

TARTU ÜLIKOOL
Arvutiteaduse instituut
Informaatika õppekava

Olga Puksberg

Eestikeelne veebipõhine Scrabble taoline mäng

Bakalaureusetöö (9 EAP)

Juhendaja: Krista Liin

Tartu 2024

Eestikeelne veebipõhine Scrabble taoline mäng

Lühikokkuvõte:

Bakalaureusetöö eesmärk oli luua eestikeelne veebipõhine Scrabble taoline mäng lisavõimalusega temaatilise mängu valikuks. Mäng on mõeldud ühele kasutajale, kelle vastaseks on arvuti. Mängu loomiseks kohandati olemasolevat arvuti käigu algoritmi ja sagedussõnastikku ning ametliku eestikeelse Scrabble lauamängu andmeid. Töös kirjeldatakse Scrabble mängu ja erinevaid algoritme selle arendamiseks ning töö käigus valminud veebirakendust.

Võtmesõnad:

Veebirakendus, sõnamäng, Scrabble

CERCS: P175 Informaatika, süsteemiteooria

Scrabble-like browser game in Estonian

Abstract:

The aim of the Bachelor's thesis was to create a Scrabble-like browser game in Estonian with an additional option to choose a thematic game. The game is intended for one player, playing against the computer. To create the game, an existing move generation algorithm and frequency dictionary were adjusted and used together with the data from the official Estonian Scrabble board game. The thesis includes descriptions of the Scrabble game, various algorithms for its move generation and the created game.

Keywords:

Web application, word game, Scrabble

CERCS: P175 Informatics, systems theory

Sisukord

Sissejuhatus.....	5
1. Scrabble.....	6
1.1 Mängud	6
1.2 Algoritmid.....	7
1.3 Algoritmide võrdlus	8
2. Kasutatud ressursid	10
2.1. Tehnoloogiad	10
2.1.1. HTML	10
2.1.2. JavaScript.....	10
2.1.3. Python	10
2.1.4. Firebase	11
2.2. Andmestik	11
2.3. Teemaatiline sõnastik	12
3. Käigu leidmise algoritm.....	13
3.1. Andmestruktuurid	13
3.1.1. Ankruruudud	14
3.1.2. Prefiksipuu	14
3.2. Algoritm.....	15
3.2.1. Vasaku poole leidmine.....	15
3.2.2. Pikendamine paremale	16
3.2.3. Hindamisfunktsioon.....	17
4. Programm.....	18
4.1. Reeglid	18
4.2. Struktuur	18

4.3.	Mänguloogika	18
4.4.	Mänguvaated.....	19
5.	Testimine.....	25
5.1.	Esimene etapp	25
5.2.	Teine etapp.....	25
5.3.	Võimalikud edasiarendused	26
	Kokkuvõte.....	28
	Viidatud kirjandus.....	29
	Lisad.....	30
I.	Küsimustik	30
II.	Litsents.....	37

Sissejuhatus

Sõnamängud¹ on keelemängud, mille käigus mängijad sooritavad erinevaid toiminguid sõnadega, näiteks valivad, muudavad või moodustavad uusi sõnu. Üks näide sellistest mängudest on Scrabble [1] - mäng, kus peab tähtede kombineerimisega uusi sõnu looma. See on esialgselt ingliskeelne lauamäng, millest on tehtud erinevaid versioone. Mäng sõltub suurelt keelest - iga versiooni jaoks on vajalik teada tähtede sagedus keeles, mille põhjal arvutatakse tähtede arv ja väärtus. Scrabble erinevate versioonide hulgas on olemas ka eestikeelne lauamäng. Peale lauamängude eksisteerivad veebiversioonid erinevate keelte jaoks, kuid nende hulgas ei ole eestikeelset mängu, mis oleks kõikide huviliste jaoks kättesaadav. Lisaks ei paku ükski veebiversioon temaatilise mängu võimalust, mis on klassikalise lauamängu reeglites kirjeldatud.

Scrabble'i arvutimängu loomiseks on vajalik käigu genereerimise algoritm. On mitmeid selliseid algoritme, mis kasutavad eesmärgi saavutamiseks sarnaseid meetodeid, kuid erinevad oma tõhususe poolest. Käesoleva töö eesmärk on, võttes aluseks ühe algoritmidest, luua veebipõhine Scrabble taoline mäng eesti keeles, millele on lisatud võimalus temaatilise mängu valimiseks.

