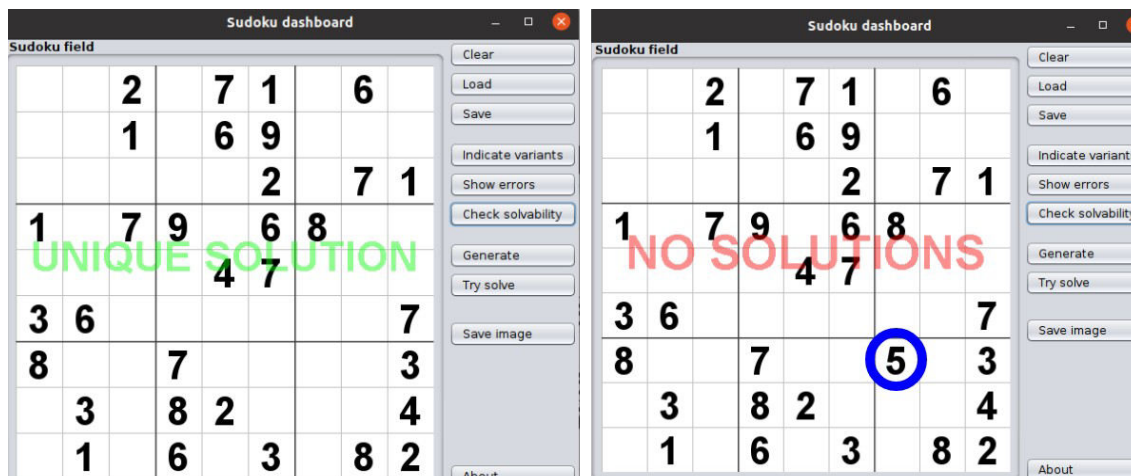
Pärast iga kasutaja mänguväljaga suhtlemist kaob tekstisõnum ära - ja selle uuesti kuvamiseks tuleb nuppu uuesti vajutada.

Võimalikud tekstisõnumid:

- Solved (Sudoku on lahendatud ja selles ei ole ühtegi täitmata lahtrit);
- Unique Solution (mänguvälja praegune olukord on õige ja sellel on ainult üks lahendus);

- Multiple Solutions (praegusele mänguvälja olukorrale on leitud rohkem kui üks lahendus);
- No Solutions (mänguvälja praegusele olukorrale ei ole leitud lahendust);
- Has Errors (mänguvälja täitmisel on vigu);

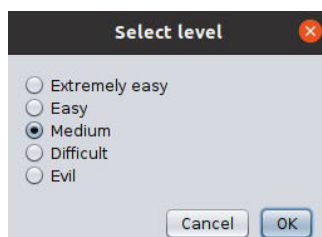
Paar näidet saab näha Joonisel 16.



Joonis 16. „Check Solvability” töö näide.

3.2.7 „Generate” nupp

Mõnikord on vaja automaatselt genereerida teatava raskusastmega mänguvälja. Kui see valik on aktiveeritud, kuvatakse raskusastme valimise dialoog (Joonis 17). Raskusena valitakse algselt täidetud lahtrite arv. Kui raskusaste suureneb, väheneb täidetud lahtrite arv.



Joonis 17. Võimalikud genereeritud sudoku tasemed.

Sõltuvalt raskusastmest valiti järgmine arv vihjeid (Joonis 18):

