TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Ilja Karpenko 185675IADB

Platvormiülese rakenduse arendus Sudoku lahendamise tehnikate õppimiseks

Bakalaureusetöö

Juhendaja: Einar Kivisalu
MSc

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: Ilja Karpenko

15.05.2023

Annotatsioon

Antud bakalaureusetöö eesmärk on luua platvormiülene rakendus, mis töötab Windows ja Android platvormidel ja mille peamine eesmärk on aidata kasutajaid Sudoku lahendamise tehnikate õppimises. Töös on käsitletud olemasolevate materjalide ja rakenduste puudused ning selle analüüsi põhjal on koostatud nõuded loodavale rakendusele. Rakenduse loomise eesmärk on muuta Sudoku lahendamise õppimist mugavamaks ja huvitavamaks protsessiks.

Arenduse käigus luuakse rakendus, mis on võimeline lahendama ja genereerima Sudoku, andma kasutajale vihjeid ning mida saab seadistada kasutaja eelistuste järgi. Rakendus sisaldab materjale Sudoku lahendamise õppimiseks, mida toetavad interaktiivsed ülesanded, kus rakenduse kasutaja võib saadud teadmisi kohe kasutada.

Töö rõhk on kasutatavatel tehnoloogiatel, mis võimaldavad luua platvormiüleseid rakendusi ühel koodipõhjal, lihtsustades arendust ja võimaldades suuremal hulgal kasutajaid neid rakendusi kasutada. Töös on kirjeldatud Sudoku genereerimise ja lahendamise tööpõhimõtted, rakenduse arhitektuur, lokalisatsioon, kohanduv kasutajaliides ja seadistamise võimalused. Arenduse tulemuseks on töötav rakendus, mis täidab oma peamist eesmärki.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 42 leheküljel, 6 peatükki, 20 joonist, 1 tabelit.

Abstract

Development of Cross-Platform Application for Learning Sudoku Solving Techniques

The aim of this bachelor's thesis is to create a cross-platform application that works on Windows and Android platforms and whose main purpose is to help users learn Sudoku solving techniques. The paper discusses the shortcomings of existing materials and applications and based on this analysis, the requirements for the application to be created are drawn up. The purpose of creating the application is to make learning to solve Sudoku a more convenient and interesting process.

During development, an application is created that is able to solve and generate Sudoku, give hints to the user and can be configured according to the user's preferences. The application contains materials for learning how to solve Sudoku, supported by interactive tasks where the user of the application can apply the acquired knowledge immediately.

The emphasis of the work is on the technologies used, which allow creating cross-platform applications on a single code base, simplifying development and enabling a larger number of users to access these applications. The paper describes the working principles of Sudoku generation and solving, application architecture, localization, adaptive user interface and configuration options. The result of the development is a working application that fulfills its main purpose.

The thesis is in Estonian and contains 42 pages of text, 6 chapters, 20 figures, 1 table.

Lühendite ja mõistete sõnastik

API	*Application Programming Interface* ehk rakendusliides
ARB	*Application Resource Bundle* ehk rakenduse ressursipakett – failivorming, mis on sarnane JSON-iga ja mida kasutatakse rakenduse lokaliseerimisel
CSLA	*Congnitively Stimulating Leisure Activity* ehk kognitiivselt stimuleeriv vabaajategevus
CSS	*Cascading Style Sheets* ehk kaskaadlaadistik
HTML	*HyperText Markup Language* ehk hüperteksti märgistuskeel
IDE	*Integrated development environment* – integreeritud arenduskeskkond
JSON	*JavaScript Object Notation* ehk andmevahetusvorming, mis põhineb JavaScript programmeerimiskeelel
kaaslased	ruudud, mis asuvad antud ruuduga samas ühikus; igal ruudul on 20 kaaslast
kandidaat	number, mida võib hetkel panna antud ruutu nii, et see ei riku reegleid
MB	megabait
plokk	3x3 ruutude maatriks Sudoku ruudustiku sees, ruudustik sisaldab 9 plokki
SDK	*Software Development Kit* ehk tarkvaraarenduskomplekt
WCAG	*Web Content Accessibility Guidelines* ehk veebi sisu juurdepääsetavussuunised
ühik	rea, veeru ja ploki ühine nimetus

Sisukord

Jooniste loetelu

Tabelite loetelu

1 Sissejuhatus

Sudoku on loogikamäng, mis kujutab endast 9x9 ruudustikku, mis on jagatud omakorda 3x3 regioonideks või plokideks, kus mõned ruudud on täidetud numbritega. Sudoku lahendamiseks tuleb täita vabad ruudud numbritega. Sudoku peetakse korrektselt lahendatuks, kui kõik ruudud on täidetud vastavalt reeglitele. Reeglid on järgmised: kõik read, veerud ja plokid peavad sisaldama numbreid 1-st 9-ni ning numbrid ei saa korduda üheski reas, veerus ja plokis.

Lihtsamaid Sudoku mõistatusi saab lahendada kasutades ainult reegleid, kuid keerukamate mõistatuste lahendamiseks tuleb rakendada loogikat, et teha järeldusi numbrite paiknemise kohta. Aastate jooksul on inimesed leidnud erinevaid loogikavõtteid ning kirjeldanud neid. Tavaliselt nimetatakse selliseid võtteid Sudoku lahendamise tehnikateks või starteegiateks. Mõned neist on lihtsad ja arusaadavad, teised on keerulisemad ning avastada neid ise võib olla raske. Internetis on olemas erinevad materjalid, kus need tehnikad on kirjeldatud ja mille järgi saab õppida, kuid tavaliselt on materjalidel olulised puudused ning nende järgi õppimine on ebamugav.

Käesoleva töö eesmärgiks on luua platvormiülene rakendus, mis aitab inimestel õppida Sudoku lahendamise tehnikaid. Rakendus sisaldab materjale, mis seletavad tehnikaid ja näidismõistatusi, kus inimene saab õpitud tehnikaid rakendada. Samuti oskab rakendus genereerida Sudoku mõistatusi, lahendada neid ja anda vihjeid nende lahendamiseks. Töö eesmärgiks on luua rakendus, mida on võimalik kasutada nii nutitelefonis (Android platvorm) kui ka arvutis (Windows).

2 Ülevaade probleemist

Sudoku lahendamine on kasulik erineva vanuse inimeste jaoks eri põhjustel. Sudoku kuulub kognitiivselt stimuleerivate vabaajategevuste ehk CSLA (*Congnitively Stimulating Leisure Activity*) hulka.

On teada, et Sudoku kui kognitiivselt stimuleeritav vabaajategevus aitab erinevalt aktiveerida eesajukoore piirkondi. Seetõttu saab Sudoku kasutada kognitiivses ravis, eriti eesajukoore jaoks [1].

Oli viidud läbi uuring, kus uuriti CSLA seost eakate inimeste kognitiivsete funktsioonidega. Uuringu esimene hüpotees väitis, et CSLA on positiivselt seotud järgneva kognitiivse toimimisega 2 aasta pärast. See hüpotees leidis täielikult kinnitust. Isegi pärast algse mälu, arvutusoskuse ja sujuvuse ning sotsiaalmajandusliku tausta, tervise, sotsiaalsete võrgustike muutujate ja riigi kontrollimist säilitas CSLA sagedus positiivse seose iga kognitiivse funktsiooni tulemustega, mis olid mõõdetud 2 aastat hiljem [2].

Veel üks uuring näitas Sudoku mõju tudengite võimekusele lahendada matemaatilisi harjutusi. Uuringu osalejad teatasid parematest tulemustest oma matemaatikatundides ja parematest probleemide lahendamise oskustest. Probleemide lahendamise võime komponendid hõlmavad töömälu, mustrituvastust, mitmetähenduslikkuse ja frustratsiooni taluvust, tähelepanuvõimet, külgmist mõtlemist ja paindlikkust probleemide lahendamisel [3].

Mõned inimesed, nagu Grace Yeoh, väidavad, et Sudoku aitas neil ravida või saada lahti nn matemaatilisest ärevusest või matemaatilisest traumast [4]. See on emotsionaalne seisund, kui inimene arvab (tavaliselt negatiivse kogemuse alusel), et ta ei ole võimeline lahendama matemaatilisi probleeme ning need emotsioonid ja mõtted tõesti ei anna temal lahendada probleeme.

Need on erinevad viisid, kuidas Sudoku võib teha inimese elu paremaks. Kuna Sudoku lahendamist peetakse vabaajategevuseks, peab see olema huvitav. Alguses piisab

inimesel lihtsamatest Sudoku mõistatustest ning lahendamistehnikatest, mida ta saab ise leida. Ajaga õpib inimene lahendama Sudoku kiiremini ning samad mõistatused muutuvad liiga kergeteks. Inimene hakkab lahendama raskemaid mõistatusi ning võib kohata neid, mida ta ei saa üldse lahendada nende võtetega, mida ta teab. Siis on inimesel kaks võimalust – kas jätta raskemad Sudoku vahele või õppida neid lahendama. Kui inimene valib esimest varianti, siis ta ei õpi midagi uut ja kasutab lahendamises ainult enda leitud tehnikaid. Selline lähenemine võib muuta Sudoku lahendamist korduvaks ja igavaks protsessiks. Kui inimene valib raskete Sudoku lahendamise õppimist, siis on temal vaja kuskil õppida erinevaid lahendamistehnikaid. Uute tehnikate õppimine aitab säilitada huvi Sudoku lahendamise vastu kauemaks ajaks, sest teades rohkem tehnikaid saab inimene lahendada ka raskemaid Sudoku mõistatusi ning ta saab seda teha mitmel viisil, kasutades erinevaid tehnikate kombinatsioone.

Sudoku lahendamiseks kasutatakse erinevaid loogikavõtteid. Need loogikavõtted ongi sisuliselt Sudoku lahendamise tehnikad. Internetis on olemas palju erinevaid materjale, kus on kirjeldatud Sudoku lahendamise tehnikad ning samuti on palju rakendusi, kus saab erinevaid Sudoku lahendada, aga tavaliselt on neil mõned puudused, mis muudavad Sudoku lahendamise õppimist ebamugavaks.

2.1 Olemasolevad materjalid

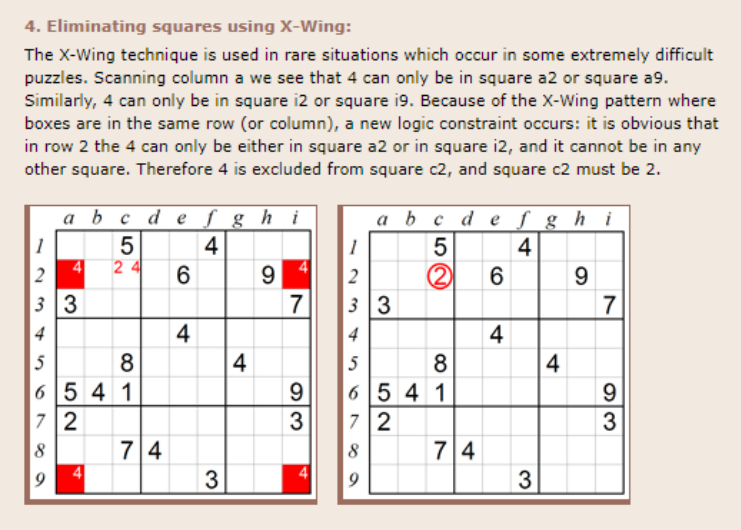
Kui inimene tahab õppida Sudoku lahendama, üks esimesi kohti, kuhu ta läheb vastusi otsima on Internet, seepärast on mõistlik saada ülevaadet Internetis olemasolevatest materjalidest. Tavaliselt, kui inimesed otsivad midagi Google’s, avavad nad kõige sagedamini esimesi otsingutulemusi ja ainult esimesel kolmel tulemusel on tõenäosus olla avatud rohkem kui 10% [5]. Seetõttu on mõistlik analüüsida ainult kolm esimest otsingu tulemust.

Google’s oli tehtud päring „*sudoku solving techniques*“ (sudoku lahendamise tehnikad) ning esimesed kolm tulemust olid järgmised:

- Sudoku techniques - Conceptis Puzzles [6],
- Sudoku Solving Techniques - Kristanix Games [7],
- Sudoku Rules - Strategies, solving techniques and tricks [8].

2.1.1 Conceptis Puzzles

Esimene ja seetõttu kõige sagedamini avatud tulemus on Conceptis Puzzles leheküljel paiknev artikkel. Seal on kirjeldatud mitu lahendamistehnikat, alustades kõige lihstamatest ja lõpetades X-Wing tehnikaga, mida peetakse edasijõudnud tehnikaks. Joonisel 1 võib näha X-Wing tehnika kirjeldust.



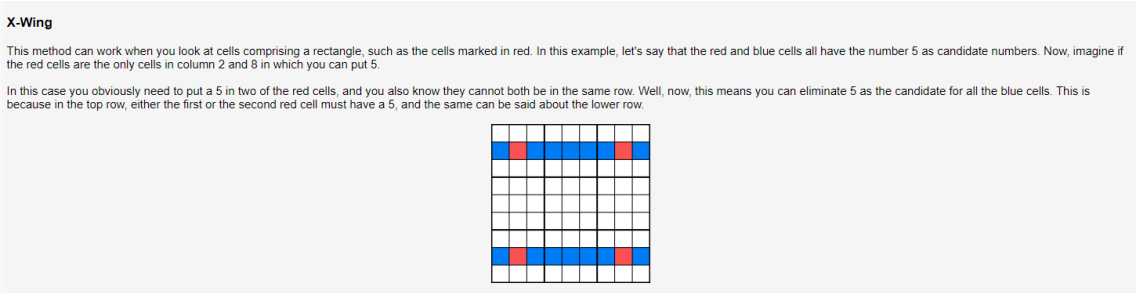
Joonis 1. X-Wing tehnika kirjeldus Conceptis Puzzles leheküljel

Kirjeldus sisaldab juhist, kuidas tehnikat rakendada, aga ei sisalda seletust, kuidas see töötab. Põhjendus on lihtne: „X-Wing mustri tõttu, kus ruudud on samas reas (või veerus), tekib uus loogiline piirang: ...“. X-Wing mustrit rakendatakse ühe näite lahendamiseks, aga ei seletata, miks X-Wing töötab. Inimene peab lihtsalt õppima pähe mustri ja kuidas seda rakendada. Kirjeldus sisaldab pilte, mis kergendab tehnikast arusaamist.

2.1.2 Kristanix Games

Teine tulemus on Kristanix lehekülg, seal on samuti kirjeldatud mitu tehnikat, nii lihtsad kui ka edasijõudnud, sealhulgas X-Wing.

Joonisel 2 võib näha X-Wing tehnika kirjeldust. Selles seletatakse, kuidas tehnika töötab ja tuuakse ühte näite. On olemas ka joonis, kuid seal ei ole numbreid ja seepärast tuleb näidet kujutada ette.

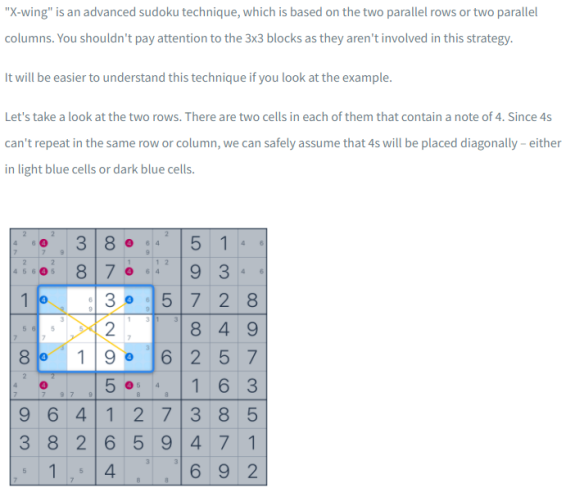


Joonis 2. X-Wing tehnika kirjeldus Kristanix leheküljel

2.1.3 Sudoku Rules

Kolmas tulemus on sudoku.com lehekülg, kus on toodud üsna palju tehnikaid, sealhulgas ka mitu edasijõudnuid. Näitlikkuse eesmärgil on mõistlik vaadelda samuti X-Wing tehnika kirjeldust.

Joonisel 3 võib näha eelmainitud tehnika kirjelduse osa – kirjeldatakse tehnika kasutamist ning loogikat selle taga, on olemas konkreetne näide piltidega. Puudustest võib esile tuua seda, et tehnika eesmärk, ehk mida sellega saavutatakse on toodud alles kirjelduse lõpus.



Joonis 3. X-Wing tehnika kirjeldus sudoku.com leheküljel

2.2 Olemasolevad mobiilrakendused

Android platovrmil on olemas suur hulk rakendusi Sudoku lahendamiseks. Selleks, et analüüsida nende sobivust Sudoku lahendamise õppimiseks, on mõistlik vaadelda olemasolevaid rakendusi ning analüüsida nende tugevaid külgi ja nõrkusi.

2.2.1 Populaarsed Sudoku rakendused

Sarnaselt materjalidega oli valitud 3 kõige populaarsemat Sudoku rakendust Google Play's:

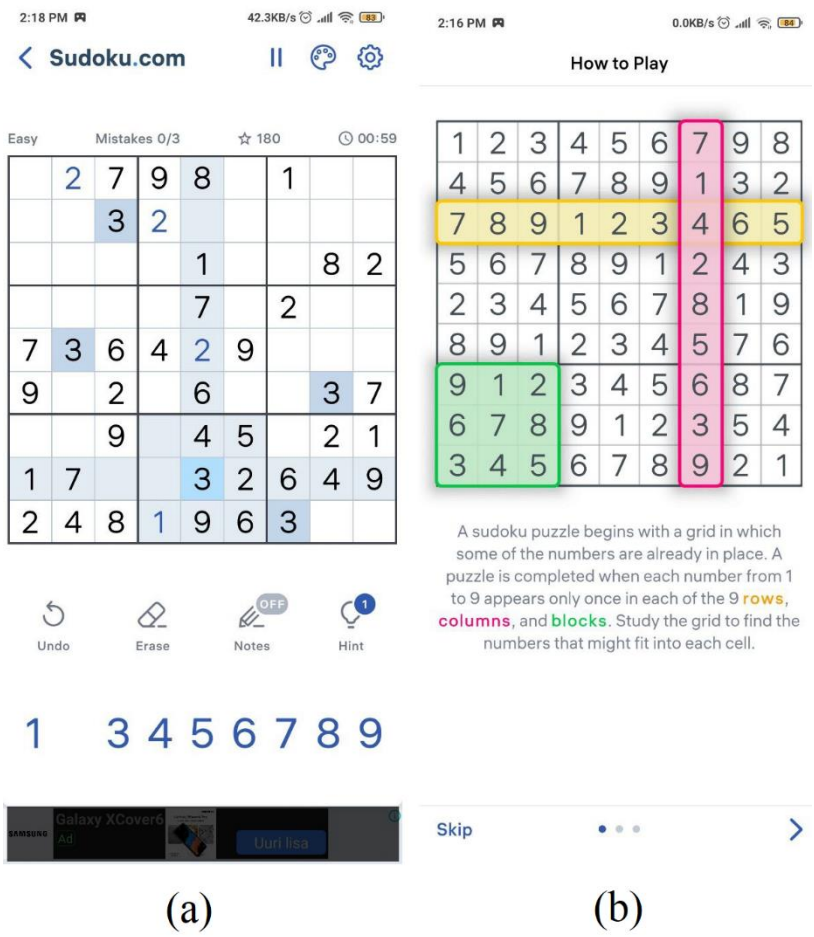
- Sudoku – Classic Sudoku Puzzle (arendaja Beetles Studio, >50mln allalaadimist),
- Sudoku.com – classic sudoku (arendaja Easybrain, >50mln allalaadminist),
- Sudoku – Classic Sudoku Puzzle (arendaja Guru Puzzle Game, >10mln allalaadimist).