Töö on jagatud viieks peatükiks. Esimeses peatükis antakse lühiülevaade Scrabble mängust ning võrreldakse erinevaid algoritme selle implementeerimiseks. Teises peatükis tutvustatakse kasutatud tehnoloogiaid ja ressursse. Kolmandas peatükis kirjeldatakse arvuti käitumise algoritmi ning selleks vajalikke andmestruktuure. Neljandas peatükis antakse ülevaade rakenduse struktuurist ning põhilistest komponentidest. Viiendas peatükis arutatakse testimistulemusi ning võimalikke edasiarendusi.

¹ <https://www.merriam-webster.com/dictionary/word%20game> (21.04.24)

1. Scrabble

Scrabble [1] on sõnamäng, kus mängijad võistlevad omavahel sõnade konstrueerimises. Sõnade koostamiseks kasutatakse seitset käes olevat täheklotsi ning klotse, mis on juba lauale pandud. Igal tähel on oma väärtus, mis sõltub tähe sagedusest keeles. Mängulaud on ruudustik, mis koosneb 225 ruudust. Iga ruut on kas tavaline või preemiaruut, mis annab rohkem punkte ühele tähele või tervele sõnale. Lõplik sõna eest saadud punktisumma arvutatakse täheklotside väärtuste liitmisel ning ruutude teguritega korrutamisel. Mängu eesmärk on saada võimalikult palju punkte.

Iga mängulauale lisatud sõna jaoks kehtivad piirangud [1]. Lisatud sõna peab olema laual kas vertikaalselt või horisontaalselt. Samuti peab uus sõna vähemalt ühte olemasoleva sõna tähte kasutama. Kuna mängulauale saab sõnu paralleelselt paigutada, siis peavad kõik kõrvuti olevad klotsid sõnad moodustama. Lisaks kõik kasutatud täheklotsid peavad asuma ühel sirgjoonel, ilma tühikuteta tähtede vahel.

1.1 Mängud

Scrabble on väga populaarne lauamäng, ja selle tõttu on tehtud palju arvuti-², mobiili-³ ja veebipõhiseid⁴ versioone. Need võimaldavad mängida teise inimese või tehisintellekti vastu ning valida mitme keele hulgast.

Eksisteerib ka eestikeelne lauamäng Scrabble ning bakalaureusetööna valminud eestikeelne Scrabble'i kujuline programm lauaarvuti jaoks [2]. See võimaldab mängida nii tehisintellekti kui ka teise inimese vastu samas arvutis.

Lisaks klassikalisele Scrabble'ile on mängust olemas ka lastele mõeldud versioon ja mõned eriväljaanded. Laste versiooni põhiline erinevus seisneb mängulauas. Selle ühel poolel asub tavaline Scrabble mängulaud. Teisel poolel on aga lastele mõeldud lihtsustatud variant, kus osa sõnu on juba mängulauale pandud. Eriväljaanded⁵⁶ on temaatilised versioonid, mis sisaldavad täiendavat sõnastikku ja lisatingimusi reeglites. Temaatiliste sõnade kasutamine ja tingimuste täitmine võimaldavad mängijal saada rohkem punkte. Töös võetakse Scrabble'i laiendustest

² nt <https://people.csail.mit.edu/jasonkb/quackle/> (01.05.24)

³ nt <https://play.google.com/store/apps/details?id=com.pieyel.scrabble&hl=en&gl=US&pli=1> (01.05.24)

⁴ nt <https://playscrabble.com/> (01.05.24)

⁵ <https://boardgamegeek.com/boardgame/114216/scrabble-cooking-edition> (01.05.24)

⁶ <https://shopping.mattel.com/en-gb/products/scrabble-harry-potter-dpr77-en-gb> (01.05.24)

üle lihtsama versiooni ja temaatilise mängu valimise võimaluse. Teema valikus piiranguid ei ole, kuna igaüks saab ise enda jaoks huvitava teema luua. Lihtsam versioon on esindatud arvuti väheskoorivate käikude kujul.

1.2 Algoritmid

Scrabble'i mängu arendamiseks tuleb lahendada mitu probleemi. Esimene tuleneb mängu nõuetest - sõnastiku koostamine ja tähtede ning nende väärtuste jaotamine. Sõnastiku loomiseks saab kasutada Internetist leiduvaid sõnade loendeid. Tähtede jaotus ning väärtused arvutatakse tähtede sageduse põhjal keeles. Jaotuses on keele sagedasemate tähtede arv suurem ning harvemini esinevate tähtede arv väiksem. Tähtede väärtuste jaotus on pöördvõrdeline tähe sagedusega - haruldastel tähtedel on suuremad väärtused ning sagedatel tähtedel on väiksemad väärtused. Teine probleem seisneb sisendi valideerimises. Iga sisestatud sõna korral tuleb kontrollida, kas see on lubatud ehk kas sõna leidub sõnastikus. Tehisintellekti käitumine, mis hõlmab käigu genereerimist ja hindamist ning sõltub sõnastiku esitusest, on kolmas probleem, mille lahendamiseks on olemas mitu lähenemisviisi.