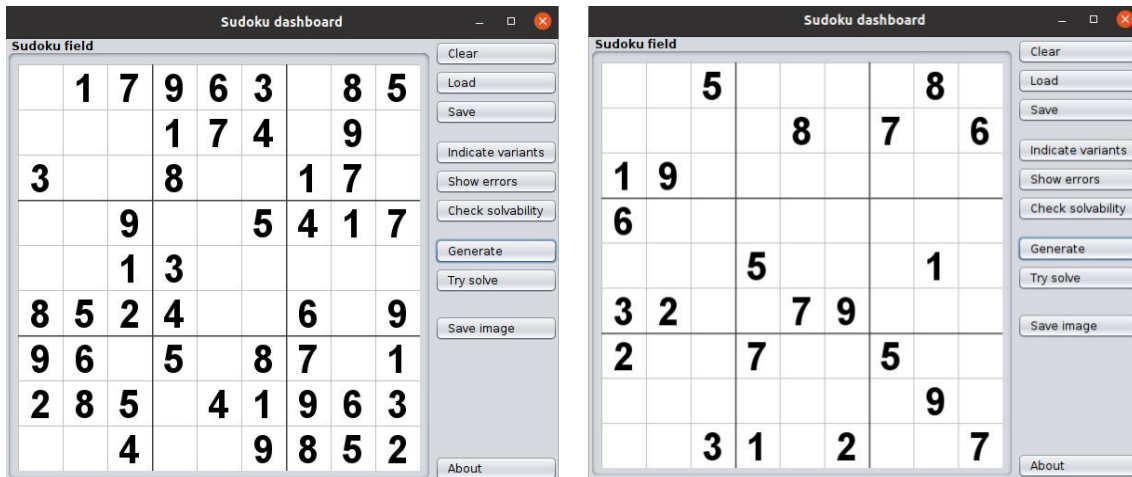
Extremely Easy – 47;

Easy – 36;

Medium – 32;

Difficult – 28;

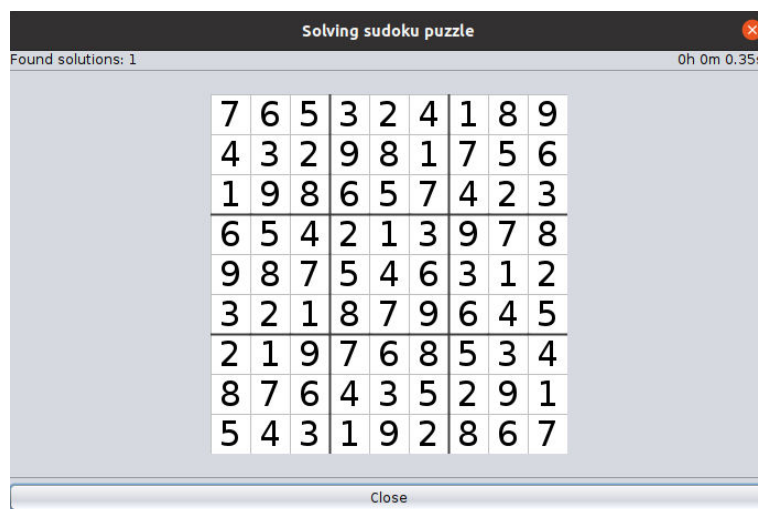
Evil – 22;



Joonis 18. Extremely Easy ja Evil sudoku näided.