Kõigis nendes rakendustes on sarnased võimalused. Seal saab lahendada erineva raskusega Sudoku, saada vihjeid (ühe ruudu lahendust), teha pliiatsimärke - märkida ruutude kandiaadid. Rakendustes seletatakse kasutajale Sudoku reegleid, aga mitte lahendamise tehnikaid. Samuti on võimalik kõikides rakendustes vahetada dünaamiliselt värviskeemi.

Beetles Studio rakenduses on võimalik vahetada numbrite suurust. Easybrain rakenduses on menüüs olemas punkt Rules (reeglid) – kui seda vajutada, avatakse kasutajale brauseris lehekülg, kus on kirjeldatud lahendamise tehnikad.

Joonisel 4 võib näha Easybrain rakenduse Sudoku ruudustikku ning reeglite seletust - kasutajaliidesed on kõigis kolmes rakenduses peaaegu samasugused.

Kõik kolm rakendust sobivad hästi Sudoku lahendamiseks, kuid mitte lahendamise tehikate õppimiseks.

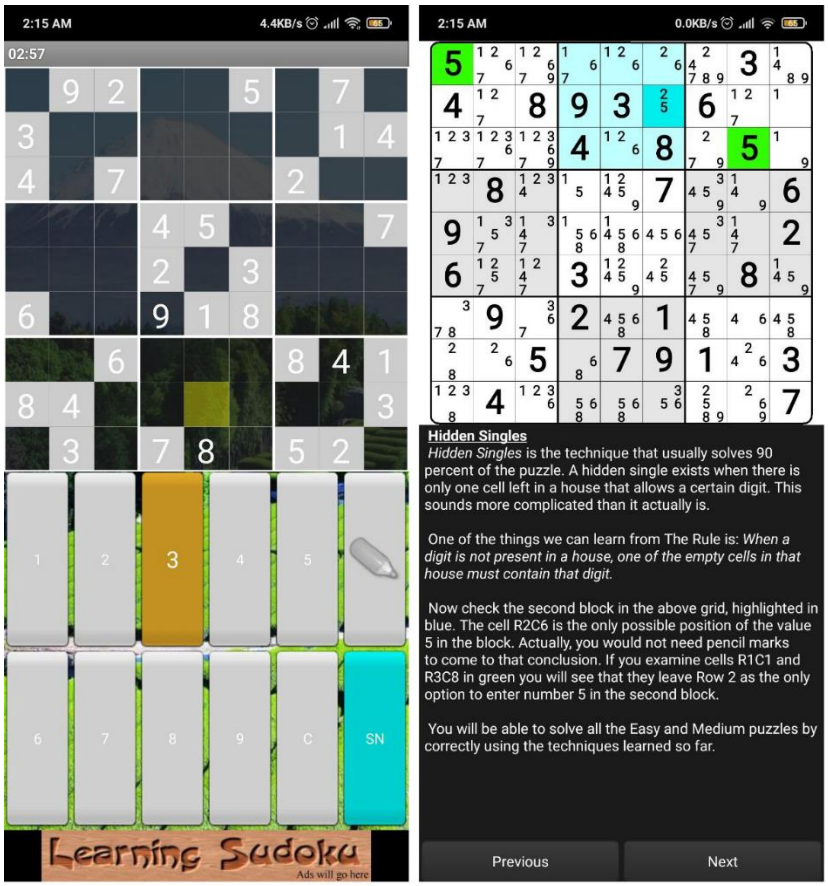


Joonis 4. Easybrain rakendus: (a) ruudustik, (b) reeglite seletus

2.2.2 Sudoku Learning

Google Play's oli leitud rakendus, mis on mõeldud õppimiseks ja mille nimeks on Sudoku Learning. Rakendus on vana, viimane uuendus oli 2012. aastal ning rakenduse avamisel teatab süsteem, et rakendus võib töötada ebakorrektselt.

Rakenduses on olemas paljude tehnikate seletused ja näited iga tehnika jaoks. Tehnikad on detailselt kirjeldatud. Näited paiknevad seletustest eraldi – selleks, et neid avada, tuleb minna tagasi menüüsse. Rakenduses on võimalik teha pliitasimärke, aga puudub võimalus saada vihjeid. Joonisel 5 võib näha rakenduse ruudustikku ning tehnika seletuse näidet.



Joonis 5. Sudoku Learning rakendus: (a) ruudustik, (b) tehnika seletus

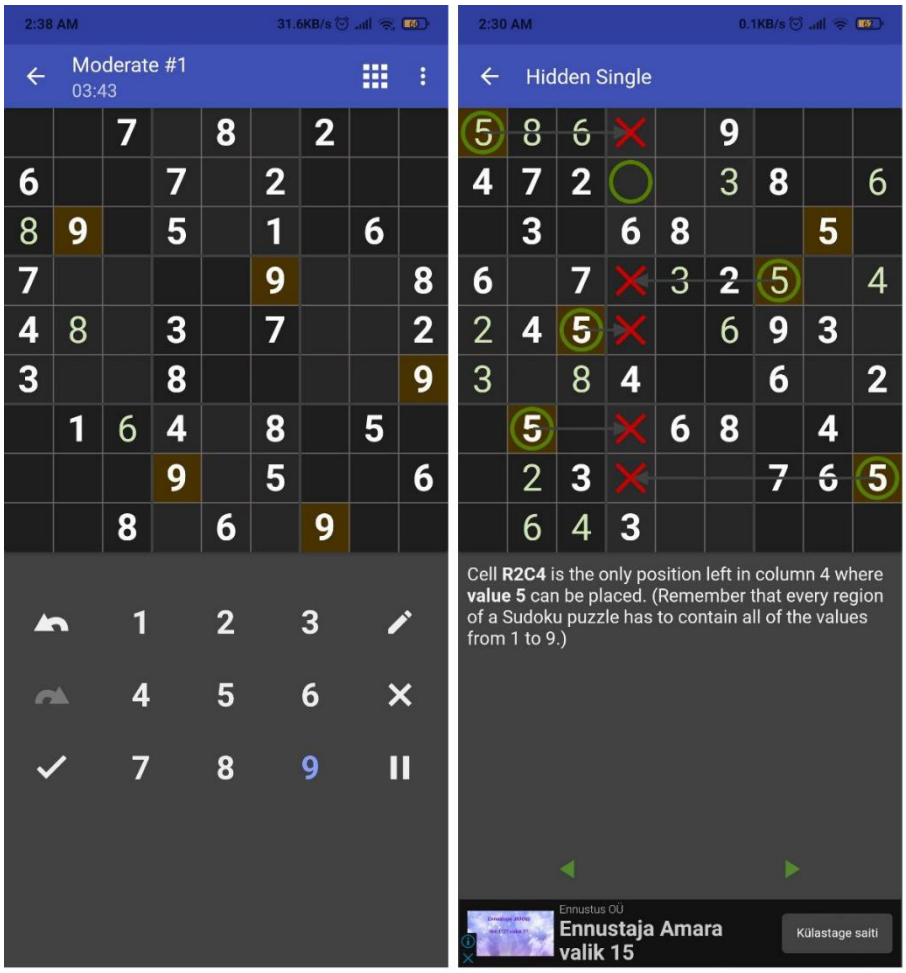
2.2.3 Andoku 3

Veel üheks rakenduseks, kus on olemas tehnikate kirjeldused on Andoku 3. Selles rakenduses on paljude tehnikate kirjeldused, tehnikad on jagatud raskuse järgi mitmesse kategooriasse. Seletused põhinevad dünaamilistel näidetel ning need näited on iga kord erinevad.

Sudoku lahendamisel saab sarnaselt teiste rakendustega teha pliiatsimärke ning küsida vihjeid – erinevuseks on see, et selles rakenduses vihjed ei ole lihtsalt ruutude lahendused, vaid nad ütlevad, mis tehnikat on vaja kasutada, et lahendada antud Sudoku.

Samuti on rakenduses olemas võimalus vahetada värviskeemi, keelt ning sisestada Sudoku, kas käsitsi või kasutades kaamerat.

Joonisel 6 võib näha rakenduse ruudustikku ning tehnika seletuse näidet.



Joonis 6. Andoku 3 rakendus: (a) ruudustik, (b) tehnika seletus

2.3 Olemasolevad töölauarakendused

Kuna osa inimesi lahendab Sudoku arvutis, on mõistlik teha ülevaadet olemasolevatest Sudoku rakendustest Windows platvormil.

2.3.1 Microsoft Store

Windows'i jaoks on olemas Microsoft Store, kus on üsna palju erinevaid Sudoku rakendusi. Sarnaselt materjalide ja mobiilrakendustega, vaadeldakse kolm esimest tasuta rakendust, mida näitab Microsoft Store:

- Microsoft Sudoku (arendaja Xbox Game Studios),
- Sudoku!!! (arendaja Spider Solitaire Sudoku),

- Sudoku Solver Generator (arendaja Anthony C).

Kõigis kolmes rakenduses on võimalik lahendada erineva raskusega Sudoku ning küsida vihjeid (ruutude lahendusi). Neist kolmest ainult Microsoft Sudoku's on võimalus sisestada pliiatsimärke, vahetada värviskeemi ja keelt. Samuti on seal võimalik vahetada numbrid teisteks märkideks.

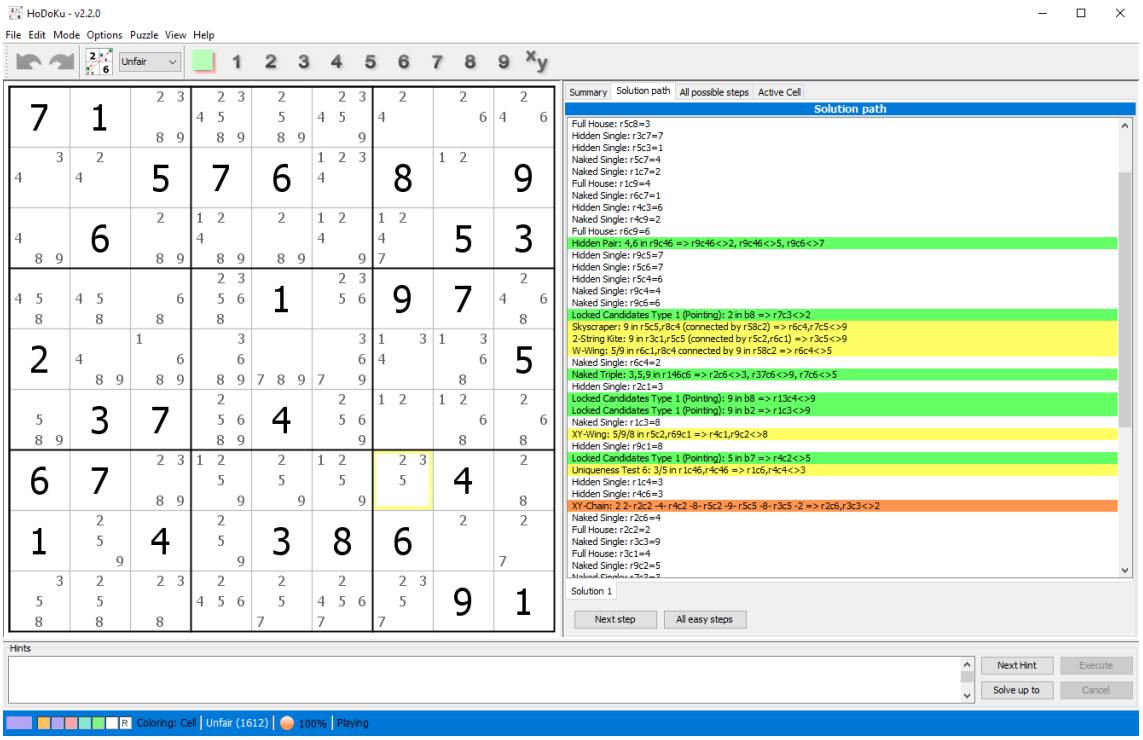
Sudoku lahendamise tehnikad ei ole kirjeldatud üheski nendest rakendustest.

2.3.2 HoDoKu

Oli leitud üks töölauarakendus, mis on tehtud edasijõudnud lahendajate jaoks – HoDoKu [9]. Rakenduses on palju erinevaid võimalusi, ta saab genereerida Sudoku, lahendada neid ning analüüsida. Sudoku genereerimisel on võimalik seadistada, mis tehnikaid tuleb genereeritud Sudoku lahendamise jaoks kasutada. Rakendus saab analüüsida antud Sudoku ja anda tehnikatel põhinevaid vihjeid ning näidata Sudoku lahenduskäik.

On olemas ka õppimise ja harjutamise režiimid. Mõlemas režiimis on võimalik valida tehnikad ning generaator teeb sellise Sudoku, mille lahendamiseks peab kasutama valitud tehnikaid vähemalt 1 korda. Erinevus režiimide vahel on see, et õppimise režiimis rakendus lahendab Sudoku hetkeni, millel tuleb valitud tehnikaid kasutada.

Joonisel 7 võib näha HoDoKu rakenduse kasutajaliides – Sudoku ruudustik ning lahenduskäik.



Joonis 7. HoDoKu rakendus

2.4 Pakutav lahendus

Selleks, et muuta Sudoku lahendamise õppimist mugavamaks, pakub autor välja rakendust, mida on võimalik kasutada nii nutitelefonis (Android) kui ka arvutis (Windows). Pakutavas rakenduses on ühendatud materjalid, kus kasutajale seletatakse lahti Sudoku lahendamise tehnikaid ja võimalus lahendada Sudoku, et saadud teadmisi rakendada. Samuti sisaldavad õppematerjalid Sudoku ülesandeid, kus eesmärgiks ei ole lahendada terve Sudoku, vaid kasutada õpitut tehnikat, et saada konkreetset tulemust. Rakenduse peamiseks eesmärgiks on aidata kasutajat õppimise käigus, seda võib vaadelda nagu interaktiivset õpikut. Platvormidest piirdutakse esialgselt Android ja Windows platvormidega, sest nendel saab autor rakendust ulatuslikult testida.

Kõikidel vaadeldavatel rakendustel on olemas sarnane funktsionaalsus. Nendes saab lahendada erineva raskusega Sudoku ning saada vihjeid. Need funktsioonid peavad olema tehtud ka loodaval rakendusel. Kuna loodav rakendus on suunatud tehnikate õppimisele, annab see vihjeid nii, et ütleb kasutajale, mis tehnikaid peab kasutama antud Sudoku lahendamiseks.

3 Loodava rakenduse analüüs

Antud peatükis analüüsitakse rakendust, mida hakatakse looma ning valmistatakse ette arenduseks.

3.1 Nõuete määramine

Nõuded on kirjeldatud kasutajalugude kujul, kus on põhirolliks kasutaja, kes soovib rakenduse abil õppida lahendada Sudoku.

Selles peatükis on loetletud minimaalsed nõuded, mida on vaja täita, et rakendus saaks täita oma eesmärki. Neid nõudeid teostatakse lõputöö raames. Muud nõuded, mis võivad olla rakenduse edasiarenduseks, on loetletud antud töö lõpus.

Kasutaja nõuded:

- Kasutajana soovin lahendada Sudoku.
- Kasutajana soovin lahendada erineva raskusega Sudoku.
- Kasutajana soovin kasutada Sudoku lahendamisel pliiatsimärke.
- Kasutajana soovin näha Sudoku lahendamisel oma vigu.
- Kasutajana soovin saada Sudoku lahendamisel vihjeid.
- Kasutajana soovin näha rakenduses Sudoku reegleid.
- Kasutajana soovin näha rakenduses Sudoku lahendamise tehnikate seletusi.
- Kasutajana soovin lahendada Sudoku ülesandeid.
- Kasutajana soovin vahetada keelt rakenduses.
- Kasutaja soovin vahetada värviskeemi.

- Kasutajana soovin vahetada teksti suurust.
- Kasutajana soovin kasutada rakendust nii nutitelefonis kui ka arvutis.

3.2 Tehnoloogiate valik

Esimeseks tähtsaks küsimuseks on tehnoloogiate valik. Esiteks on vaja valida programmeerimiskeelt ja raamistikku. Selleks on mõistlik teha ülevaadet olemasolevatest variantidest.

Tasuta tehnoloogiaid, mis võimaldavad luua rakendusi korraga nii mobiilsetele platvormidele kui ka töölauarakendusena ei ole palju ja sellepärast on mõistlikum kohe võrrelda neid raamistikku kaupa, mitte keele kaupa.

Tehnoloogiaid võrreldakse järgmiste parameetrite järgi:

- Jõudlus – Sudoku genereerimine ja lahendamine võivad olla ressursimahukad protsessid, aga kasutaja ooteaeg peab olema võimalikult väike;
- Dokumenteeritus – põhjalik dokumentatsioon lihtsustab õppimist ning tihti esinevate probleemide lahendamist;
- Populaarsus – mida populaarsem on tehnoloogia, seda lihtsam on leida infot ja lahendusi erinevatele probleemidele, samuti on tavaliselt rohkem erinevaid teeke, mida saab kasutada;
- Kogemus – isikliku kogemuse olemasolu aitab kaasa kiirele arendamisele.

Allpool on toodud leitud raamistikud:

- Ionic – kasutatakse HTML (*HyperText Markup Language*), CSS (*Cascading Style Sheets*), JavaScript;
- Apache Cordova – kasutatakse HTML, CSS, JavaScript;
- React Native – kasutatakse HTML, CSS, JavaScript;
- Flutter – kasutatakse Dart programmeerimiskeelt ja sisseehitatud vidinaid (komponente);

- Solar2D – varem Corona SDK (*Software Development Kit*) – kasutatakse Lua programmeerimiskeelt.

3.2.1 Ionic

Ionic on avatud lähtekoodiga kasutajaliidese tööriistakomplekt mobiilirakenduste loomiseks, mis kasutab veebitehnoloogiaid – HTML, CSS ja JavaScript – ja millel on integratsioonid populaarsete raamistikkudega, nagu Angular, React ja Vue.

Töölauarakendusena saab Ionic rakendust käivitada teise raamistiku kaudu, mille nimeks on Electron.js. See on omakorda käitusaegne raamistik, mis võimaldab luua HTML, CSS ja JavaScriptiga töölauarakendusi. See on avatud lähtekoodiga projekt, mida käivitas üks GitHubi inseneridest.

Ionic jõudlust peetakse umbes samaks kui platvormipõhistel mobiilrakendustel, näiteks Ionic ja Reacti Native võrdluses selgus, et React Native oli kiirem mõnedel juhtudel [10].

Olukord on teine töölauarakeduste puhul - võrreldi Electron ja Flutter Desktop jõudlust macOS süsteemis. Esiteks, isegi lihtsam program, mis kuvab ekraanile teksti võtab Electroni puhul 184 MB (megabait), Flutteri puhul oli see näitaja 23-37 MB. Samuti võttis Electroni rakendus rohkem protsessori, muutmälu ja graafikakaardi ressurse [11].

Ionic raamistikul on olemas üksikasjalik dokumentatsioon koos näidetega. Statista järgi oli Ionic 2021. aastal populaarsuselt 4. kohal mobiilsete platvormiüleste raamistike hulgas, kuid selle populaarsus on 2019. aastast oluliselt kahanenud [12]. Autoril kogemus selle raamistikuga puudub, kuid on kogemus Vue ja React raamistikega, mida Ionic toetab. Puuduste hulga võib lisada ka seda, et Ionic ei toeta nn *hot reload*, mis võimaldab näha koodi muudatusi rakenduses kohe, seda ei ole vaja taas kompileerida ja käivitada.

3.2.2 Apache Cordova

Samuti kui Ionic, kasutab Cordova HTML, CSS ja JavaScript tehnoloogiaid ning seda saab käivitada kui töölauarakendust Electron JS abil. Cordova rakendused on üldiselt aeglasemad kui need, mis on tehtud React Native abil [13] või Flutteri abil [14].