Kõige ilmsem viis käikude genereerimiseks seisneb tähtede permutatsioonide kasutamises. Permutatsioon tähendab mängijal käes olevate tähtede ümberkorraldamist, et kontrollida iga tähe jaoks iga võimalikku asukohta. Idee on otsida mängulaua kõikide kohti, kuhu saab juba seal olevatele tähtedele käes olevaid täheklotse lisada, kas ette, vahele või taha. Seejärel läbi proovida kõikide leitud tähtede permutatsioonid, et leida kehtivad sõnad ja nende asukohad mängulaua. See algoritm on lihtne, kuid ei ole hea lahendus. Mängulaua läbimine ja kõikide tähe kombinatsioonide läbiproovimine võtab palju aega ning sellist algoritmi kasutatav programm on aeglane.

Stuart C. Shapiro ja Howard R. Smithi publikatsioonis kirjeldatud algoritm [3] kasutab käigu genereerimiseks erilist andmestruktuuri nimega prefiksipuu (ingl *trie*) või tähetabel. Prefiksipuu on kahendpuu, kus iga tipp sisaldab spetsiifilist informatsiooni. Töös kasutatud andmestruktuuri iga tipp sisaldab informatsiooni kõikidest võimalikest sõnadest, mis võivad olla konstrueeritud tippu viivatest tähtede permutatsioonidest ja kombinatsioonidest. Tähed asuvad vastavalt nende väärtustele mängus kahanevas järjekorras. Käigu leidmise algoritm edastab kõik sõna asetamise jaoks võimalikud positsioonid leksikoni otsingu algoritmile. Iga tagastatud leitud sõna jaoks kontrollitakse sobivust paigutusreeglitele ning seejärel hinnatakse. Kui leitakse käik, mis on väärt mingi etteantud minimaalse arvu punkte või lihtsalt kõrgeima punktisummaga käik, tehakse see.

Andrew W. Appeli ja Guy J. Jacobsoni töös esitatud algoritm [4] kasutab ankruruute (ingl *anchor squares*), ristkontrolli hulkasid (ingl *cross-check sets*) ja spetsiaalset andmestruktuuri sõnastiku esituseks nimega DAWG ehk suunatud atsükliline sõnagraaf (ingl *Directed Acyclic Word Graph*). Ankruruut on tühi ruut, mis külgneb horisontaalselt või vertikaalselt tähte sisaldava ruuduga. Ristkontrolli hulk on tähtede hulk, mida saab ankruruutu panna ning mis moodustavad kehtivaid risti sõnu ehk vertikaalseid või horisontaalseid sõnu, kui pannakse vastavalt horisontaalset või vertikaalset sõna läbi antud ankruruudu. Ankruruutude ja ristkontrolli hulkade kasutamine lihtsustab käigu leidmise, kuna ei ole vaja kogu mängulauda läbi vaadata, vaid ainult ankruruutudega ridu. Samuti lükatakse sobimatud sõnad kohe tagasi, sest arvestatakse ainult sõnu, mille tähed sisalduvad vastava ruudu ristkontrolli hulgas. Kasutades seda informatsiooni ja tagurdusalgoritmi (ingl *backtracking algorithm*) leitakse kõik võimalikud sõnad ja nende asukohad mängulaua. Käigu leidmise algoritm koosneb kahest osast. Kõigepealt leitakse kõik võimalikud vasakud pooled, mis koosnevad ankruruudust vasakul asuvatest ruutudest. Iga vasaku poole jaoks leitakse paremad pooled, mis koosnevad ankruruudust ja sellest paremal asuvatest ruutudest. Kui on jõutud sõna lõppu, salvestatakse leitud kõik. Selle algoritmiga on käigu genereerimine üldjuhul kiire. Viivitust võib esineda juhul, kui täheklotside seas on tühjad ruudud, mida võib kasutada iga tähena. Sellisel juhul suureneb võimalikkude sõnade arv, mis omakorda toob kaasa otsinguaja pikenemise.

Steven A. Gordoni algoritm [5] on Andrew W. Appeli ja Guy J. Jacobsoni töö edasiarendus. Töös kirjeldatud algoritm kasutab samamoodi ankruruute ja ristkontrolli hulkasid, kuid pakub välja teise andmestruktuuri sõnastiku esituseks nimega GADDAG ehk kahesuunaline suunatud atsükliline graaf (ingl *Directed Acyclic Graph*). Selles on erinevalt DAWG struktuurist hoitud iga sõna jaoks ka selle ümberpööratud prefiksid. Kõikide võimalike käikude genereerimiseks kasutatakse samuti tagurdusalgoritmi. Algoritmi töö algab sõna pikendamisega vasakule, kuni on ruumi. Pärast seda püütakse sõna paremale pikendada. Kui kõik arvuti täheklotsid on kasutatud ning sõna vasakule või paremale pikendada ei ole võimalik, salvestatakse leitud kõik.