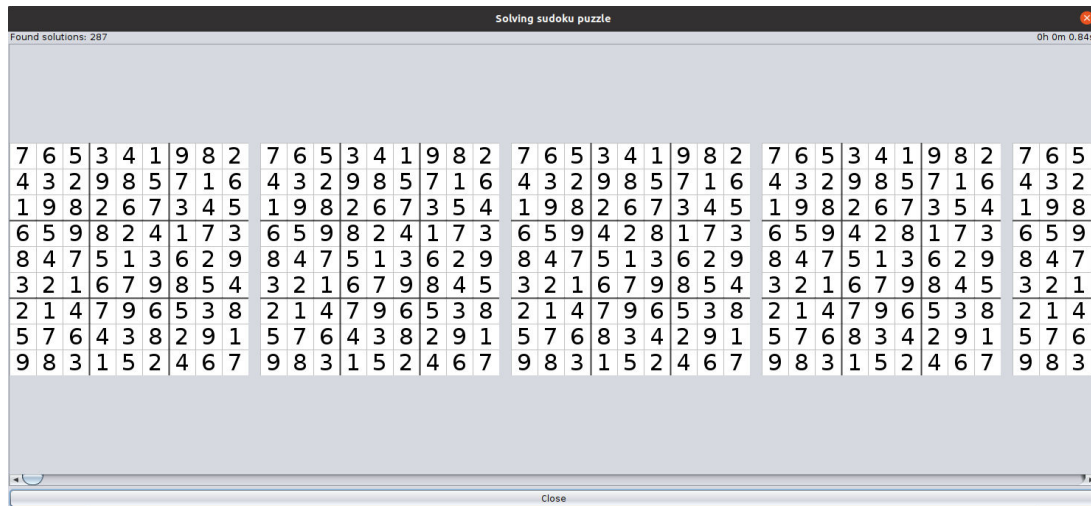
3.2.8 „Try solve” nupp

Kasutaja saab aktiveerida automaatse mõistatuste otsimise dialoogi. Sel juhul kontrollitakse väljal vigade olemasolu - ja kui vigu ei ole ja täidetud lahtrite arv on 17 või rohkem, avaneb automaatse lahenduse otsimise dialoog (Joonis 19).



Joonis 19. „Try solve” nupu vajutamise tulemus (kui sudoku on õige).

Dialogi ülaosas on teave kulunud aja kohta (silt paremal) ja leitud lahenduste arvu kohta (tekst vasakul). Mälu ülevoolu vältimiseks on lisatud lahenduste otsimise piirang. Leitud lahendused lisatakse dialoogi keskosas piltidena (Joonis 20).



Joonis 20. „Try solve” tulemus, kui sudokul on mitu lahendust.

3.2.9 „Save Image” nupp



Joonis 21. Kuidas „Save Image” nupp töötab.

Mõnikord on vaja salvestada mänguvälja visuaalne kujutis graafilise pildina. See valik aktiveerib dialoogiakna mänguvälja salvestamiseks PNG-pildina. Samuti sisaldab pilt kõiki hetkel aktiivseid märke (valikud, vead, lahendatavus) (Joonis 21).

3.2.10 „About” nupp

Kuvab rakendusteabe dialoogi (Joonis 22).



Joonis 22. Mida näidetakse „About” nupu vajutamisel.

4 Programmi funktsionaalsed osad

Programm koosneb kahest peamisest osast. Esimene osa on sudokuvälja mudel, teine osa on graafiline kasutajaliides.

Programm koosneb Java-klasside kogumitest, millest igaüks kuulub ühte kahest eespool kirjeldatud kategooriast.

4.1 Kasutajaliidese realiseerimine

Programm on kirjutatud Java keeles, kasutades Swingi raamatukogu. Programm algab klassiga Starter, mis sisaldab põhifunktsiooni ja loob rakenduse põhivormi. Rakenduse peamine aken on realiseeritud klassis "SudokuUiFrame". See klass laiendab standardset Java JFrame klassi ja seda kuvab operatsioonisüsteem vormina. Selle klassi sees loome visuaalseid komponente, millest üks on komponent, mis joonistab sudokuvälja, mis on realiseeritud klassis "SudokuUiField". Kõik teised komponendid vormil viitavad standardsetele Java objektidele nagu nupud ja paneelid.

Komponent "SudokuUiField" on rakendatud JPanel'i paneeliklassi alusel. Lahtrite kuvamiseks ja nende paigutuseks kasutatakse Swingi standardset layout-komponenti GridLayout suurusega 9 x 9 lahtrit. Igale lahtrile luuakse ja kirjutatakse SudokuUiCell tüüpi graafiline lapskomponent, mis vastutab sudoku lahtri sisu käsitleva teabe esitamise ja kasutajaga suhtlemise eest.