Apache Cordova raamistikul on olemas dokumentatsioon koos näidetega. Populaarsuselt oli 2021. aastal Cordova 3. kohal ning selle populaarsus vähenes võrreldes 2019. aastaga peaaegu 2 korda [12]. Autoril puudub kogemus selle tehnoloogiaga.

3.2.3 React Native

React Native kasutab JavaScript programmeerimiskeelt, aga HTML asemel kasutab ta platvormipõhiseid vaateid ja komponente. Selleks, et käivitada React Native rakendusi Windows (või macOS) süsteemil, on olemas veel üks raamistik, React Native for Windows + macOS.

React Native rakendused töötavad kiiremini kui need, mis kasutavad Ionic [10] ja Apache Cordova [13] raamistikke, kuid nad on ikka aeglasemad kui platvormipõhised [15] ja Flutter [16] rakendused.

Nii mobiilsel kui ka töölauarakenduse versioonil on olemas üksikasjalik dokumentatsioon, kus on palju näiteid. Aastatel 2019-2020 oli React Native kõige populaarsem mobiilne platvormiülene raamistik, kuid 2021. aastal liikus ta teisele kohale [12]. Autoril on olemas natuke kogemust React Native raamistikuga.

3.2.4 Solar2D

Solar2D on mängumootor, mis põhineb Lua programmeerimiskeelel. See on avatud lähtekoodiga projekt, mis on tehtud Corona SDK raamistiku põhjal, mida enam äriliselt ei toetata. Vastavalt dokumentatsioonile, toetab Solar2D palju platvorme, sealhulgas Windows ja Android [17].

Solard2D jõudluse võrdlust ei leitud. Sellel raamistikul on olemas detailne dokumentatsioon näidetega. Statista järgi ei olnud Solar2D ega Corona SDK platvormiüleste mobiilrakenduste puhul populaarne – nad ei sattunud esimese 12 raamistiku hulka [12].

Autoril puudub kogemus Solar2D raamistikuga ja Lua keelega.

3.2.5 Flutter

Flutter on Google'i avatud lähtekoodiga raamistik mitme platvormi rakenduste loomiseks ühe koodipõhjaga. Flutter rakenduste arendamiseks kasutatakse Dart

programmerimiskeelt ning graafika jaoks kasutatakse Impeller - avatud lähtekoodiga 2D graafikateeki, mis pakub API-sid (*Application Programming Interface*), mis töötavad erinevatel riist- ja tarkvaraplatvormidel.

Nagu oli eelnevalt mainitud, on Flutter jõudluse järgi parem kui Electron (Ionic ja Apache Cordova) ning React Native. Solar2D kohta andmeid ei leitud. Flutteril on olemas üksikasjalik dokumentatsioon koos interaktiivsete näidetega, mida saab kohe brauseris muuta ja käivitada. Statista andmed näitavad, et Flutteri populaarsus arendajate seas on kasvanud 30st protsendist 2019 aastal 42 protsendini 2021 aastal, millal ta sai kõige populaarsemaks raamistikuks [12]. Autoril on olemas kogemus Flutteriga mobiilrakenduste arenduses.

3.2.6 Võrdlus

Tabelis 1 on toodud raamistikkude võrdluse kokkuvõte.

Tabel 1. Raamistikkude võrdlus

	Jõudlus mobiilrakendus / töölauarakendus	Dokumenteeritus	Populaarsus	Kogemus
Ionic	väga hea / rahuldav	väga hea	suur, kahaneb	puudub
Apache Cordova	hea / -	hea	suur, kiiresti kahaneb	puudub
React Native	väga hea / väga hea	väga hea	suur, kahaneb	vähe
Solar2D	- / -	hea	väike	puudub
Flutter	parim / parim	väga hea	suur, kasvab	on olemas

Raamistikuks on valitud Flutter ning programmeerimiskeeleks Dart, sest see variant on jõudluse ja kogemuse poolt parim. Samuti on Flutteril olemas üksikasjalik dokumentatsioon ja see on populaarne, mis tähendab, et selle kohta on palju artikleid ja õppematerjale.

3.3 Koodihalduskeskkond

Kolmeks parimaks koodihalduskeskkonnaks, mis toetavad Git-i, peetakse GitHub, GitLab ja Bitbucket [18]. Neid kõiki on võimalik kasutada tasuta ning kõik nad pakuvad

võimalust luua nii avalikud kui isiklikud repositooriumid [19]. Kuna loodav rakendus ei ole suur ning sellel ei ole erilisi nõudeid peale koodi versioonihaldust, nagu pidevat integratsiooni, sobivad kõik kolm keskkonda võrdselt antud rakenduse jaoks.

3.4 Integreeritud arenduskeskkonna valik

Flutter rakenduste arenduseks peetakse parimateks keskkondadeks Android Studio, Visual Studio Code ja IntelliJ IDEA [20]. Kõik kolm IDE-d (*Integrated development environment*) toetavad Flutter rakenduste arendust pistikprogrammide (Android Studio, IntelliJ) või laienduste (Visual Studio Code) abil. Android Studio põhineb IntelliJ IDEA-l ning seetõttu on sellega sarnane, nii kasutuskogemuse kui ka jõudluse poolest. Kõik kolm IDE-d pakuvad põhilisi tööriiste ja võimalusi nagu Git-i integratsioon, rakenduse kompileerimine ja käivitamine. Samuti toetavad kõik rakendused Flutter-i mugavat funktsiooni, *hot reload*, mis võimaldab rakendada töötavasse rakendusse muudatused koodis nii, et rakendust ei ole vaja uuesti kompileerida. Andoid Studio ja IntelliJ pakuvad mõningaid tööriiste, mida ei ole Visual Studio Code-is, aga samas on Visual Studio Code kergekaalulisem programm, millega on autoril rohkem kogemust. Seetõttu oli antud rakenduse arendamiseks valitud Visual Studio Code.

3.5 Arhitektuur

Antud rakendus koosneb kolmest komponendist:

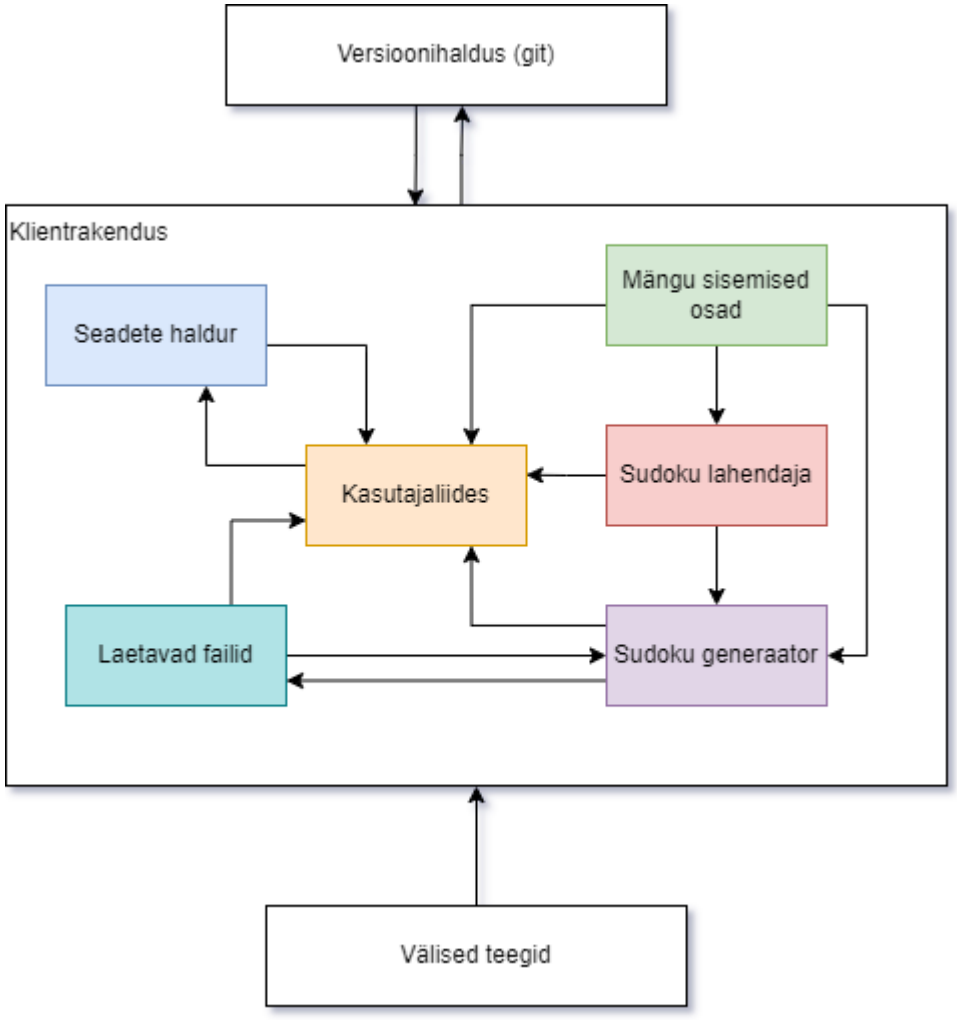
- Klientrakendus,
- Versioonihaldus,
- Välised teegid.

Klientrakendus koosneb omakorda veel mitmest osast või moodulist:

- Mängu sisemised osad - Sudoku domeenimudel, loogika ja ülesanded,
- Sudoku lahendaja,
- Sudoku generaator,
- Kasutajaliides,

- Seadete haldur,
- Laetavad failid – pildid, JSON (*JavaScript Object Notation*) failid.

Loodava rakenduse arhitektuur ei ole kihiline nagu see on tavaliselt veebirakenduste puhul. Kuna rakenduses ei ole tagarakendust ja andmebaasi ning rakendus on rohkem sarnane mänguga, on selle arhitektuur pigem modulaarne – rakenduses on olemas mitu olulist osa ning iga osa täidab oma kindlat ülesannet. Rakenduse arhitektuuri võib näha Joonisel 8.



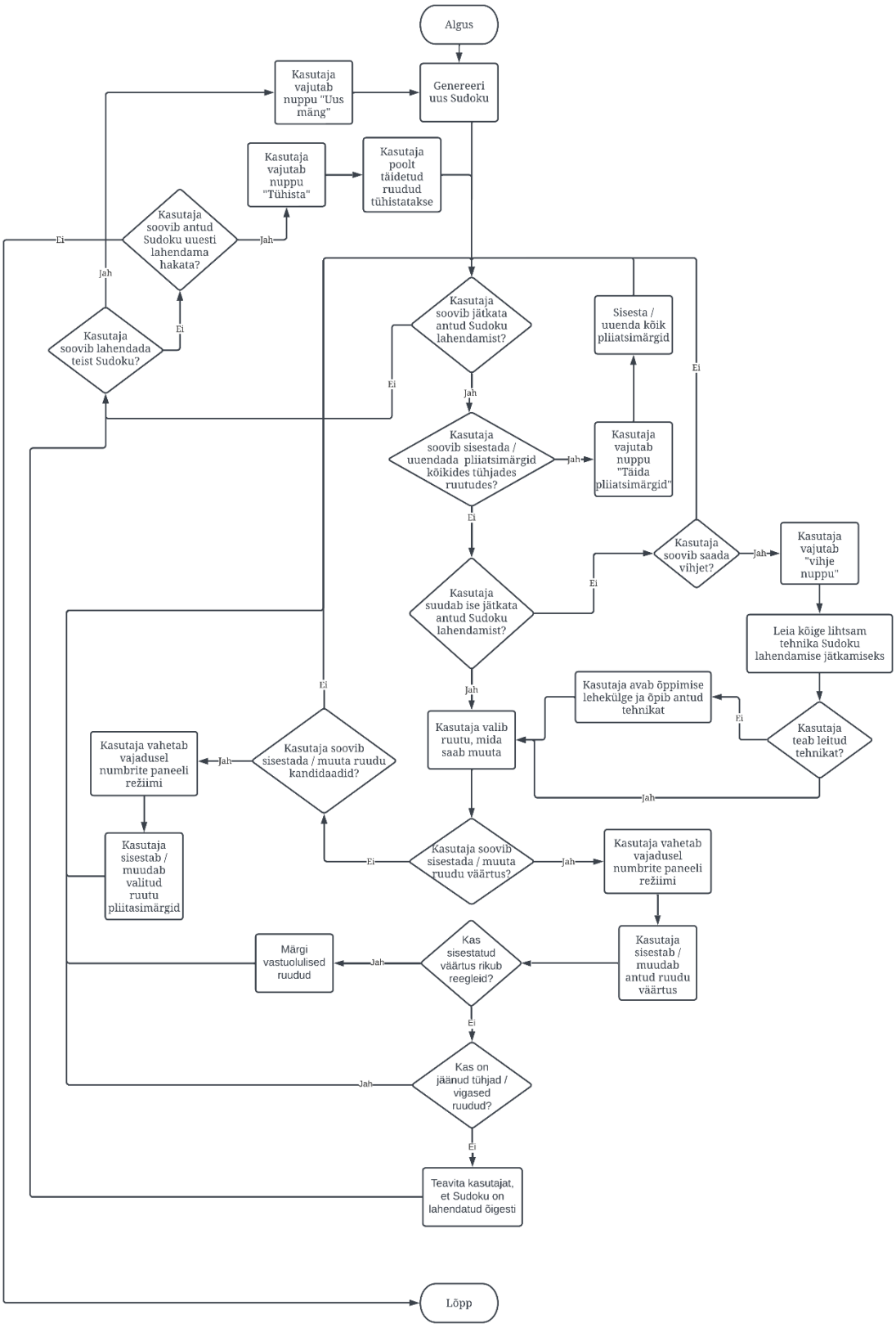
Joonis 8. Rakenduse arhitektuur

Rakenduse keskseks osaks võib nimetada kasutajaliidest, sest just kasutajaliides ühendab kõiki teisi osi kokku. Mõned moodulid on aga seotud oteseselt ükteise vahel, näiteks Sudoku generaator kasutab genereerimise protsessis igal sammul Sudoku lahendajat sammu tulemuse kontrollimiseks.

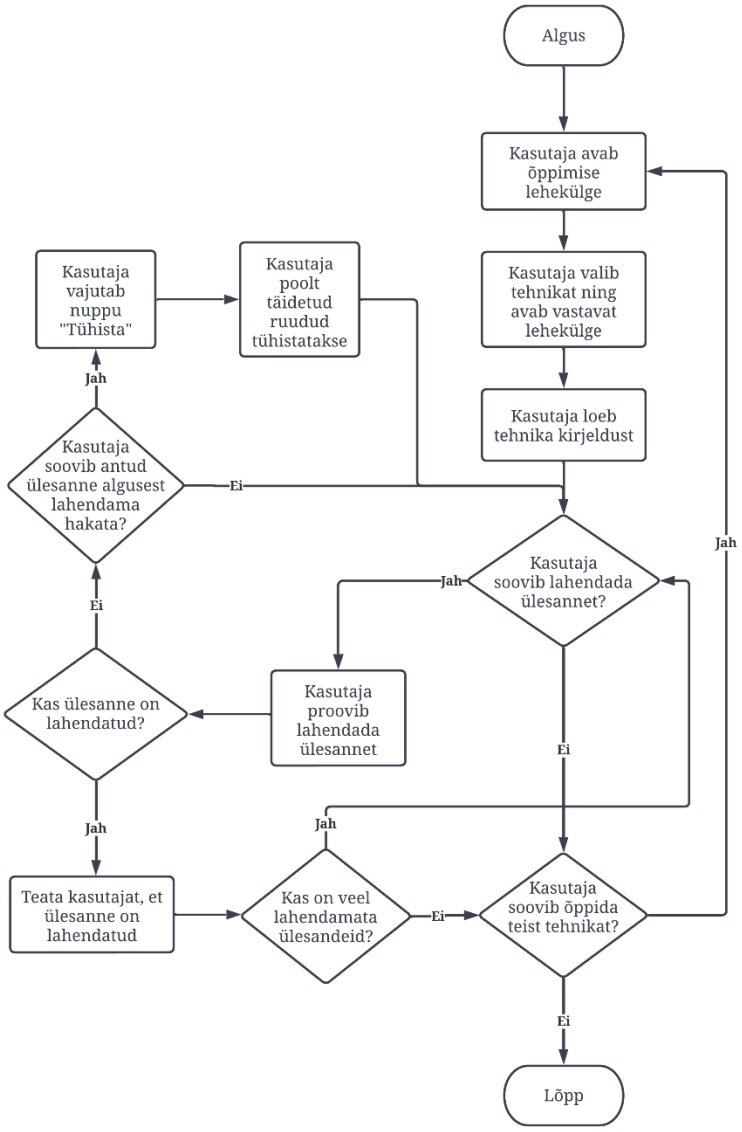
3.6 Kasutajakogemuse disain

Rakenduses saab kasutaja tegeleda kahe peamise asjaga – Sudoku lahendamisega ja õppimisega. Sudoku lahendamisel võib kasutaja saada rakenduselt abi mitmel kujul. Esiteks saab rakendus täita kasutaja jaoks kõik võimalikud pliiatsimärgid – sisestada tühjadesse ruutudesse kõiki numbreid, mida võiks sinna panna reegleid rikumata. See ei vaja mingite tehnikate rakendamist ning aitab kasutajal säästa aega. Teiseks on rakendus võimeline anda kasutajale vihjeid – teavitada kasutajat, mis tehnikat ja kus tuleb kasutada, et jätkata antud Sudoku lahendamist. Joonisel 9 on toodud välja kasutajakogemus, mis kirjeldab Sudoku lahendamist loodavas rakenduses.

Teiseks peamiseks tegevuseks on õppimine, mis sisaldab materjalide lugemist ning ülesannete lahendamist. Iga lehekülg õppematerjalidega sisaldab vähemalt ühte ülesannet, mida kasutaja saab soovi korral lahendada. Joonisel 10 toodud kasutajakogemus kirjeldab õppimise protsessi loodavas rakenduses. Selleks, et hoida kokku ruumi ja muuta diagrammi loetavamaks, on numbrite sisestamine ruudustikku kujundatud lihtsustatult: „Kasutaja proovib lahendada ülesannet“. Ruutude väärtuste ja kandidaatide sisestamise protsess on ülesannetes samasugune nagu Sudoku lahendamise puhul, mis on kirjeldatud Joonisel 9.



Joonis 9. Sudoku lahendamise kasutajakogemuse diagramm



Joonis 10. Õppimise kasutatjakogemuse diagramm

Rakenduses on samuti võimalik vahetada keelt, värviskeemi ja teksti suurust, kuid kuna iga tegevuse jaoks tuleb avada seadete lehekülge ja muuta vastavat seadet, jäeti neid välja lihtsuse ja lühiduse tõttu.

3.7 Analüüsi kokkuvõte

Rakenduse analüüsis olid kirjeldatud nõuded rakendusele ning nende alusel olid koostatud kasutajakogemuse diagrammid.

Samuti olid võrreldud erinevad raamistikud mitme kriteeriumi alusel. Rakenduse arendamiseks oli valitud raamistik Flutter, mis töötab Dart programmeerimiskeeles. Koodihalduskeskkonnaks oli valitud GitHub, kuna funktsionaalselt sobisid kõik

keskkonnad, aga GitHubi kasutatakse tihti nagu ka arendaja portfoolio. Arenduskeskkonnaks oli valitud Visual Studio Code.

4 Rakenduse loomine

Antud peatükis on kirjeldatud rakenduse loomine ning sellega kaasnevate küsimuste analüüs. Töö peamiseks eesmärgiks on luua platvormiülene rakendus, mis töötab Android ja Windows platvormil ning aitab inimestel õppida Sudoku lahendamise tehnikaid. Rakenduse kood asub GitHub repositooriumis, mida võib leida Lisas 2. Rakenduse nimeks sai Sudoku Guide.

4.1 Keskkonna seadistus ja Flutter projekti loomine

Enne projekti loomist oli paigaldatud Android SDK ja loodud Android emulaator Android Studio-s, mida hiljem kasutati rakenduse testimiseks arenduse käigus. Samuti oli paigaldatud Flutter SDK versioon 3.7.6, mis oli sel hetkel viimane versioon. Flutter rakenduste arendamiseks Windows platvormil on samuti vaja Visual Studio 2022 või Visual Studio Build Tools 2022 ning lisakomponent „Desktop development with C++“, mida on võimalik paigaldada Visual Studio Installer rakenduse abil [21]. Visual Studio 2022 oli arvutis juba paigaldatud, lisakomponent sai paigaldatud töö käigus.

Sellele järgnes arenduskeskkonna seadistus – Visual Studio Code'i olid paigaldatud Dart ja Flutter laiendused, mida on vaja rakenduste kompileerimiseks ja käivitamiseks.

Project oli loodud „flutter new“ käsu abil, mida käivitati Windows käsureas. Edasine projekti haldamine toimub Visual Studio Code keskkonnas.

4.2 Rakenduse arhitektuur

Flutter raamistik loob uutes projektides mitu kausta, suurem osa nendest on platovrmispetsiifiline kood, mis on jaotatud kaustadesse platvormi kaupa ning mida kasutatakse rakenduse kompileerimiseks nende platvormide jaoks. Rakenduse kood läheb *lib* kausta. Projekti mugavamaks haldamiseks on arendatava rakenduse failid jaotatud järgnevatesse kaustadesse:

- *assets* – failid, mida laetakse rakendusse käitusajal – pildid ja mall-Sudoku;

- *game_internals* – Sudoku-ga seotud klassid – mänguloogika, Sudoku lahendaja ja generaator;
- *l10n* – ARB (*Application Resource Bundle*) failid, mille põhjal genereeritakse teegi abil abiklassid rakenduse lokaliseerimiseks;
- *name_pickers* – abiklassid lokaliseeritud sõnede pärimiseks genereeritud klassidest;
- *pages* – lehekülgede põhjad, mille vahel saab kasutaja navigeerida;
- *settings* – klassid seadete haldamiseks rakenduses;
- *widgets* – komponendid (vidinad), millega täidetakse lehekülgi.

4.3 Domeenimudel

Rakenduses on 5 peamist klassi, mida kasutatakse infromatsiooni hoidmiseks ja edastamiseks:

- *SudokuCell* klass esitab ühe Sudoku ruutu ning sisaldab väärtust, kandidaate, rea- ja veerunumbri ruudustikkus ning omadust, mis määrab, kas antud ruudu väärtust ja kandidaate on võimalik muuta.
- *Sudoku* klass esitab Sudoku mõistatust. See hoiab infot antud Sudoku ruudustiku seisu kohta (*SudokuCell* klassi abil), lahendust, algseisu ja vigaseid ruute. Samuti sisaldab see klass meetodeid võidu tingimuse kontrollimiseks, vigade leidmiseks, kõikide kandidaatide täitmiseks ruudustikus ning ruutude ja nende väärtuste mugavaks küsimiseks – neid võib küsida ridade, veergude ja plokide kaupa, samuti on võimalik küsida antud ruutu kaaslasi. Klassis on olemas ka meetodid, et kontrollida, kas antud ruudud asuvad ühes reas, veerus või plokis.
- *TechniqueResult* klassi kasutatakse Sudoku lahendamisel, et edastada infot selle kohta, mis tehnikat on võimalik kasutada Sudoku lahendamiseks ja mida sellega võib saavutada – leitud ruudu väärtust või ruutude nimekirja ning kandidaate, mida võib antud ruutudes eemaldada.

- *SudokuTemplate* klassi kasutatakse mall-Sudoku genereerimisel (kirjeldatud peatükis **Mall-Sudoku genereerimine**) ja lõpliku Sudoku genereerimisel mall-Sudoku alusel. *SudokuTemplate* klass sisaldab Sudoku algseisu ja lahendust ning meetodeid selle klassi objektide serialiseerimiseks ja deserialiseerimiseks.

- *SudokuProblem* esitab Sudoku ülesannet ning sisaldab ülesande hetkeseisu, algseisu ja lahendust – kõik need omadused on nimekirjad, mis sisaldavad *SudokuCell* objekte. Samuti sisaldab klass omadusi, mis näitavad ruutude arvu, mida tuleb muuta ülesande lahendamiseks, valesti muudetud ruutude arvu ning ruute, mis peavad ülesande algseisus olema märgitud teise värviga.

4.4 Sudoku lahendamine

Käesoleva töö raames tehtav rakendus peab oskama lahendada Sudoku mitmel eesmärgil. Sudoku on vaja lahendada Sudoku genereerimise protsessis, Sudoku lahenduse unikaalsuse kontrollimiseks ning selleks, et anda kasutajale vihjeid lahendamiseks. Sellised eesmärgid seadistavad mõned nõuded lahendamisviisile:

- programm peab leidma lahendust nii kiiresti kui võimalik, et kasutaja ei peaks kaua ootama;

- lahenduse unikaalsuse kontrollimiseks on vaja, et programm saaks leida vähemalt 2 lahendust, kui nad on olemas või tunda ära olukordi, kus antud mõistatusel on rohkem kui üks lahendus;

- programm peab olema võimeline lahendama Sudoku kasutades reegleid ja lahendamistehnikaid, et anda kasutajal teada, mis tehnikaid tuleb lahendamiseks kasutada.

4.4.1 Sudoku lahendamise meetodid

Peamised lähenemised Sudoku lahendamisel on programmiline ja loogiline. Programmilise lähenemise alla kuuluvad näiteks toore jõu meetod ja tagurdamise meetod. Toore jõu meetodi puhul proovitakse ühekaupa kõik võimalikud lahendused – kõik võimalikud kombinatsioonid kõikide ruutude võimalikkudest väärtustest. See meetod ei ole efektiivne, kuna erinevaid kombinatsioone võib olla liiga palju. Võrreldes

toore jõu meetodiga on tagurdamise meetod efektiivsem, kuna see võimaldab eemaldada korraga palju võimalikke kombinatsioone.

Sudoku võib vaadelda kui piiranguga rahulolu probleemi (*constraint satisfaction problem*) – ülesannet, kus mingi objekti (ruudustikku) seis peab vastama antud piirangutele (Sudoku reeglitele).

Tagurdamise meetodi põhimõte piiranguga rahuloluprobleemi puhul – iga sammu järel, kui mingile muutujale antakse mingi väärtus, on vaja kontrollida, kas kõik piirangud on rahuldatud. Juhul kui mingi piirang on rikutud, on vaja minna tagasi seisuni, kus kõik piirangud olid rahuldatud ning proovida anda antud muutujale teise väärtuse [22]. Sudoku korral on muutujateks ruutude väärtused ning väärtusteks tavaliselt numbrid 1-9.

Loogiline meetod on Sudoku lahendamine kasutades loogilisi lahendamistehnikaid – seda lähenemist kasutavad tavaliselt inimesed. Sellisel lähenemisel on omad puudused, näiteks see ei saa garanteerida lahenduse leidmist, kuna see on piiratud tehnikatega, mida saab kasutada, aga on ka eelised, näiteks see on kiirem kui kõikide kombinatsioonide proovimine.

Rain Haabu töös [23] oli leitud, et kombineerides loogilisi lahendustehnikaid ja tagurdamise meetodi võib saada paremaid tulemusi võrreldes lihtsa targudamisega. Esialgu kasutatakse loogilisi lahendusmeetodeid, et leida nii paljude ruutude lahendused kui võimalik. Seejärel kasutatakse tagurdamist, et leida ülejäänud ruutude väärtused.

Kuna loodav rakendus genereerib ise Sudoku ning genereerimise ajal kontrollib, et saadud Sudoku oleks lahendatav tehnikate abil, ei mängi olulist rolli see, et loogiline meetod ei saa garanteerida üldjuhul tulemuse leidmist, sest kõik rakenduse poolt genereeritud Sudoku on lahendatavad tehnikate abil.

Loogilise meetodi kasutamine rakenduses annab veel ühe eelise Sudoku genereerimisel – puudub vajadus kontrollida eraldi lahenduse unikaalsust – kui rakendus proovis kasutada kõiki tehnikaid ja ikka kõikidesse ruutudesse sobib rohkem kui üks number, siis võib järeldada, et lahendusi on rohkem kui üks või tuleb lahendamiseks kasutada tehnikaid, mida rakenduses ei ole. Sel juhul astub generaator sammu tagasi ja püüab

eemaldada ruudustikkust teise numbri. Tulemuseks on Sudoku, millel on üks lahendus ja mis on kindlasti lahendatav rakenduses kasutatavate tehnikate abil.

Sama lahendamismeetodi saab kasutada ka vihjete genereerimiseks, kuna see kasutab kohe lahendamistehnikaid.

Ülaltoodud põhjustel oli valitud rakenduse jaoks loogiline meetod. Selle meetodi jaoks on vaja valida tehnikaid, mida see kasutama hakkab. Selleks, et piirata töö maht, oli otsustatud vaadelda kõik alg- ja kesktaseme tehnikad ning 3 edasijõudnud tehnikat – tegelikkuses on edasijõudnud tehnikaid rohkem.

Rakenduses vaadeldakse järgmiseid algtaseme tehnikaid:

- Sudoku reeglid,
- Ainus võimalik number,
- Ainus võimalik ruut,
- Alasti üksik,
- Alasti paar,
- Alasti kolmik,
- Peidetud üksik,
- Peidetud paar,
- Peidetud kolmik.

Samuti vaadeldakse kaks kesktaseme tehnikat, Osutav paar ja Osutav kolmik. Edasijõudnud tehnikatest on rakenduses esitatud X-Wing, Y-Wing ja Swordfish tehnikad.

4.4.2 Implementeeritud Sudoku lahendaja

Tehnikate jaoks on loodud liides *ISudokuTechnique*, mis sisaldab ainult ühte meetodit – *trySolve*. Seda liidest võib näha Joonisel 11.

```
abstract class ISudokuTechnique {
	TechniqueResult? trySolve(Sudoku sudoku, {bool applyResult = false});
}
```

Joonis 11. ISudokuTechnique liides

Tuleb märkida, et Dart programmeerimiskeeles puudub võtmesõna *interface*, mida kasutatakse paljudes keeltes liideste tähistamiseks. Liideste rolli täidavad Dart keeles abstraktsed klassid, mis sisaldavad kõiki vajalikke omadusi ja vajalike meetodite signatuure. Konkreetsed klassid, mis implementeerivad mingit liidest, peavad kirjutama üle kõiki liidese meetodeid ja omadusi [24].

Liidese meetod võtab parameetriteks *Sudoku* objekt, mida hakatakse lahendama ja *applyResult* parameetrit, mis näitab, kas leitud tulemust peab *Sudoku* objektis rakendama – vaikimisi on selle väärtus *false*, ning seda lülitatakse sisse ainult testimiseks. Meetod tagastab *TechniqueResult* objekti kui tehnika kasutamine õnnestus või *null* kui tehnika abil midagi ei saavutatud.

Iga tehnika jaoks on loodud eraldi klass, mis implementeerib *ISudokuTechnique* liidest. Joonisel 12 võib näha ühe tehnika klassi „Ainus võimalik number“ tehnika näitel.

```
class LastPossibleNumberTechnique implements ISudokuTechnique {
       const LastPossibleNumberTechnique();

  @override
  TechniqueResult? trySolve(Sudoku sudoku, {bool applyResult = false}) {
    for (var row in sudoku.board) {
      for (var cell in row.where((c) => c.value == 0)) {
        var missingNums = Sudoku.numSet.difference(
          sudoku.getPeerValues(cell.row, cell.col));
        if (missingNums.length == 1) {
          if (applyResult) {
            cell.candidates.clear();
            cell.value = missingNums.first;
            var peerCells = sudoku.getPeerCells(cell.row, cell.col);
            for (var pc in peerCells) {
              pc.candidates.remove(missingNums.first);
            }
          }
          return TechniqueResult(
            applicableCells: [
              SudokuCell(cell.row, cell.col, true, missingNums.first)
            ],
            usedTechnique: LastPossibleNumberTechnique
          );
        }
      }
    }
    return null;
  }
}
```

Joonis 12. „Ainus võimalik number“ tehnika klassi näide

Kuna kõik tehnikate klassid sisaldavad ainult ühte meetodit ning see meetod töötab igas klassis alati samamoodi, sõltudes ainult parameetritest, on loodud *TechniquesContainer* klass, mis hoiab staatilises omaduses iga tehnika klassi kohta ühte objekti. See lihtsustab tehnikatele ligipääsu ja samas aitab tehnikate puhul rakendada singli mustrit. Samuti sisaldab konteineri klass meetodit, mis aitab tehnikaid küsida.

Tehnikaid küsib konteinerist ja rakendab *SudokuSolver* klass. Sellel on olemas kaks staatilist meetodit, ühte kasutatakse Sudoku genereerimisel terve Sudoku ruudustiku lahendamiseks, teist – vihjete andmiseks, et leida esimest ruutu, millel saab rakendada mingit tehnikat. Joonisel 13 on toodud meetod, mida kasutatakse vihjete andmiseks.

```
static TechniqueResult? solveCellWithTechniques(
  Sudoku game,
  Iterable<SudokuTechniquesEnum> techniques,
  {bool applyResult = false}) {
  for (var technique in TechniquesContainer.getTechniques(techniques)) {
    var result = technique.trySolve(game, applyResult: applyResult);
    if (result != null) return result;
  }
  return null;
}
```

Joonis 13. Sudoku lahendaja meetod vihjete leidmiseks

Antud meetod võtab parameetriteks *Sudoku* objekti, mida proovitakse lahendada, tehnikate nimekirja, millega hakatakse Sudoku lahendama ja *applyResult* parameetrit, mida antakse edasi tehnikate klassidele, et vajadusel rakendada leitud tulemust.

4.5 Sudoku genereerimine

Täiesti uue ja juhusliku Sudoku genereerimine võib võtta palju aega, nii et teha seda reaalajas ei ole kõige efektiivsem lähenemine, sest genereerimise ajal peab kasutaja ootama.

Genereerida Sudoku kiiremini võivad aidata nn mall-Sudoku – see on selline Sudoku, mis oli eelnevalt genereeritud vastavalt mingitele parameetritele (kirjeldatud hiljem selles peatükis) ning mille alusel võib permutatsioonide abil kiiresti genereerida kasutaja jaoks lõplik Sudoku.

Mall-Sudokus kasutatakse numbrite asemel mingeid muid tähistusi (põhjuseks on esimene mall-Sudoku permutatsioon, mis on kirjeldatud hiljem selles peatükis). Seda saab teha, kuna Sudoku ei ole seotud just numbritega, numbrid on lihstalt tähised. Neid saab vabalt vahetada värvideks, sümboliteks, tähtedeks – peaasi, et nad ei korduks ridades, veergudes ja plokides. Selle töö raames kasutatakse mall-Sudokudes tähti a-i.

4.5.1 Esimene permutatsioon

Tähed a-i ei pea alati vastama numbritele 1-9 kindlas järjekorras. Sudokus on võimalik vahetada näiteks kõik ühed kaheksateks ja kõik kaheksad ühtedeks. Tulemuseks on ikka korrektne Sudoku, sest kõik numbrid esinevad kõikides ridades, veergudes ja plokkides ainult 1 kord. Nii saab teha kõikide numbrite paaridega mitu korda. See tähendab, et iga täht võib tähendada igas konkreetses Sudoku-s erinevat numbrit, näiteks ühes Sudoku-s

võib olla a=1, teises Sudokus – a=6. Kombineerides tähti ja numbreid, võib ühe mall-Sudoku põhjal saada 9! = 362 880 erinevat Sudoku. Täidetud ruutude paigutus ruudustikus jääb aga samasuguseks sõltumata tähtede tähendusest, nii et saadavad Sudoku näevad välja üsna sarnaselt.

4.5.2 Teine permutatsioon

Mall-Sudoku võib pöörata 90, 180 võib 270 kraadi võrra, lahendus jääb sel juhul samaks – ta on ikka unikaalne ja selle saavutamiseks on vaja kasutada samu tehnikaid, ta lihtsalt pöörleb koos täidetud ruutudega. Kasutades pööramist on võimalik teha ühest mall-Sudokust 4 erinevat varianti.

4.5.3 Kolmas permutatsioon

Mall-Sudoku saab peegeldada kas horisontaalselt, vertikaalselt või mõlemat pidi. Lahendus jääb põhimõtteliselt samaks, lihtsalt peegeldub koos täidetud ruutudega. Kasutades peegeldamist võib saada ühest mall-Sudokust 4 varianti.

4.5.4 Permutatsioonide kombineerimine

Kombineerides neid kolme permutatsioone on võimalik ühest mall-Sudokust saada 362 880 * 4 * 4 = 5 806 080 ehk rohkem kui 5 miljonit Sudoku. Mõned permutatsioonide kombinatsioonid võivad anda samu tulemusi (eriti kui mall-Sudoku on sümmeetriline, sest siis pööramine ja peegeldamine ei vaheta täidetud ruutude paigutust), kuid suurem osa neist on unikaalsed.

Kui rakenduses on iga Sudoku raskusastme kohta olemas teatud arv mall-Sudoku, siis saab rakendus kasutades permutatsioone genereerida nende põhjal kiiresti iga raskusastme jaoks Sudoku.

4.5.5 Mall-Sudoku genereerimine

Nii mall-Sudoku kui ka lõpliku Sudoku genereerimine toimub *SudokuGenerator* klassis. Enne genereerimist on teada, kui palju ruute peavad olema tühjad ning mis tehnikate abil peab saadud mall-Sudoku olema lahendatav. Need parameetrid koos esindavad antud Sudoku raskust. Mall-Sudoku genereerimine algab korrektse Sudoku lahenduse (ruudustik, kus kõik ruudud on täidetud vastavalt reeglitele) genereerimisest. Selle algoritm on võetud Internetist [25] ning implementeeritud Dart keeles. Seejärel hakatakse eemaldama ühekaupa ruutude väärtused. Iga sammu järel kontrollitakse, et

saadud Sudoku oleks võimalik lahendada antud tehnikate abil. Selleks kasutatakse *SudokuSolver* klassi.

Sudoku lahendamine ainult tehnikate abil toob kaasa ühe eelise - ei ole vaja eraldi kontrollida Sudoku lahenduse unikaalsust. Kui tekib selline olukord, kus peale ruudu eemaldamist on tulemusel kaks võimalikku lahendust, siis ei suuda rakendus seda lahendada ainult tehnikate abil ja läheb ühe sammu tagasi ning proovib eemaldada mingi muu ruudu väärtust.