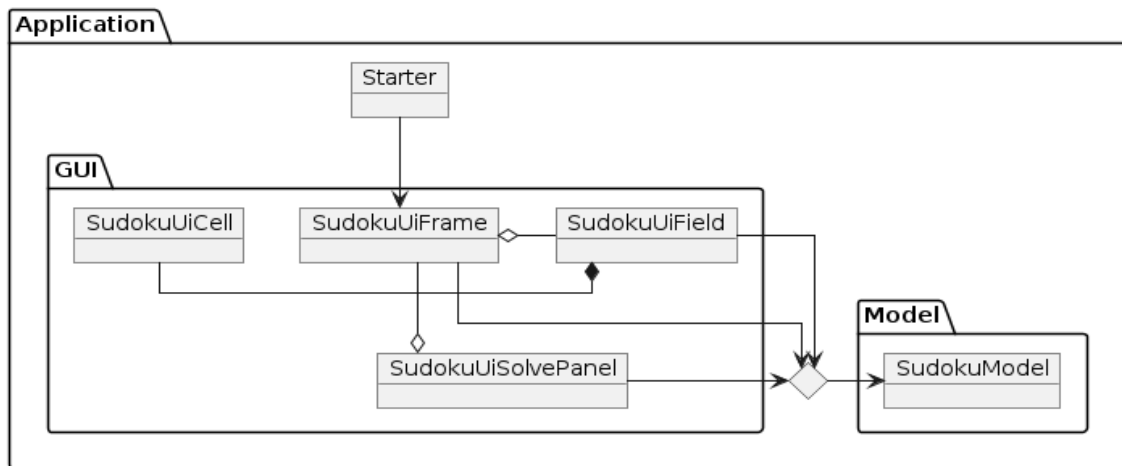
Komponendil "SudokuUiCell" ei ole tütarelemente ja see laiendab Java-klassi JComponent. See joonistab lahtri väärtuse, võib muuta värvi, näidata lahtri valikute arvu ja suhelda kasutajaga hiire ja klõpsude abil.

Lisaks kõigele sellele on rakendusel veel üks dialoogiboks, mida rakendab klass "SudokuUiSolvePanel". See on dialoogaken, mida kuvatakse automaatse sudoku otsingu ajal. See sisaldab kerivat paneeli leitud lahenduste piltidega, infosilte, mis näitavad leitud lahenduste arvu ja kulunud aega, ning „progress bar” (roheline riba

programmiakna allosas), mis kuvatakse lahenduste otsimise ajal ja peidetakse pärast töö lõppu.

Starter-klassis aktiveerime enne taotlusvormi loomist look-and-feel teema "Nimbus", et parandada rakenduse visuaalset tunnetust.

Lõplikku rakenduse objektidigrammi saab näha Joonisel 23.



Joonis 23. „Sudoku Dashboard” rakenduse objektidigramm.

4.2 Sudoku mänguvälja mudel klassis SudokuModel

Funktsionaalselt sisaldab see klass baidimassiivi, et salvestada teavet sudokuvälja oleku kohta. Sudoku-väli salvestatakse ühemõõtmelisse baidimassiivi. Seda tehakse selleks, et suurendada väljale salvestatud teabe töötlemise kiirust, samuti kasutatakse abifunktsioone, mis käsitsevad massiive (nt System.arrayCopy). Samuti sisaldab see klass meetodeid, mis manipuleerivad sudoku väljal olevaid andmeid, funktsiooni andmete genereerimiseks ja lahenduste otsimiseks. See klass on rakenduse „süda” ja seda katavad ühiktestid.

Peamiselt käsitleme seda klassi, kuna see on peamine lõputööga seotud lahendus. See klass on tehtud parameetriliseks, mis tähendab, et ta saab hakkama sudokuga, mille mõõtmed on ebastandardised (nt 16x16 lahtrit), kuid koodi on pandud piirang, et sudokuvälja külje maksimaalne suurus ei saa olla suurem kui 32 lahtrit. Et testida jõudlust mittestandardsete väljadega, on olemas üksuse test, mis lahendab 16x16

sudokut. Rakenduses kasutatakse seda mudelit ainult standardse sudoku lahendamiseks, mille alamvälja suurus on 3x3 lahtrit ja kolm alamvälja sudoku väljal mõlemal poolel (kokku 81 lahtrit).

Vaadake Sudoku väljade genereerimist SudokuModel klassi abil. Mänguväli genereeritakse SudokuModel#generate(int) meetodi kutsumisega. Argumendina antakse üle väljakule jäetavate vihjete arv (vihje- lahter, mille väärtus ei ole null). Välja genereerimine algab kogu Sudoku välja täielikust täitmisest mittevastavate andmetega. See tähendab, et me täidame kõik lahtrid mittetühjade väärtustega - ja kasutades nihkeid täidame sudoku korrektsuse tingimuse (Joonis 24) [4] .