Ruutude väärtusi eemaldatakse kuni on saavutatud nõutud tühjade ruutude arv – saadud ruudustik on genereeritava mall-Sudoku algseis. Seejärel asendatakse lahenduses ja algseisus numbrid tähtedega (tühjades ruutudes kasutatakse miinuseid) ja luuakse nende põhjal *SudokuTemplate* objekti, mida serialiseeritakse ning salvestatakse vastavasse JSON faili. Faile on 4, vastavalt Sudoku raskusastmetele (kerge, keskmine, raske, väga raske), igas failis on ühe raskusastmega mall-Sudoku. Joonisel 14 võib näha näidet kerge raskusastmega serialiseeritud mall-Sudoku-st, mida hoitakse JSON failis.


```
{
  "initState": "f-h-bd-ei--ga---fh----h-----ei-achg---c-gf---gd-eiha-c-gd-
fai--h-bid-f-g----c-eh-",
  "solution":
"fahcbdgeidbgaeicfhciefhgbdabeidachgfahcbgfdiegdfeihabcegdhfaicbhcbidefagifag
cbehd"
}
```

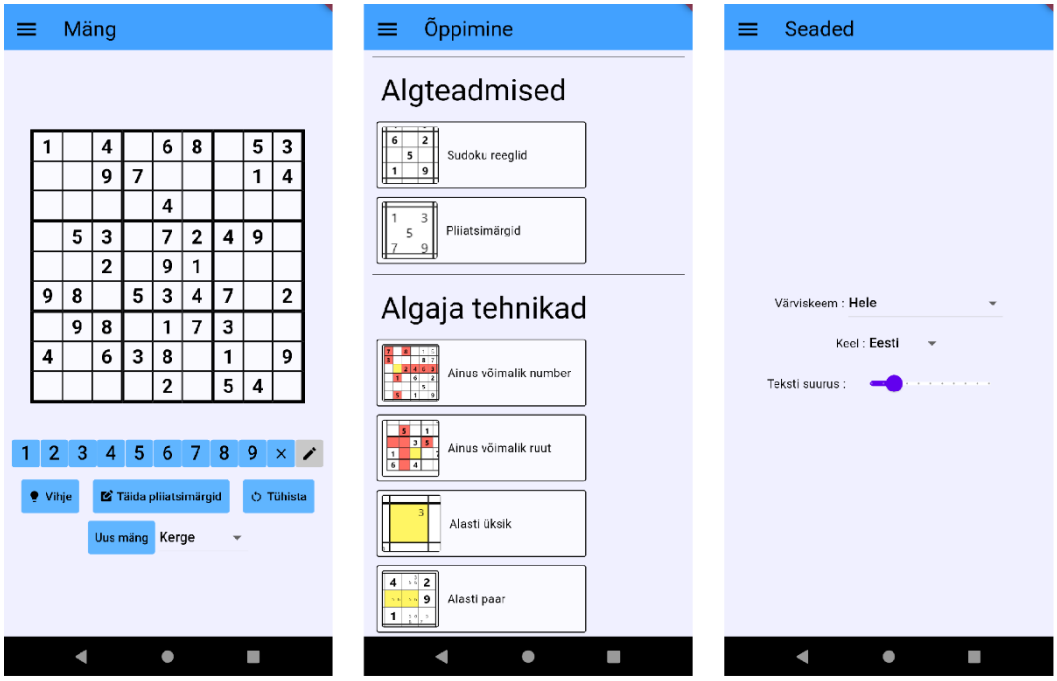

Joonis 14. Serialiseeritud mall-Sudoku näide

4.5.6 Lõpliku Sudoku genereerimine

Rakenduse käivitamisel või kui kasutaja soovib alustada uut mängu võetakse JSON failidest üks juhuslik mall-Sudoku nõutud raskusastmega, deserialiseeritakse seda *SudokuTemplate* objektiks, rakendatakse juhuslikult kolm permutatsiooni ning tulemuseks saadakse lõplikku Sudoku, mida antakse kasutajale lahendamiseks.

4.6 Vaadete struktuur

Rakenduses on kolm peamist vaadet, mida võib näha Joonisel 15 – mängu vaade, õppimismooduli vaade ja seadete vaade .



(a) (b) (c)

Joonis 15. Loodud rakenduse vaated: (a) mängu vaade, (b) õppimismooduli vaade, (c) seadete vaade

Mängu vaates saab kasutaja lahendada Sudoku. See koosneb Sudoku ruudustikust ja kahest paneelist nuppudega. Esimene võimaldab sisestada rudustikku ruutude väärtused ja pliiatsimärgid. Teine paneel võimaldab saada vihjeid, täita kõik pliiatsimärgid korraga, tühistada antud Sudoku progressi või genereerida uus Sudoku. Õppemooduli vaatesse on koondatud kõik õppematerjalid, mis on jaotatud kategooriatesse raskuse järgi. Iga materjali kaardile vajutamisel avaneb materjali lehekülg, kus kasutaja võib õppematerjaliga tutvuda ning lahendada soovi korral ülesandeid. Seadete leheküljel on kasutajal võimalus muuta värviskeemi, keelt ja õppematerjalide teksti suurust.

4.7 Käitusajal laetavad failid

Rakenduses on olemas hulk faile, mida kasutatakse käitusajal. Suurem osa nendest failidest on pildid, mida näidatakse õppematerjalides. Käitusajal laetakse ka mall-Sudoku, mida hoitakse JSON failides.

Flutteris on olemas sisseehitatud lihtne viis lisada pilte ja JSON faile rakendusse. Selleks peab neid lisama *pubspec.yaml* manifestis *flutter* sektsioonis *assets* alamsektsiooni. Seejärel võib neid laadida koodis *AssetBundle* klassi abil [26]. Joonisel 16 on näidatud osa *assets* alamsektsioonist loodavas rakenduses.

```
assets:
  - lib/game_internals/generator/generated_templates/easy.json
  - lib/game_internals/generator/generated_templates/medium.json
  - lib/game_internals/generator/generated_templates/hard.json
  - lib/game_internals/generator/generated_templates/very_hard.json
  - lib/assets/
  - lib/assets/tileListIcons/
  - lib/assets/techniques/
  - lib/assets/techniques/sudoku_rules/
```

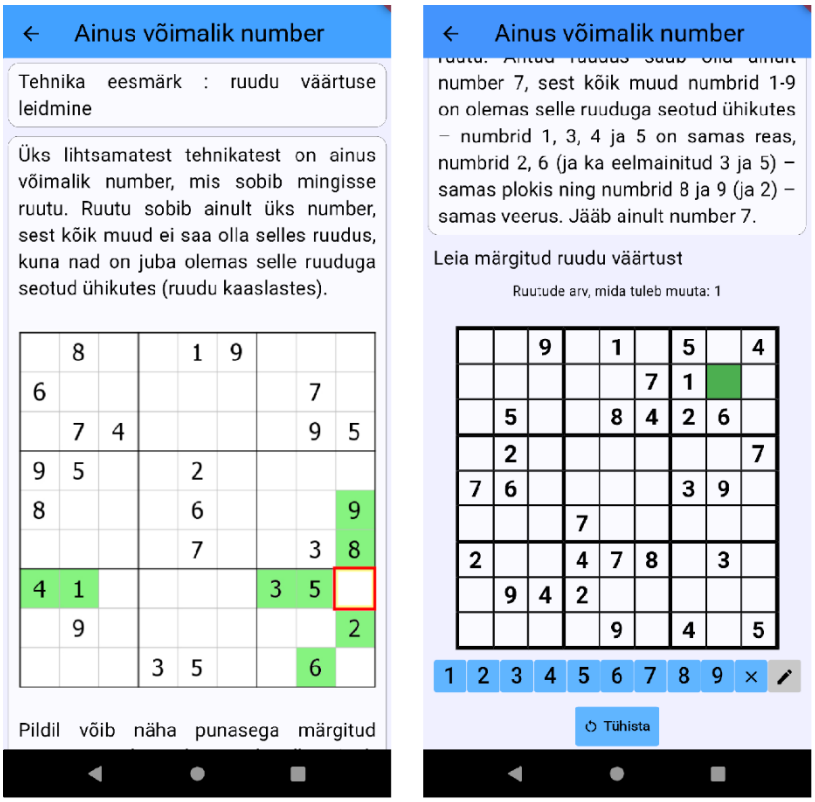
Joonis 16. Assets alamsektsiooni osa rakenduse manifestis

4.8 Õppematerjalid

Õppematerjalide tekst on koostatud eesti keeles (Lisa 3) ning tõlgitud inglise ja vene keeltesse. Selle teksti alusel on loodud tehnikate kirjeldused rakenduses. Peale teksti sisaldavad õppematerjalid pilte näidetega ja ülesandeid. Selleks, et lihtsustada õppematerjalide vaadete täitmist sisuga, viia miinimumini korduv kood ja ühtsustada õppematerjalide väljanägemist, on loodud meetodid, mis loovad õppematerjalide vidinaid teksti, piltide ja ülesannetega ning meetod, mis võimaldab ühendada neid komponente sektsioonidesse. Joonisel 17 on näidatud „Ainus võimalik number“ tehnika kirjeldus ning sellega seotud ülesanne.

Ülesanded kujutavad endast Sudoku ruudustikku, kus on antud mingi algseis ning kasutajal tuleb midagi teha ülesande lahendamiseks, näiteks rakendada mingit tehnikat, et leida märgitud ruudu väärtust. Ülesanded ei nõua terve Sudoku lahendamist, vaid on mõeldud isoleeritud tegevuste harjutamiseks.

Näidete loomiseks (Lisa 3) kasutati HoDoKu rakendus.



Joonis 17. Õppematerjali lehekülg: (a) kirjeldus, (b) ülesanne

4.9 Isikupärastamine

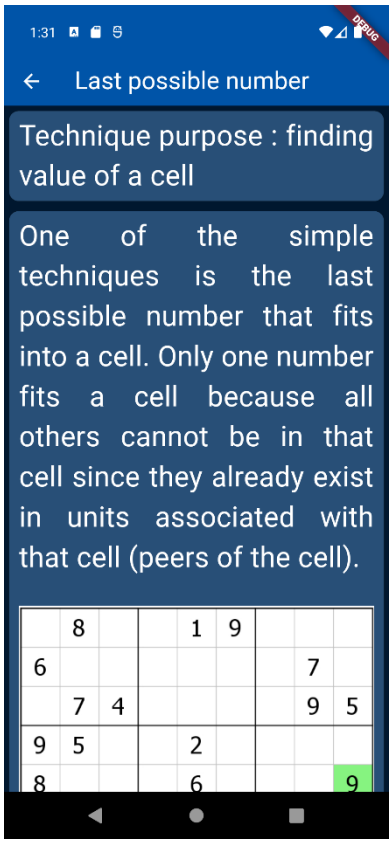
Rakenduse seadeid salvestatakse igal seadmel eraldi. Seadete haldamiseks on loodud klass *SharedPreferencesManager*, milles kasutatakse seadete salvestamiseks ja pärimiseks *shared_preferences* teeki [27].

4.9.1 Värviskeemid ja ligipääsetavus

Rakenduses on olemas võimalus valida ühe kolmest värviskeemist – hele, tume ja kõrge kontrastsusega. Joonisel 17 on näidatud õppematerjali vaade heleda värviskeemiga, aga Joonisel 18 võib näha sama vaadet tumeda värviskeemiga (samuti on seal vahetatud keel ja teksti suurus).

Värvide valimisel lähtus autor sellest, et rakenduse ligipääsetavus peab olema kõrgel tasemel. Kuigi antud rakendus ei ole veebirakendus, võib teksti ja tausta kontrastsuse suhet hinnata WCAG (*Web Content Accessibility Guidelines*) standardi järgi, sest värve näevad inimesed samamoodi nii veebirakenduses, mobiilsetes rakendustes kui ka töölauarakendustes.

Heleda ja tumeda värviskeemide teksti ja tausta kontrastsuse suhe (ruudustik, õppematerjalid, nupud) vastab alati vähemalt WCAG 2.1 AA tasemele – 4.5:1, enamasti vastab see ka AAA tasemele – 7:1 [28]. Kõrge kontrastsusega värviskeem vastab täiesti AAA tasemele.



Joonis 18. Õppematerjali vaade muudetud seadetega

4.9.2 Keeled

Rakendus toetab lokalisatsiooni – seda on võimalik kasutada eesti, inglise ja vene keeles. Lokalisatsioon on toetatud *flutter_localizations* teegi abil. Teegi kasutamiseks on seda vaja lisada rakenduse manifesti, luua selle konfiguratsioonifail ning lisada *l10n* kausta *.arb* faililaiendiga failid vastavalt sellele, millised keeled on toetatud. Nendes failides hoitakse sõnede tõlked keeltesse formaadis, mis on sarnane JSON-iga. Teegis on olemas käsk *flutter gen-l10n*, mis genereerib *.arb* failide alusel klasse, mida võib hiljem koodis kasutada [29].

Joonisel 17 võib näha materjali teksti eesti keeles ning Joonisel 18 on sama materjali tekst inglise keeles.

4.9.3 Teksti suurus õppematerjalides

Kuna õppematerjalid sisaldavad suure hulga teksti, on rakenduses olemas võimalus muuta teksti suurust kuni kaks korda suuremaks. Joonistel 17 ja 18 on näidatud sama õppematerjali tekst, aga Joonisel 18 on tekst märgatavalt suurem.

4.10 Kohanduv kasutajaliides

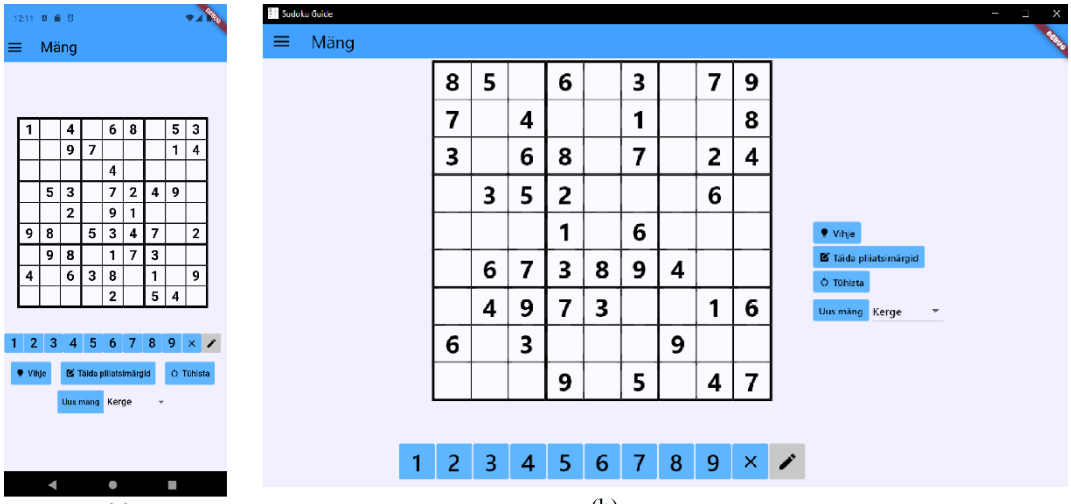
Kuna antud rakendust saab kasutada nii telefonis kui ka arvutis, on oluline, et rakendus suudab töötada erinevate ekraani suuruste ja asenditega, sest telefoni on võimalik pöörata ja Windowsis võib muuta rakenduse akna suurust. Flutter võimaldab luua kasutajaliidest, mis reageerib ekraani suuruse ja asendi muutustele, mis on loodavas rakenduses aktiivselt kasutatud. Joonisel 19 võib näha mängu vaadet telefoni horisontaalses asendis – üks nuppudega paneel liikus ruudustikust paremale, et jätta rohkem ruumi ruudustikule. Samuti muutub rakenduses sõltuvalt ekraani suurusest ja asendist Sudoku ruudustiku, nuppude ja piltide suurus ning tehnikate kaartide paigutus õppemooduli vaates.

Joonis 19. Mängu vaade telefoni horisontaalses asendis

4.11 Platvormiülesus

Loodava rakenduse mõned stiilid sõltuvad platovrmist ning seda kontrollitakse koodis, aga kõik töötab Windows ja Android platvormidel ühel koodipõhjal. Flutter toetab erinevatel platvormidel erinevaid sisendi tüüpe, näiteks Androidis toimub lehekülje

kerimine ekraani lohistamise abil, aga Windowsis saab seda teha hiireratta abil. Joonisel 20 on toodud mängu vaade telefonis ja töölauarakenduses.



(a) (b)

Joonis 20. Mänguvaade: (a) telefonis, (b) arvutis

4.12 Testimine

Suurem osa testimisest toimub rakenduse arenduse ajal ning testimine on manuaalne. Iga uue funkntsionaalsuse lisamisele järgneb selle testimine, võimalusel erinevate sisenditega. Aeg-ajalt tehakse nn *smoke testing*, mille käigus kontrollitakse, et rakenduse peamised osad ja funktsionaalsused töötavad korrektselt. Sellisel juhul kontrollitakse järgmiseid funktsioone:

- Sudoku genereerimine mall-Sudoku alusel,
- ruutude valimine ruudustikus,
- väärtuste ja pliiatsimärkide sisestamine ruutudesse,
- kõikide pliiatsimärkide täitmine rakenduse poolt,
- mängu tühistamine,
- vihjete andmine,
- vigade näitamine Sudoku ruudustikus,
- värviskeemi, keele ja teksti suuruse vahetus,

- ülesannete korrektne töötamine,
- kohanduv kasutajaliides – erinevad ekraani suurused ja asendid.

Samuti oli rakendus antud testimiseks 8 inimesele, kes proovisid rakendust kasutada ning andsid selle kohta tagasisidet, mis oli enamuses positiivne. Testimise protsessis ilmusid testijatel mõned ideed, kuidas võiks rakendust paremaks teha. Need ideed on kirjeldatud peatükis **Edasiarendus**.

5 Hinnang loodud rakendusele

Lõputöö raames arendatud rakendust võib hinnata saadud rakendust nõuetele vastavuse määra alusel. Lisaks võib kirjeldada rakenduse võimalikku edasiarendust, sest nii autoril kui ka rakenduse testijatel on ideid, kuidas saab rakendust paremaks teha.

5.1 Saavutatud kasutatavus

Nõuded, mis on kirjeldatud kasutajalugudena on täidetud rakenduse arenduse käigus, mis toimus paralleelselt lõputöö kirjutamisega.

Rakenduses töötab mängu vaade, kus kasutaja saab lahendada erineva raskusega rakenduse poolt genereeritud Sudoku. Rakendus saab anda kasutajale vihjeid Sudoku lahendamiseks. Samuti töötab õppimise moodul, kus kasutaja saab lugeda materjale ning lahendada ülesandeid.

Seadete vaates saab kasutaja rakendust personaliseerida - muuta värviskeemi, keelt ja teksti suurust õppimise materjalides. Valitud seadeid salvestatakse.

Rakendus suudab genereerida mall-Sudoku alusel Sudoku mõistatusi ning saab genereerida ka mall-Sudoku, kuid seda peab käivitama eraldi, kasutajaliideses puudub hetkel võimalus käivitada mall-Sudoku genereerimist.

Põhjuseks on see, et mall-Sudoku genereerimine on resurssimahukas protsess ning võib võtta palju aega. Alternatiivne lähenemine on kirjeldatud peatükis **Edasiarendus**. Samas peab kasutajal piisama nendest Sudoku mõistatustest, mida on võimalik genereerida rakenduses olevate mall-Sudoku alusel.

Nagu oli enne kirjutatud, saab rakendus ühe mall-Sudoku põhjal genereerida mitu miljonit Sudoku varianti.

Rakendust on võimalik kasutada nii Windows kui ka Android platvormidel, kõik funktsionaalsus töötab mõlemal platvormil.

Rakendus on hetkel kasutatav iseseisva rakendusena ning saab täita oma peamist eesmärki - abistada kasutajat Sudoku lahendamise õppimises.

5.2 Edasiarendus

Rakenduse edasisel arendamisel võib teha paremaks mall-Sudoku genereerimist - luua tagarakendus ning genereerida mall-Sudoku seal. Töös loodud rakendus muutub sel juhul eesrakenduseks ning hakkab vajadusel pärima mall-Sudoku tagarakendusest. Nii saab võimaldada uute mall-Sudoku pidevat genereerimist ja kasutamist rakenduses.

Veel võib rakendusse lisada progressi jälgimist, et kasutaja saaks näha, milliseid tehnikaid ta on juba õppinud ja mis ülesandeid lahendanud.

Samuti saab rakendusse lisada rohkem Sudoku lahendamise tehnikaid, et rakendus suudaks lahendada kõige raskemaid Sudoku.

Rakendusse võib lisada rohkem toetatud keeli, et suurem hulk inimesi saaks rakendust kasutada.