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

Joonis 24. Algne mänguväljaku täitmine arvudega.

Klass rakendab meetodeid, mis võimaldavad sudokuvälja andmetega manipuleerida, kahjustamata nende terviklikkust ja korrektsust.

1. Terve tabeli transponeerimine - veerud muutuvad ridadeks ja vastupidi (Joonis 25);

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

→

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

Joonis 25. Mänguväljaku transponeerimine.

2. Kahe rea või kahe veeru vahetamine samal ruudul (Joonis 26);

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

↕

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

↔

Joonis 26. Mänguväljaku veergude/reae vahetus ruudus.

3. Kahe ruutu vahetus horisontaalselt või kahe ruutu vahetus vertikaalselt (Joonis 27);

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

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

Joonis 27. Mänguväljaku ruudude vahetus.

Neid meetodeid kutsutakse juhuslikult (500000 korda), et saada enam-vähem juhuslikult täielikult täidetud ja kehtiv sudoku mänguväli.

Järgnevalt nullime juhuslikud lahtrid, kuid teeme seda allpool kirjeldatud viisil, et tagada üks unikaalne lahendus genereeritud sudoku'le.

1. samm: võtame kõik lahtrimassiivi nihked, mis ei ole tühjad (mille väärtus ei ole 0);
2. samm: kirjutame nulli juhuslikult valitud lahtrisse (millel on mittenullväärtus). Lahtri väärtus jäetakse meelde;
3. samm: proovime lahendada saadud sudoku mänguvälja. Kui on ainult üks lahendus ja mittenullist lahtrid on rohkem kui antud vihjeid, minge sammu 1 juurde. Kui oleme jõudnud antud vihjete arvuni, lõpetame välja genereerimist;
 - 3.1. samm: kui sudoku lahendus ei anna ühte unikaalset lahendust, siis taastame lähtestatud lahtri väärtust ja läheme 1. sammule;

4.3 Sudoku automaatse lahenduse algoritm

Lahenduse algoritmidest on palju variante (nt X-wing, Naked Subset, Hidden Subset jne.), kuid mina olen valinud lihtsama rekursiivse sügavusotsingu algoritmi tagasitulekuga. Tuleb meeles pidada, et meie töötame piiratud andmekogumiga (tavalises Sudokus on ainult 81 lahtrit ja 9 võimalikku väärtust iga lahtri jaoks) - ja see

võimaldab meil vältida mälu puuduse või stack'i ületäitmist ohtu otsingu ajal (eriti praegustes arvutites).

Automaatne lahenduse otsimine toimub meetodi `SudokuModel#solve(int, Consumer<SudokuModel>)` abil. Argumendid tähendavad: esimene on soovitud maksimaalne piirang leitud lahenduste arvule ja teine annab üle objekti, mis saab infot iga leitud lahenduse kohta. Kasutame seda leitud lahenduste kuvamiseks lahenduste paneelil.

Kirjeldame lahenduse leidmise algoritmi:

1. samm: otsime välja kõik tühjad lahtrid ja kogume infot kehtivate väärtuste kohta, mida need võivad võtta.

2. samm: sorteerime leitud lahtrid vähemate ja rohkemate valikute vahel. Kui lahtrites on sama palju valikuid, võrdleme nende ruudude täituvust ja valime seda lahtri, mille ruut on kõige rohkem täidetud;

3. samm: võtame meie sorteeritud nimekirja esimese lahtri. Tõmbame selle esimese kehtiv väärtus välja ja paneme seda lahtrisse.

4. samm: kontrollime, et puuduvad tühjad lahtrid ja et väli on õigesti täidetud. Kui tühje lahtrid ei ole ja väli on korrektselt täidetud, tagastame leitud lahendust uue `SudokuModel` objektina ja lõpetame funktsiooni.

4.1. samm: kui lahtrid on tühjad, läheme sammu 1 juurde ja rekursiivselt kordame kogu operatsiooni;

Igal iteratsioonil, kui lahendus on saadud, kontrollime, kas lubatud arv lahendusi on saavutatud - ja lõpetame töötlemise, kui see on saavutatud.