Flutter raamistik toetab ka teisi platvorme, nagu Linux, iOS, macOS ja veeb [30] ja selleks, et suurendada võimalike kasutajate arvu, võib rakendust peale testimist lasta välja ka nendel platvormidel.

Mõned funktsioonid olid küsitud inimeste poolt, kes rakendust testisid:

- võimalus vajutada vihjele, et kohe avada vastava tehnika materjali;
- võimalus sisestada oma Sudoku, et rakendus aitaks selle lahendamisega;
- "Undo" nupp - võimalus tühistada viimane muutus ruudustikus;
- mängu salvestamine rakenduse sulgemisel, et seda saaks hiljem jätkata.

Flutter raamistikku pidevalt arendatakse, mistõttu rakenduse edasiarendus eeldab ka raamistiku uuendamist.

6 Kokkuvõte

Bakalaureusetöö käigus oli loodud Sudoku Guide rakendus. Rakendus suudab täita oma peamist eesmärki - abistada kasutajat Sudoku lahendamise õppimisel. Rakendus saab genereerida ja lahendada erineva raskusega Sudoku ja anda kasutajale vihjeid. Samuti on rakenduses olemas õppematerjalid ja ülesanded, mille abil saab kasutaja õppida. Rakendust on võimalik seadistada oma eelistuste järgi ja kasutada Android ja Windows platvormidel. Rakendus on arendatud Flutter raamistikus.

Antud rakendus aitab muuta Sudoku lahendamise tehnikate õppimist mugavamaks, koondades erinevate tehnikate kirjeldused ühte kohta. Ülesannete lahendamine võimaldab saadud teadmisi rakendada praktikas, mis aitab kaasa teadmiste meeldejäämisele. Tehnikatel põhinevad vihjed stimuleerivad kasutajat pidevalt õppima ja rakendama õpituid tehnikaid, viies õppimise ja lahendamise protsessid kokku.

Kasutatud kirjandus

[1] Ashlesh P, Deepak KK, Preet KK. Role of prefrontal cortex during Sudoku task: fNIRS study. Transl Neurosci. 2020 Nov 3;11(1):419-427. doi: 10.1515/tnsci-2020-0147. PMID: 33335780; PMCID: PMC7718610. (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7718610/#j_tnsci-2020-0147_ref_004)

[2] Litwin H, Schwartz E, Damri N. Cognitively Stimulating Leisure Activity and Subsequent Cognitive Function: A SHARE-based Analysis. Gerontologist. 2017 Oct 1;57(5):940-948. doi: 10.1093/geront/gnw084. PMID: 27117305; PMCID: PMC5881687. (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5881687/)

[3] J. Pinkney, „The Effects of Sudoku on Mathematical Problem Solving Ability in College Students“, studylib.net. [Võrgumaterjal] https://studylib.net/doc/7029180/the-effects-of-sudoku-on-mathematical-problem-solving-abi [Kasutatud 02.11.2022]

[4] G. Yeoh, „If School Made You HATE Math, Sudoku Will Teach You to Love It“, ricemedia.co. [Võrgumaterjal] https://www.ricemedia.co/culture-life-school-made-hate-math-sudoku-will-teach-love/ [Kasutatud 02.11.2022]

[5] B. Dean, „We analyzed 4 million Google Search Results. Here's What We Learned About Organic Click Through Rate“, backlinko.com. [Võrgumaterjal] https://backlinko.com/google-ctr-stats [Kasutatud 04.10.2022]

[6] „Sudoku techniques“, conceptispuzzles.com. [Võrgumaterjal] https://www.conceptispuzzles.com/index.aspx?uri=puzzle/sudoku/techniques [Kasutatud 04.10.2022]

[7] „Tips on Solving Sudoku Puzzles - Sudoku Solving Techniques“, kristanix.com. [Võrgumaterjal] https://www.kristanix.com/sudokuepic/sudoku-solving-techniques.php [Kasutatud 04.10.2022]

[8] „Sudoku rules“, sudoku.com. [Võrgumaterjal] https://sudoku.com/sudoku-rules/ [Kasutatud 04.10.2022]

[9] “HoDoKu”, hodoku.sourceforge.net. [Võrgumaterjal] https://hodoku.sourceforge.net/en/index.php [Kasutatud 31.10.2022]

[10] H. Dhaduk, „React Native vs Ionic: Comparing Performance, User Experience, and much more!“, simform.com. [Võrgumaterjal] https://www.simform.com/blog/react-native-vs-ionic/ [Kasutatud 22.11.2022]

[11] Gordon H. „macOS Performance Comparison: Flutter Desktop vs. Electron“, getstream.io. [Võrgumaterjal] https://getstream.io/blog/flutter-desktop-vs-electron/ [Kasutatud 22.11.2022]

[12] „Cross-platform mobile frameworks used by software developers worldwide from 2019 to 2021“, statista.com. [Võrgumaterjal] https://www.statista.com/statistics/869224/worldwide-software-developer-working-hours/ [Kasutatud 22.11.2022]

[13] P. Barnhate, „Cordova vs React Native: Choosing the Best Framework for Your App“, mobisoftinfotech.com. [Võrgumaterjal] https://mobisoftinfotech.com/resources/blog/cordova-vs-react-native-comparison/ [Kasutatud 22.11.2022]

[14] „Flutter VS Apache Cordova: Choosing the Best Technology for Cross-Platform Development“, surf.dev. [Võrgumaterjal] https://surf.dev/flutter-vs-apache-cordova/ [Kasutatud 22.11.2022]

[15] F. Ferreira, „React Native vs Native for Mobile App Development“, scalablepath.com. [Võrgumaterjal] https://www.scalablepath.com/react-native/react-native-vs-native [Kasutatud 22.11.2022]

[16] C. Deshpande, „Flutter vs. React Native: Which One to Choose in 2023“, simplilearn.com. [Võrgumaterjal] https://www.simplilearn.com/tutorials/reactjs-tutorial/flutter-vs-react-native [Kasutatud 22.11.2022]

[17] „Introduction to Solar2D“, docs.coronalabs.com. [Võrgumaterjal] https://docs.coronalabs.com/guide/programming/intro/index.html [Kasutatud 15.02.2023]

[18] „Best Version Control Hosting Software“, g2.com. [Võrgumaterjal] https://www.g2.com/categories/version-control-hosting [Kasutatud 23.04.2023]

[19] A. Kumar KP, „Github vs Gitlab vs Bitbucket“, disbug.io. [Võrgumaterjal] https://disbug.io/en/blog/github-vs-gitlab-vs-bitbucket [Kasutatud 23.04.2023]

[20] „6 Best Flutter IDE & Text Editors for App Development“, bairesdev.com. [Võrgumaterjal] https://www.bairesdev.com/blog/best-flutter-ide-text-editors/ [Kasutatud 23.04.2023]

[21] „Windows install“, docs.flutter.dev. [Võrgumaterjal] https://docs.flutter.dev/get-started/install/windows [Kasutatud 10.05.2023]

[22] „Constraint Solving via Backtracking“, ilyasergey.net. [Võrgumaterjal] https://ilyasergey.net/YSC2229/week-09-backtracking.html [Kasutatud 03.10.2022]

[23] R. Haabu, “Sudoku lahendaja optimeerimine kiirusele”, 2019. Kättesaadav: https://digikogu.taltech.ee/et/Item/de193213-37a9-4009-a284-6d5f8cc33793

[24] „Dart Interface“, darttutorial.org. [Võrgumaterjal] https://www.darttutorial.org/dart-tutorial/dart-interface/ [Kasutatud 11.05.2023]

[25] „Program for Sudoku Generator“, geeksforgeeks.org. [Võrguamterjal] https://www.geeksforgeeks.org/program-sudoku-generator/ [Kasutatud 11.04.2023]

[26] „AssetBundle class“, api.flutter.dev. [Võrgumaterjal] https://api.flutter.dev/flutter/services/AssetBundle-class.html [Kasutatud 11.05.2023]

[27] „Shared preferences plugin“, pub.dev. [Võrgumaterjal] https://pub.dev/packages/shared_preferences [Kasutatud 16.04.2023]

[28] „Web Content Accessibility Guidelines (WCAG) 2.1“, w3.org. [Võrgumaterjal] https://www.w3.org/TR/WCAG21/#contrast-minimum [Kasutatud 11.05.2023]

[29] „Internationalizing Flutter apps“, docs.flutter.dev. [Võrgumaterjal] https://docs.flutter.dev/accessibility-and-localization/internationalization [Kasutatud 13.03.2023]

[30] „Supported deployment platforms“, docs.flutter.dev. [Võrgumaterjal] https://docs.flutter.dev/reference/supported-platforms [Kasutatud 11.05.2023]

Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks1

Mina, Ilja Karpenko

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose „Platvormiülese rakenduse arendus Sudoku lahendamise tehnikate õppimiseks“, mille juhendaja on Einar Kivisalu
   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.

15.05.2023

1 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 loomingulise 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 – Rakenduse versioonihaldus

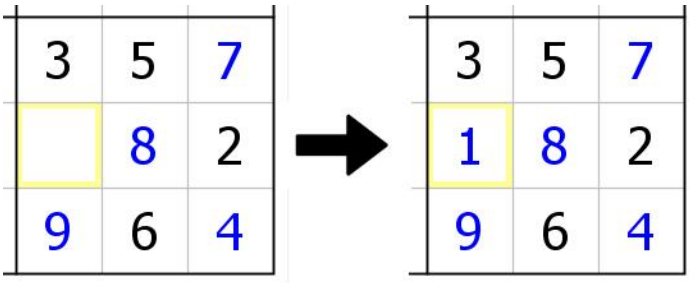
https://github.com/Majuo/thesis

Lisa 3 – Sudoku lahendamise tehnikate kirjeldus

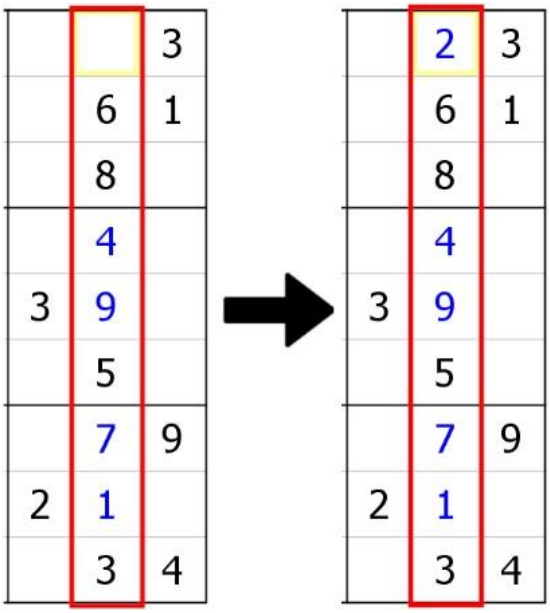
Sudoku reeglid

Kõige lihtsam on Sudoku lahendamine reeglite järgi. Selle tehnika eesmärk on ühe ruudu väärtuse väärtuse leidmine. Tehnikat saab rakendada, kui mingis reas, veerus või plokis on jäänud ainult 1 vaba ruut. Sellesse ruutu saab vastavalt reeglitele panna puuduv number. Sellist tehnikat nimetatakse mõnikord inglise keeles *Full House*.

Piltidel võib näha, kuidas lahendatakse Sudoku reeglite abil. Esimesel pildil puudub plokist ainult number 1, seepärast ongi selle ruudu lahendus 1. Teisel pildil on punasega märgitud veerus olemas kõik numbrid peale 2, mis tähendab, et tühjas ruudus peab olema 2. Sama loogika kehtib ka ridade puhul.



Lahendamine reeglite abil ploki näitel

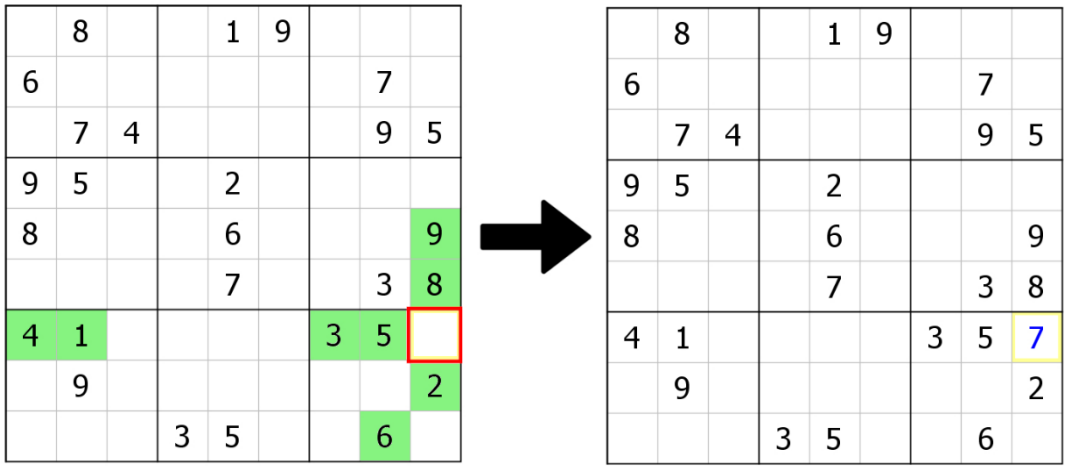


Lahendamine reeglite abil veeru näitel

Ainus võimalik number

Üks lihtsamatest tehnikatest on ainus võimalik number, mis sobib mingisse ruudusse.

Tehnika eesmärk on leida ruudu väärtus. Ruutu sobib ainult üks number, kui kõik muud numbrid ei saa selles ruudus olla, kuna nad on juba olemas selle ruuduga seotud ühikutes.



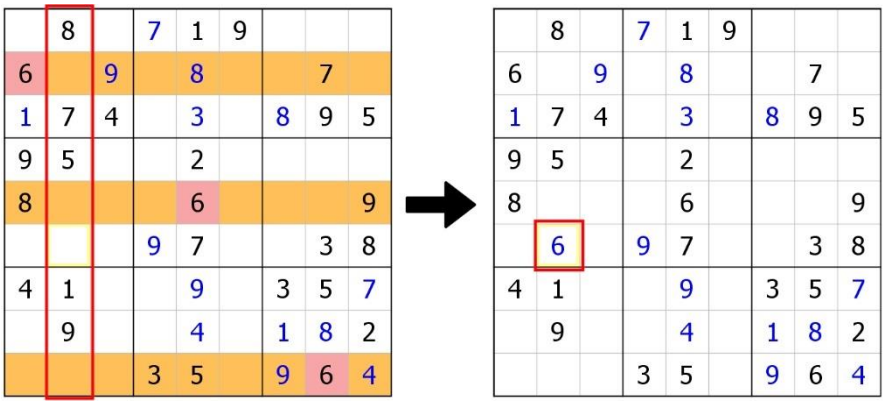
Ainsa võimaliku numbri tehnika näide

Pildil võib näha punasega märgitud ruutu. Antud ruudus saab olla ainult number 7, sest kõik muud numbrid ühest üheksani on olemas selle ruuduga seotud ühikutes – numbrid 1, 3, 4 ja 5 on samas reas, numbrid 2, 6 (ja ka eelmainitud 3 ja 5) – samas plokis ning numbrid 8 ja 9 (ja 2) – samas veerus. Jääb ainult number 7.’

Ainus võimalik ruut

Tehnika eesmärk on leida ruudu väärtus.

Mõnikord juhtub nii, et plokis, reas või veerus on mitu vaba kohta, kuid mingi numbri saab panna ainult ühte ruutu, kuna ta ei saa olla üheski teises ruudus.

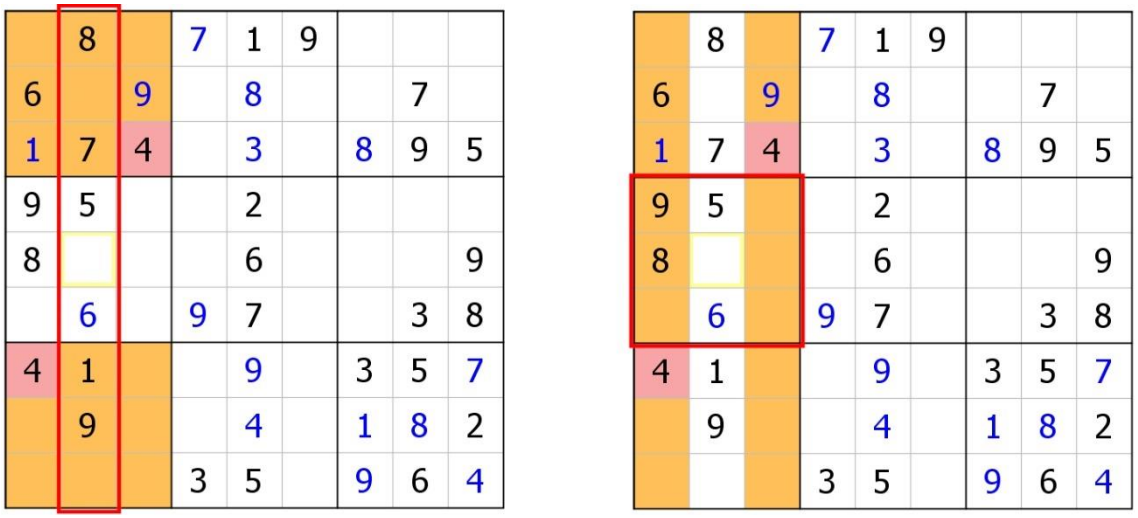


Ainsa võimaliku ruudu näide

Pildil võib näha, et punasega märgitud veerus on puudu mitu numbrit, sealhulgas number 6. Veerus on 4 tühja kohta, kuid 6 saab panna ainult ühte nendest, sest kõikides teistes kohtades rikub see Sudoku reegleid. Seetõttu asub number 6 kuuendas reas.

Sellel tehnikal on olemas variatsioonid. Allpool oleval pidil (vasakul) on üks veerg märgitud punase värviga. Selles veerus puudub number 4 ning selle jaoks on ainult üks sobiv koht – 5. reas. Vabad ruudud 2. ja 9. reas ei sobi, kuna nendes plokkides (värvitud oranžiga) on juba olemas number 4.

Sama olukorda võib vaadelda teisest küljest – pildil (paremal) on punasega märgitud plokk. Selles plokis ei ole 4 ja seda saab panna ainult ühte ruutu, sest muude vabade ruutude veergudes (värvitud oranžiga) on juba olemas number 4 (värvitud punasega).



Ainsa võimaliku ruudu tehnika variatsioonid

Alasti kandidaadid

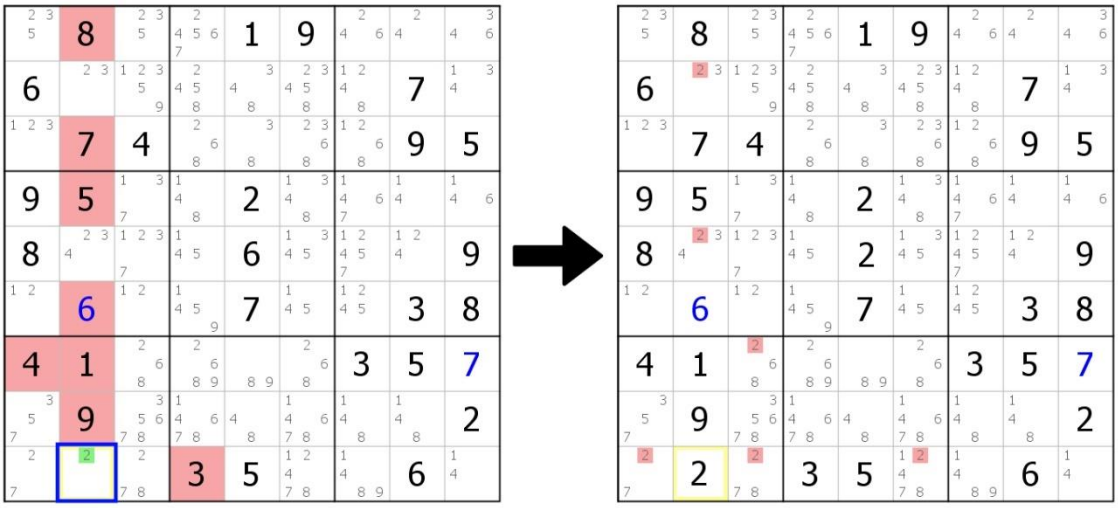
Alasti kandidaatide tehnikate eesmärk on kandidaatide eemaldamine ruutudest.

Alates sellest tehnikast tuleb Sudoku lahendamiseks kas jätta meelde, mis numbrid saavad olla ruutudes või kasutada pliiatsimärke, et märkida ruutude kandiaate.

Alasti kandidaadid (neid nimetatakse ka paljasteks ja ilmseteks) hõlmavad kohe mitu tehnikat, millel on sama põhimõte ning eesmärk – eemaldada ruutudest võimalikud kandidaadid.

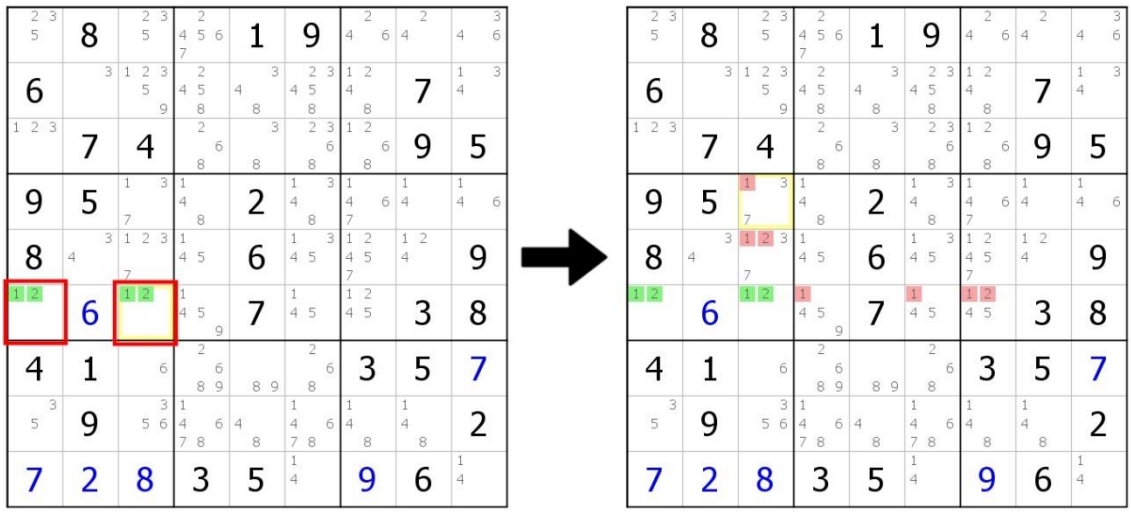
Alasti üksik – kui ruudus on täidetud kandidaadid ning selles ruudus on ainult üks kandidaat, siis ta ongi selle ruudu väärtus. Seda kandidaati saab eemaldada teistest ruutudest samas reas, veerus ja plokis.

Esimesel pildil võib näha, et sinisega märgitud ruudus on ainult üks kandidaat. See tähendab, et teised numbrid ruutu ei sobi, sest see rikuks reegleid. Kuna number 2 on ainus variant, võib seda panna kui ruudu lahendus ning eemaldada 2 märgitud ruudu kaaslaste kandidaatide hulgast – need on märgitud paremas ruudustikus punasega.



Alasti üksiku tehnika näide

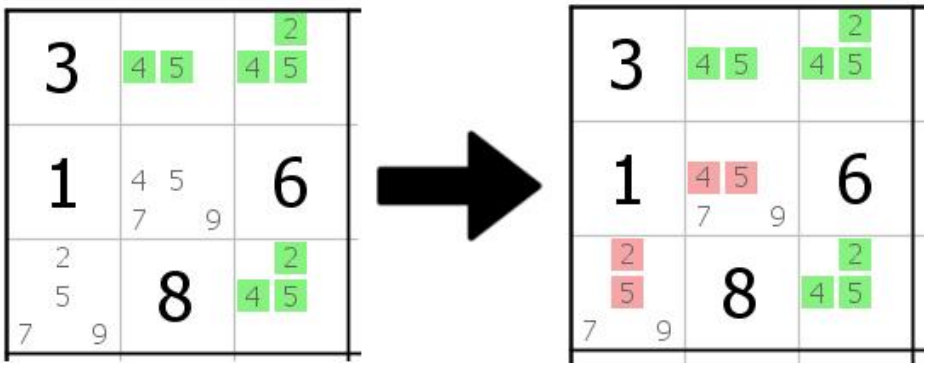
Alasti paar – kui on olemas kaks ruutu, mis asuvad ühes veerus, reas või plokis ning nendel ruutudel on ainult kaks võimalikku kandidaati, siis need numbrid sisalduvad antud ruutudes ning neid võib eemaldada teiste ruutude kandidaatide hulgast.



Alasti paari tehnika näide

Teisel pildil (vasakul) on märgitud punasega kaks ruutu – nendel on kaks võimalikku kandidaati, 1 ja 2. Ruudud asuvad samas plokis ja samas reas, seepärast saab rakendada alasti paari tehnikat. Võib väita, et numbrid üks ja kaks sisalduvad märgitud ruutudes, kuna nendes ei saa olla ükski teine number. Kuna numbrid ei saa korduda ühikutes, võimaldab see eemaldada mõned kandidaadid, mis on märgitud punasega teisel pildil (paremal). Kuna alasti paari sisaldavad ruudud asuvad ühes plokis, saab eemaldada kõikides muudes ploki ruutudes kandidaadid 1 ja 2. Sama võib teha 6. rea ruutudega, sest alasti paariga ruudud asuvad samas reas.

Alasti kolmik – põhimõte on sama kui paari juhul – kui kolm seotud ruutu sisaldavad ainult kolm võimalikku kandiaati, siis need numbrid asuvad antud ruutudes ning ei saa olla ruutude kaaslastes – neid võib eemaldada kaaslaste kandidaatide hulgast.



Alasti kolmiku tehnika näide

Kolmandal pildil on näidatud üks plokk, kus on olemas alasti kolmikku sisaldavad ruudud. Kandidaadid nendes ruutudes on 2, 4, 5, nad on märgitud rohelisega. Kuna kolmes ruudus saavad olla ainult need kolm numbrit, võib väita, et nad asuvadki nendes ruutudes. Selle pärast võib need numbrid eemaldada plokki teiste ruutude kandidaatide hulgast. Antud näites asub alasti kolmik ruutudes, mis on seotud ainult ühe ühikuga – plokiga, seepärast ei saa näides eemaldada kandidaate väljaspool ploki. Tähelepanu väärib ka see asjaolu, et kõik numbrid ei pea kindlasti sisalduma kõigis kolmes ruudus – näides sisaldab üks ruutudest ainult numbreid 4 ja 5.

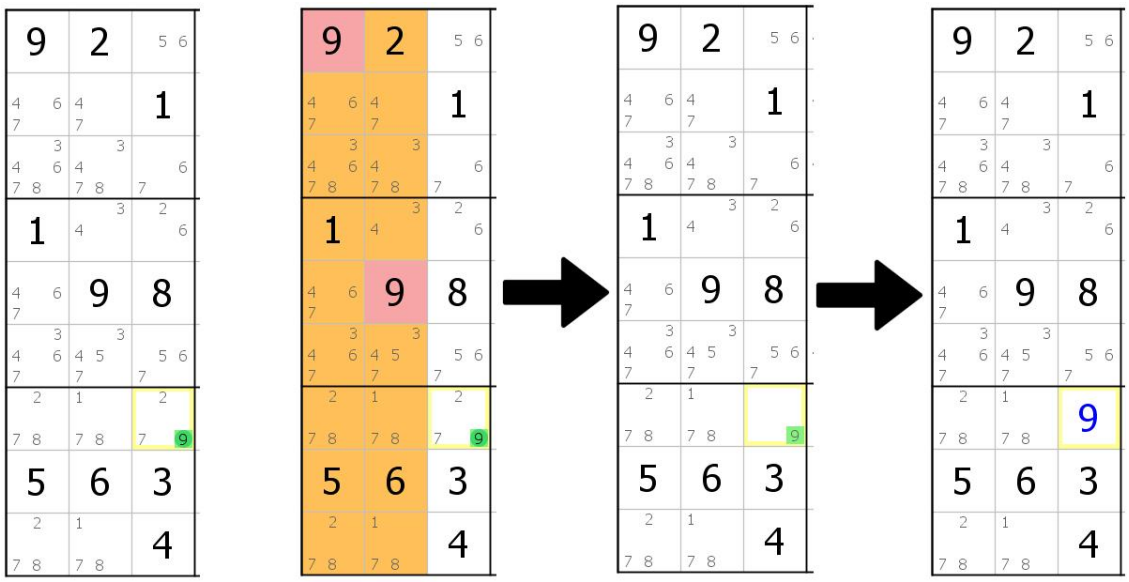
Sama põhimõtet võib rakendada ka suurema ruutude numbri puhul – peaasi, et võimalikke kandidaatide arv langeks kokku ruutude arvuga.

Peidetud kandidaadid

Peidetud kandidaatide tehnikate eesmärk on ruutude kandidaatide eemaldamine.

Peidetud kandidaadide tehnikate rakendamiseks kasutatakse samuti pliiatsimärke ning nad kuuluvad algtaseme tehnikate hulka.

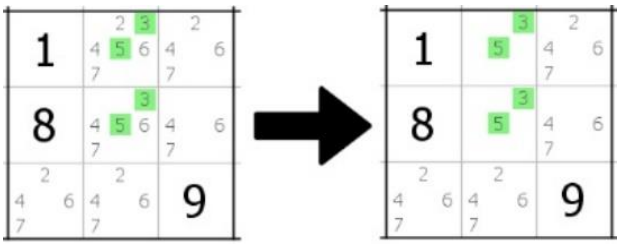
Peidetud üksik - kui ühikus on olemas ruut pliiatsimärgiga, mida pole mingis teises ühiku ruudus, siis see number peab olema selles ruudus, kuna see on ainus võimalik koht. Tehnika põhimõte on sarnane teise tehnikaga, „ainus võimalik number“, mis on kirjeldatud punktis 4.1.3, kuid siin kasutatakse tehnika rakendamiseks pliiatsimärke.



Peidetud üksiku tehnika näide

Esimesel pildil on rohelisega märgitud kandidaat, mis sisaldub alumises plokis ainult ühes ruudus. Teistes ploki ruutudes ei saa 9 olla, sest nendes veergudes on juba olemas 9. Kuna teisi variante ei ole, võib väita, et number 9 asub antud ruudus, seepärast võib sellest ruudust eemaldada teised kandidaadid ning panna lahenduseks 9.

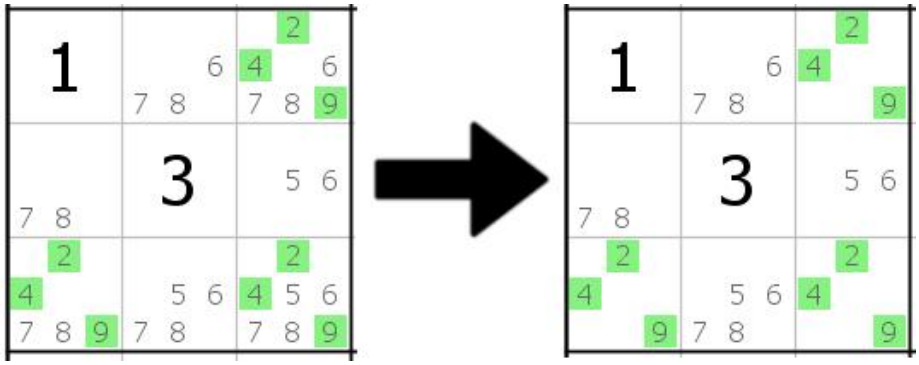
Peidetud paar on sarnane peidetud üksikuga, kuid sellel juhul on numbrite ja ruutude arv 2. Kui ühikus leiduvad kaks ruutu, kus on 2 pliiatsimärki, mida ei ole üheski teises ruudus, siis võib nendes ruutudes eemaldada kõik muud kandidaadid, kuna nendes ruutudes on 2 antud numbrit.



Peidetud paari tehnika näide

Teisel pildil on rohelisega märgitud kandidaadid 3 ja 5, mis sisalduvad antud plokis ainult kahes ruudus. Kuna need numbrid ei saa olla teistes ruutudes, võib väita, et nad sisalduvad antud ruutudes ja neist võib eemaldada kõik muud kandidaadid. Seejärel muutub peidetud paar alasti paariks ning seda võib kasutada edasi teiste kandidaatide eemaldamiseks.

Peidetud kolmik – põhimõte sama nagu peidetud paari puhul, ainult numbrite ja ruutude arv on 3. Kui ühikus on olemas kolm ruutu kolme pliiatsimärkidega, mis on ainult nendes ruutudes, siis võib nendes ruutudes eemaldada muud kandidaadid.



Peidetud kolmiku tehnika näide

Kolmandal pildil võib näha plokki, kus on kolmes ruudus märgitud rohelisega kandidaadid 2, 4 ja 9. Kasutades sama põhimõtet nagu peidetud paari puhul võib nendest ruutudest eemaldada kõik muud kandidaadid.

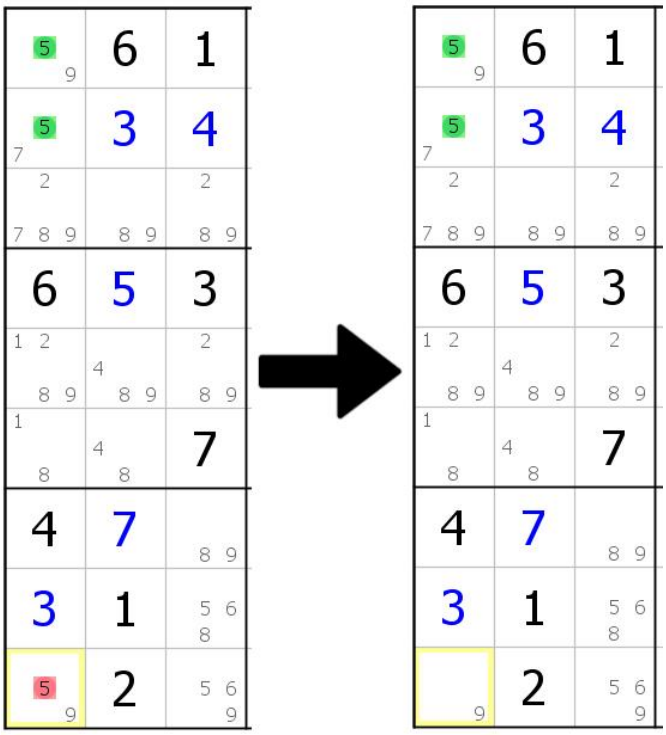
Sama põhimõte töötab ka suurema ruutude arvuga, aga selliseid variante võib kohata harva.

Osutavad paarid ja kolmikud

Osutava paari ja osutava kolmiku tehnikaid peetakse keskmise taseme lahendustehnikateks ning nende eesmärgiks on ruutude kandidaatide eemaldamine.

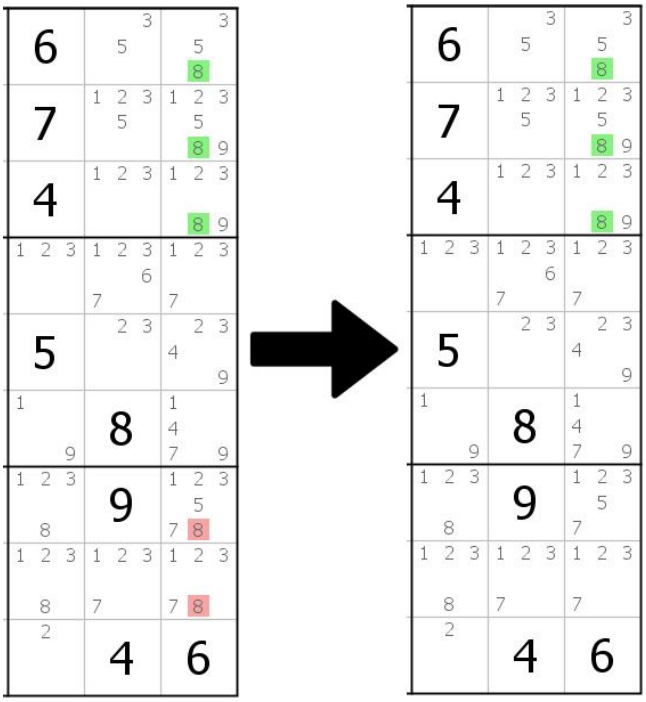
Osutava paari tehnikat saab kasutada, kui mingi pliitasimärk kohtub ühes plokis kaks korda ning mõlemad asukohad on samas reas või veerus. Kuna number peab esinema plokis 1 kord, asub see kindlasti ühes kahest antud ruudust. Seetõttu võib eemaldada see pliitasimärk teistest rea / veeru ruutudest.

Esimesel pildil on punase ja rohelisega märgitud ühe veeru kandidaadid. Kuna number 5 peab sisalduma ülemises plokis ja seal on ainult kaks sobivat ruudu, asub ta ühes nendest ruutudest. Sellest võib järeldada, et number 5 ei saa olla punasega märgitud kandidaadi sisaldavas ruudus.



Osutava paari tehnika näide

Osutava kolmiku tehnikal on sama põhimõte – kui mingi pliaatsimärk kohtub plokis kolm korda ning kõik ruudud asuvad ühes reas või veerus, siis võib see pliiatsimärk eemaldada teistest rea / veeru ruutudest. Osutava kolmiku näidet saab näha teisel pildil.

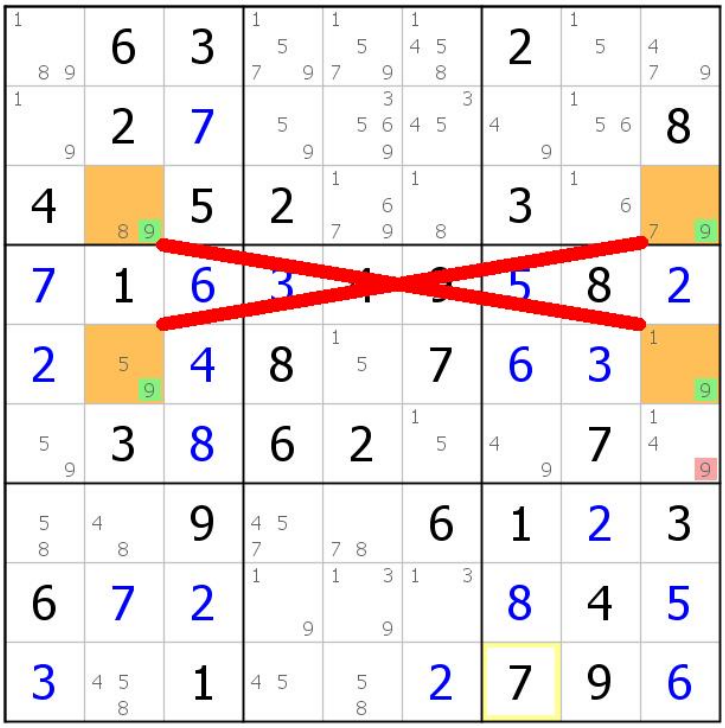


Osutava kolmiku tehnika näide

X-Wing

X-Wing tehnikat peetakse lihtsaimaks edasijõudnud tehnikate hulgas. Selle rakendamiseks tuleb kasutada pliiatsimärke ning selle eesmärgiks on kandidaatide eemaldamine.