Selle algoritmi keerukus on minu hinnangul $O(N^2)$. Minu koduarvutis Intel® Core™ i5-8250U 1,60 GHz protsessoriga tehtud testid näitasid, et selle algoritmi jaoks kulub ühe Sudoku mõistatuse lahendamiseks keskmiselt 400 mikrosekundit kuni 500 millisekundit. Ühiktestides on esitatud test 1 miljoni Sudoku lahenduse kohta Kaggle'i andmebaasist.

4.3.1 Ühe miljoni sudoku lahendamise test

Selleks, et kontrollida, kui kiiresti programm Sudoku lahendab, tehti mitu testi. Üks neist oli Sudoku lahendamine Kaggle'i andmebaasist, mis sisaldab 1000000 sudokut loetavas formaadis [5].

Lahenduse tulemusena näitas programm järgmisi tulemusi (Joonis 28):

```
[INFO] Running com.taltechleon.sudoku.KaggleSetTest
Loading Kaggle puzzle set...
Loaded 1000000 puzzle(s), spent time 947 ms
Start test, 1000000 records
.....
Completed, spent time 348272 ms, approx time per puzzle 0.348272 ms
```

Joonis 28. Võimalikud genereeritud sudoku tasemed.

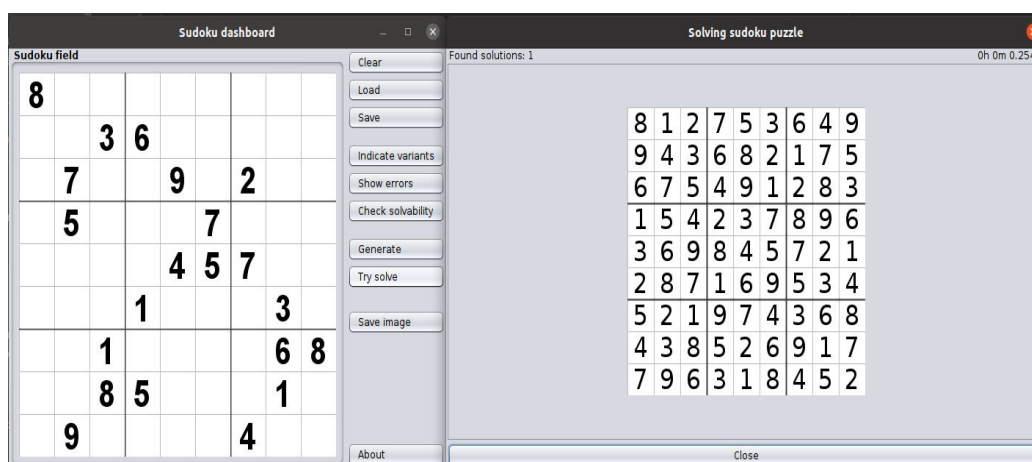
Kõigi variantide lahendamiseks kulus veidi üle 348 sekundi (peaaegu 6 minutit) ja keskmiselt 348 mikrosekundit ühe sudoku kohta.

4.3.2 Sudokud kõrge tasemega

Lisaks sudoku põhiosa lahendamisele huvitas mind ka see, kui kaua võtab aega kõige keerulisemate sudokuvariantide lahendamine.