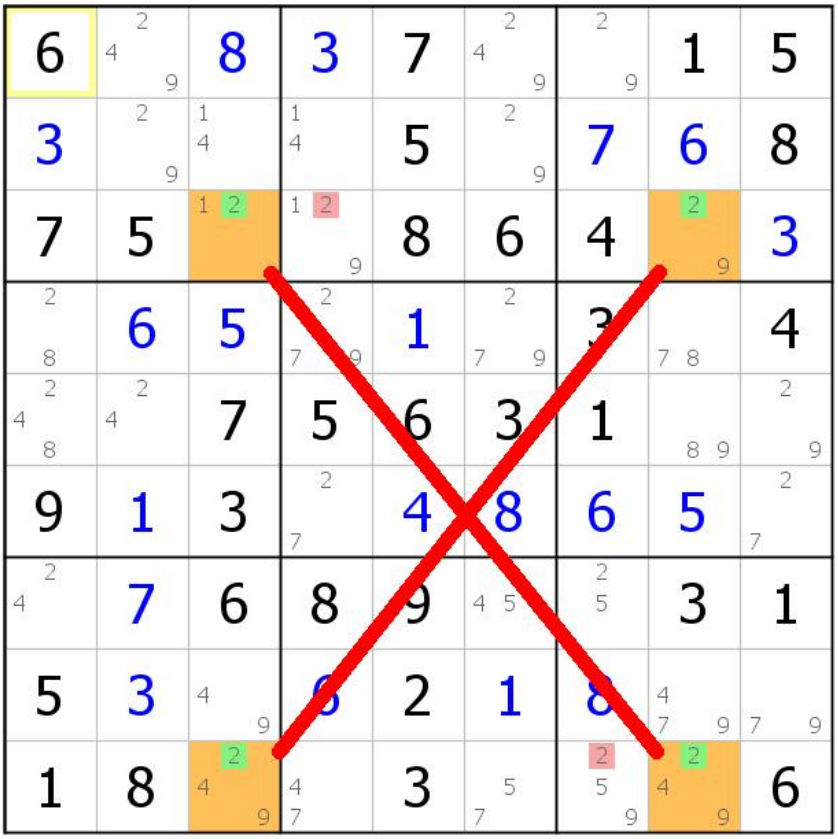
Selleks, et rakendada X-Wing tehnikat, on vaja leida kahes reas kaks ruutu (mõlemas reas), kus on olemas kandidaat, mida ei ole teistes ridade ruutudes. Ruudud mõlemas reas peavad olema samades veergudes. Võib kujutada ette, et ruudud moodustavad ristküliku, milles ruudud on ristküliku tipud. Pildil on need ruudud märgitud oranžiga, vaadeldatavaks kandidaadiks on 9.



X-Wing tehnika näide, kandidaatide eemaldamine veergudes

Tehnika seisneb selles, et antud 4 ruudus peab kindlasti esinema kandidaat 2 korda – üks kord mõlemas reas ning samas asuvad need numbrid „ristküliku“ vastastes tipudes ehk ristküliku diagonaalidel, sest muidu rikuksid nad Sudoku põhilist reeglit – numbrid ei saa ühikus korduda. Igal juhul asub mõlemas veerus number 9 ühes oranžiga märgitud ruutudest. Seepärast saab seda kandidaadi eemaldada teistest veergude ruutudest.

Tehnikat saab kasutada ka teistpidi – kui kandidaadid on valitud kahes veerus ning kandidaatide eemaldamine toimub ridades. Sellise variatsiooni näide on toodud teisel pildil. Tulemusena eemaldatakse ridades 3 ja 9 punasega märgitud kandidaadid.



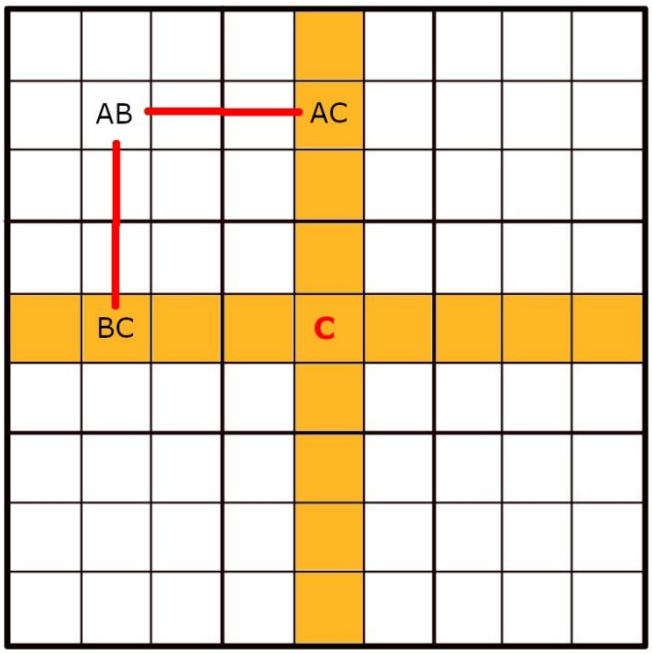
X-Wing tehnika näide, kandidaatide eemaldamine ridades

Y-Wing

Y-Wing on veel üks edasijõudnud tehnika, mille eesmärk on samuti eemaldada kandidaadid.

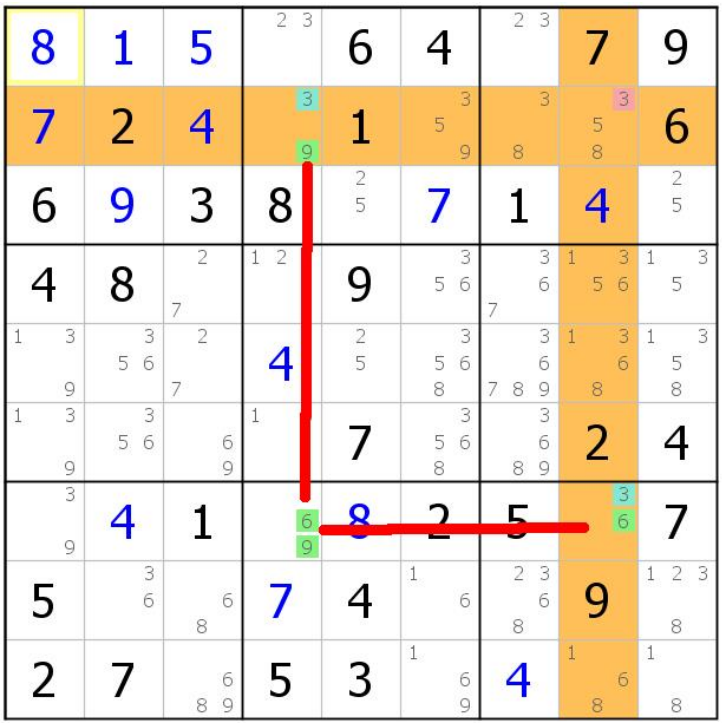
Erinevalt X-Wing tehnikast põhineb Y-Wing tehnika kolmel ruudul ja kolmel kandidaadil, neljas ruut on selle tehnika puhul ruut, kus saab eemaldada kandidaadi. See on kõige lihtsam Y-Wingi variant, aga on olemas ka teisi. Tehnika arusaamiseks on mõistlik alustada lihstast variandist.

Olgu A, B ja C kolm numbrit. Selleks, et saaks rakendada Y-Wing tehnikat, tuleb leida kolm ruudu, mis paiknevad nii, et moodustavad täisnurga (neid võib kujutada ette kui 3 ristküliku tipu). Kõigis kolmes ruudus peab olema täpselt kaks kandidaati. Täisnurga tipus olevas ruudus peavad olema kandidaadid AB, teistes ruutudes peavad olema kandidaadid AC ja BC (ehk number C ja üks numbritest A ja B). Üks võimalik ruutude paiknemise variant on toodud esimesel pildil.



Y-Wing tehnika, ruutude paiknemine

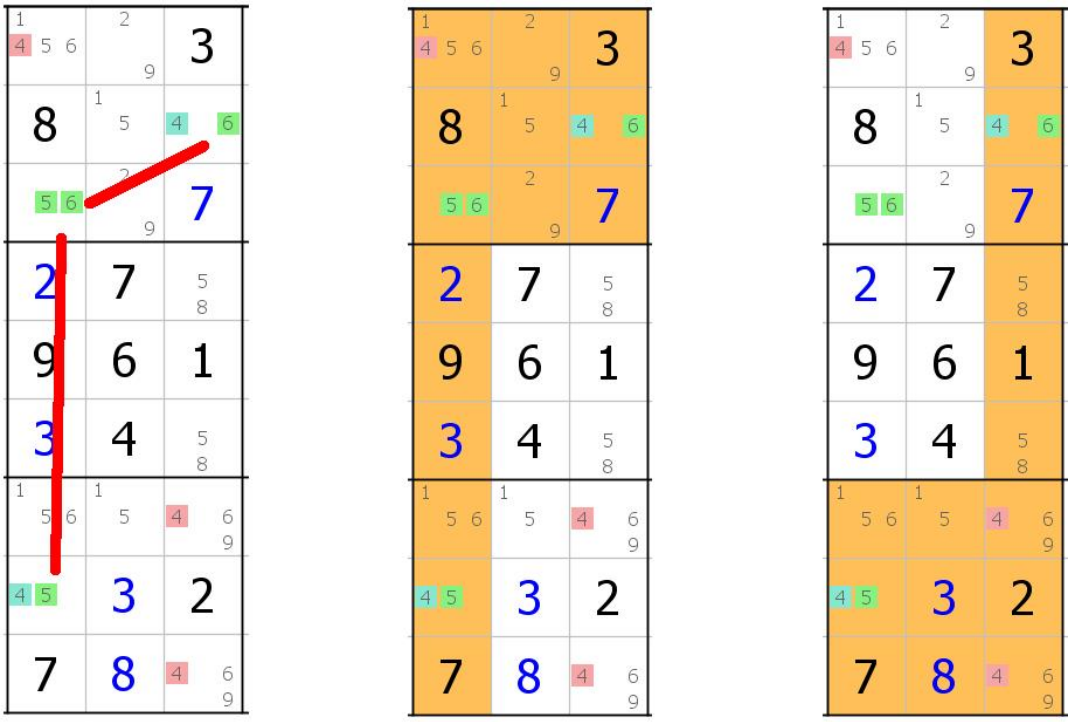
Kui ruudus AB on number A, siis ruudus AC on number C. Teiseslt poolt, kui ruudus AB on number B, siis ruudus BC on number C. Mõlemal juhul on number C kas ruudus AC või BC. Sellest võib järeldada, et ruudus, millega ristivad ruudud AC ja BC (ristküliku neljandas tipus) ei saa mingil juhul olla number C.



Y-Wing tehnika näide

Teisel pildil on toodud Y-Wing tehnika lihtsa variatsiooni näide. Selles näites eemaldatakse kandidaat 3 ruudust mis asub oranžiga värvitud veergude ristpunktis.

Y-Wing tehnikal on olemas ka teised variatsioonid. Kõige lihstamas variatsioonis oli üks ruut seotud keskmise (AB) ruuduga rea järgi, teine ruut – veeru järgi, aga see ei pea kindlasti olema nii. On olemas selline variant, kus üks ruutudest on seotud kesmise ruuduga ploki järgi – nad asuvad ühes plokis. Sellist varianti saab näha kolmandal pildil.



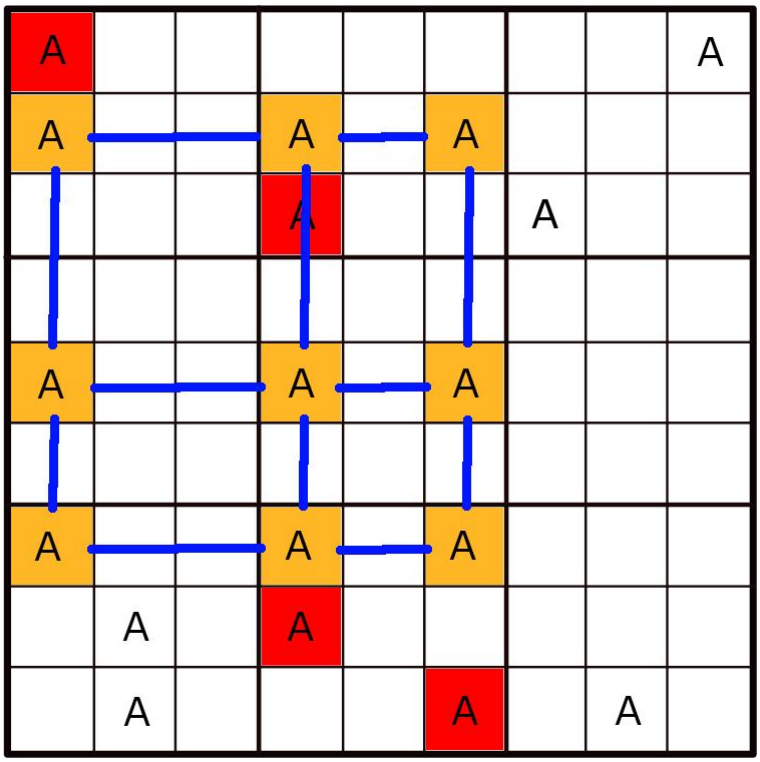
Y-Wing tehnika teine variatsioon

Sellises variatsioonis saab kandidaade eemaldada mitmes ruudus. Ülemises plokis saab punasega märgitud kandidaadi eemaldada, sest ta asub 1. veergu ja ülemise ploki ristimistsoonis. Teisisõnu on mõlemad Y-Wingi ruudud, kus saab olla 4, selle ruudu kaaslased ja üks neist sisaldab kindlasti numbri 4. Sama saab öelda ka alumises plokis asuvate punasega märgitud kandidaatide kohta. Seal ristivad alumine plokk ja 3. veerg.

Swordfish

Swordfish tehnika on põhimõtteliselt sarnane X-Wing tehnikaga, selle eesmärk on samuti kandidaatide eemaldamine. X-Wing tehnika puhul moodustasid ruudud 2x2 maatriksit, Swordfish tehnikas on see 3x3 maatriks. Tehnika rakendamiseks tuleb leida 3 rida, kus mingi number saab olla komes ruudus ja need ruudud paiknevad samades

veergudes. Ruutude paiknemise näidet saab näha esimesel pildil, ruudud on värvitud oranžiga.

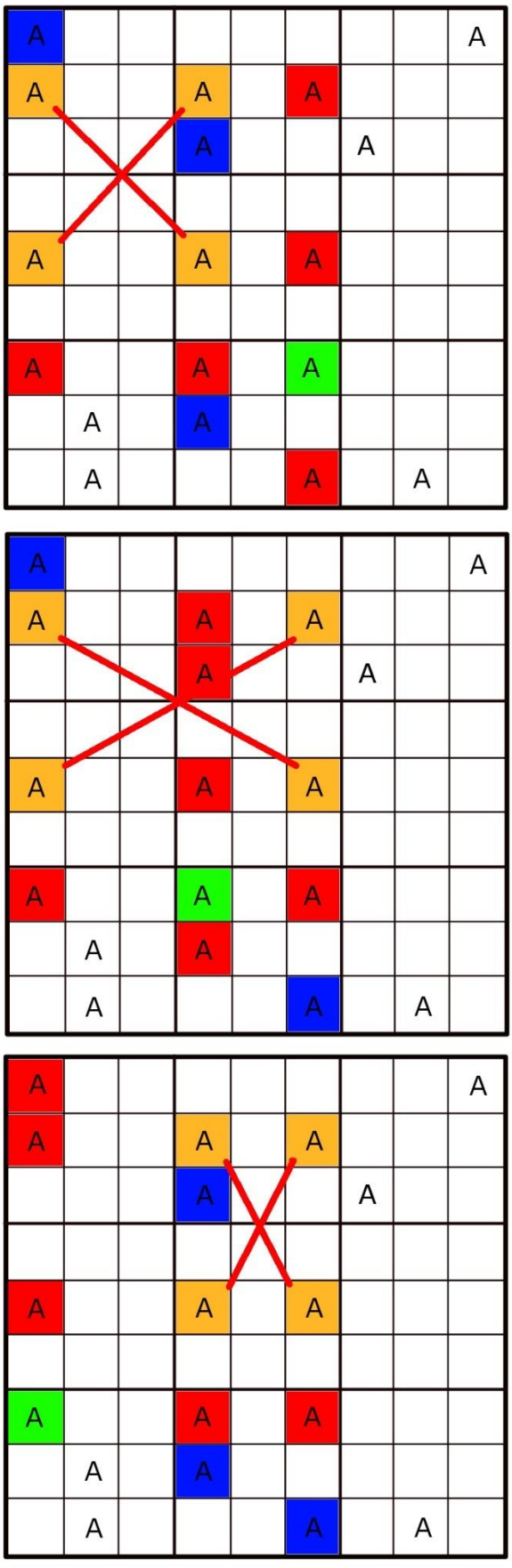


Swordfish tehnika, ruutude paiknemine

Kuna antud number peab esinema igas reas ja kolmes reas on ainult kolm veergu, kus see number võib paikneda, võib öelda, et igas veerus asub see number ühes kolmest antud reast. Seepärast võib antud veergudes eemaldada antud number kõikide muude ruutude kandidaadide hulgast – joonisel on need ruudud märgitud punasega.

Swordfish tehnikat peetakse X-Wing tehnika laienduseks ning põhjust, miks Swordfish tehnika töötab, saab seletada X-Wing abil. Selleks võib vaadelda näiteks esimesel pildil toodud 3x3 maatriksi alumist rida – seal võib antud number paikneda kolmes ruudus. Kõik paiknemise variandid on näidatud teisel pildil (numbri paiknemise asukoht on märgitud rohelisega). Igas variandis eemaldab pandud number mõned võimalikud asukohad maatriksist ja teistest ruutudest – need on joonisel märgitud punasega. Samas jääb igas variandis maatriksis 4 ruutu (värvitud oranžiga), mis moodustavad X-Wing tehnika jaoks nõutava positsiooni. X-Wing tehnika abil saab eemaldada veel hulk kandidaate – joonisel on nad märgitud sinisega. Kõigis kolmes variandis olid eemaldatud kandiaadid kõikidest ruutudest, mis on esimesel pildil värvitud punasega. Sama olukord tekib ka teistes variantides, kui proovida panna number ülemisse või

keskmisse ritta – osa kandidaate on eemaldatud pandud numbri poolt ja kõik muud – X-Wing tehnika abil.



Swordfish tehnika seletamine X-Wing tehnika abil

Swordfish tehnika töötab ka vastupidi, kui on olemas kolm veergu kolme võimalike numbri asukohaga. Sel juhul eemaldatakse kandidaadid ridadest. Sellise variatsiooni näide on toodud kolmandal pildil. Samuti võib märgata, et kandidaat 4 ei esine selles näides igas veerus täpselt 3 korda. Tegelikult on Swordfish tehnika rakendamiseks vaja, et kandidaadid esineksid igas veerus (või reas, nagu eelmistel joonistel) vähemalt 2 korda. See ei muuda tehnika põhimõtet ja oranžiga värvitud ruudud on ikka ainsad võimalikud kohad nendes ridades (veergudes), lihtsalt võimalike paiknemise kombinatsioonide arv on väiksem. Sellist Swordfish tehnika variandi nimetatakse 2-3-2 Swordfish – kandidaatide arvu järgi veergudes (ridades). Variatsiooni, mis oli enne, kui kandidaadid esinesid kõikides ridades 3 korda ja moodustasid tervet 3x3 maatriksit, nimetatakse 3-3-3 Swordfish.

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

Swordfish tehnika näide, kandidaatide eemaldamine ridades