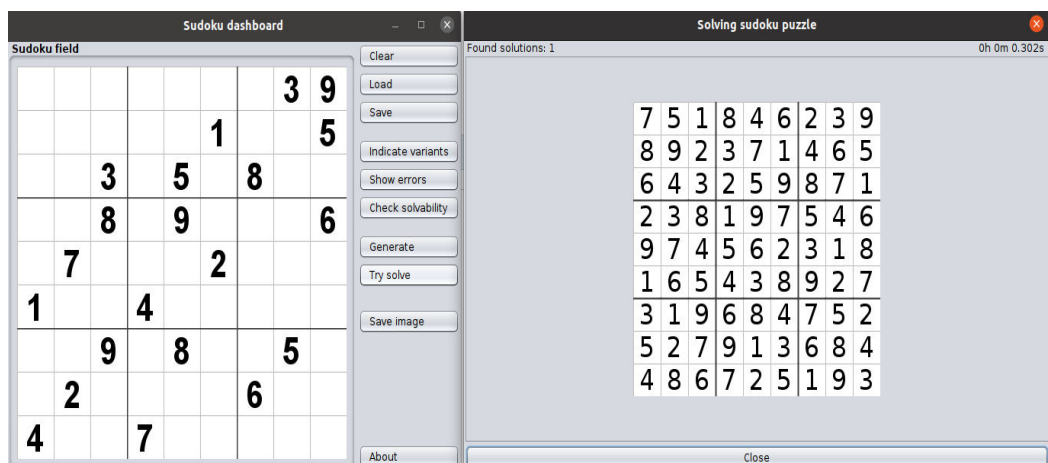
Esimene neist mõistatustest oli Everest, mille lõi 2012. aastal Soome matemaatik Arto Inkala, kes nimetas seda "maailma kõige keerulisemaks sudokuks".

Sisestades väärtused programmi ja klõpsates automaatsele lahendusele, sain vastuse 254 millisekundiga (Joonis 29).



Joonis 29. „Everest” sudoku – algne mänguväli ja lahendus.

Lisaks Everestile õnnestus autoril leida veel üks mõistatus (nii nimetatud "Gold Nugget"), mille lahendamine võttis veelgi kauem aega (Joonis 30) [7].



Joonis 30. „Gold Nugget” sudoku – algne mänguväli ja lahendus.

5 Kokkuvõte

Bakalaureusetöö tulemusena kirjutati rakendus, mis võimaldab kontrollida ja lahendada Sudoku mõistatusi (mõistliku aja jooksul), näidata võimalikke väärtusi lahtrites ja vigade olemasolu, kontrollida lahenduse unikaalsust - ja seda kõike kasutajasõbralikult tänu kasutajaliidesele. Rakendus töötab stabiilselt ja vigadeta - seega võib eesmärke pidada täidetuks.

Rakenduses kasutataval Sudoku algoritmil on võimalusi edasiseks optimeerimiseks, näiteks lahenduse otsimise paralleelistamine protsessori tuumade vahel ja töödeldavate variantide arvu vähendamine lahendamistehnikate rakendamise abil.

Lõputööd tehes omandasin olulised oskused programmide kirjutamisel Java keeles, kasutades Swingi raamatukogu ja programmeerimisvahendit Maven, samuti sain kogemusi uurimistöde koostamisel ja kirjutamisel.

Kasutatud kirjandus

- [1] “Sudoku ajalugu”, 2009. [võrgumaterjal] : <https://onsudoku.com/ru/history-sudoku/>
- [2] “The History of Sudoku”, 2018. [võrgumaterjal] : <https://sudoku.com/how-to-play/the-history-of-sudoku/>
- [3] “Mathematicians Solve Minimum Sudoku Problem”, 12.01.2012. [võrgumaterjal] : <https://www.technologyreview.com/2012/01/06/188520/mathematicians-solve-minimum-sudoku-problem/>
- [4] “Sudoku genereerimise algoritm”, 01.09.2013. [võrgumaterjal] : <https://habr.com/ru/articles/192102/>
- [5] “1 million Sudoku games”, 29.12.2016. [võrgumaterjal] : <https://www.kaggle.com/datasets/bryanpark/sudoku>
- [6] “Everest Sudoku: kõige raskem sudoku”, 05.02.2022. [võrgumaterjal] : https://www.youtube.com/watch?v=AxX2MljKZpM&ab_channel=Sudokuplus
- [7] “The Hardest Sudokus”, 09.01.2009. [võrgumaterjal] : <http://forum.enjoysudoku.com/the-hardest-sudokus-new-thread-t6539.html>

Lisa 1– Link projekte GitLab'is

Projekt „Sudoku Dashboard” asub GitLab'is ja link on järgmine:
<https://gitlab.cs.ttu.ee/lemazn/SudokuDashboard>