1.3 Algoritmide võrdlus

Kolm ülal kirjeldatud algoritmi kasutavad erinevaid andmestruktuure sõnastiku esitamiseks, mis mõjutab nende kiirust. Stuart C. Shapiro ja Howard R. Smithi programmis kasutatakse muudetud tähejärjestusega tähetabelit. Selle ülesehituse tõttu leitakse käigu jaoks eeskätt rohkem punkte arvu väärt sõnad, kuid käigu leidmine on aeglane. Andrew W. Appeli ja Guy J.

Jacobsoni algoritmi sõnastiku andmestruktuur on prefiksipuu, kus on ühised sufiksids kokku liidetud. Selle tulemusena tekib suunatud atsükliline sõnagraaf ehk DAWG, mis lubab kiiret käigu genereerimist. Seda kasutades läks 94240 sõnast koosnevast leksikonist kõikide võimalikkude käikude leidmiseks üks või kaks sekundit, võrreldes 30 või 40 sekundiga, mis kulusid Shapiro programmil käigu leidmiseks 1500 või 2000 sõnast koosnevast leksikonist [4]. Steven A. Gordoni töös esitatud andmestruktuur on prefiksipuu, mis hoiab iga sõna jaoks selle iga ümberpööratud prefiksi. Tulemusena tekib kahesuunaline suunatud atsükliline graaf ehk GADDAG. Selle kasutamine lubab umbes kaks korda kiiremat käigu genereerimist võrreldes DAWG andmestruktuuriga, kuid võtab umbes viis korda rohkem mälu sama leksikoni jaoks [5].

2. Kasutatud ressursid

Töö praktilises osas on kasutatud erinevaid tehnoloogiaid ja ressursse programmi loomiseks ning mängu jaoks sõnastiku koostamiseks.

2.1. Tehnoloogiad

Programm on kirjutatud peamiselt kasutades JavaScripti programmeerimiskeelt ja sellel põhinevaid tehnoloogiaid ning vähesel määral HTML'i. Baassõnastiku eeltöötlemiseks on kasutatud programmeerimiskeelt Python ja selle teeki EstNLTk. Mäng on kasutajatele kättesaadavaks tehtud Firebase pilvteenuste abil.

2.1.1. HTML

HTML [6] (*HyperText Markup Language*) on märgistuskeel, mida kasutatakse veebilehe ja selle sisu struktuuri loomiseks. HTML-i kasutatav veebileht on programmi sisenemispunkt.

2.1.2. JavaScript

JavaScript [6] on programmeerimiskeel, mida kasutatakse veebilehele interaktiivsuse lisamiseks. Kõik funktsioonid ja komponendid, mis kirjeldavad mängu loogikat ning on vajalikud programmi toimimiseks on kirjutatud JavaScript keeles. Lisaks sellele kasutab programm tehnoloogiaid Node.js serveripoolse ja Phaser 3 kliendipoolse komponendi jaoks.

Node.js [7] on JavaScripti käituskeskkond, mida kasutatakse võrgurakenduste loomiseks. Programm kasutab Node.js raamistikku Express serveri loomiseks ning kliendile andmete saatmiseks; olemust fetch päringute tegemiseks; pakendit CORS serveri ja kliendi vahelise ressursside jagamise võimaldamiseks.

Phaser [8] on HTML5 raamistik veebimängude loomiseks. Programm kasutab selle sisseehitatud funktsioonid mängu struktuuri loomiseks ja rexUI pistikprogramme [9] kasutajalt sisendi saamiseks.

2.1.3. Python

Python on programmeerimiskeel, millel on mitmeid rakendusvaldkondi. Töös kasutatakse selle loomulike keele töötluse paketti EstNLTK [10] baassõnastiku eeltöötlemiseks. Sõnastikust eemaldatakse võõrkeelsed sõnad Vabamorf taggeri abil ning genereeritakse sõnavorme Vabamorfi morfoloogilise süntesaatori abil.

2.1.4. Firebase

Firebase [11] on Google'i pakutav rakenduste arendusplatvorm. Selle teenuseid Firebase Hosting, Cloud Storage for Firebase ja Cloud Functions for Firebase on kasutatud programmi veebi üles panemiseks.

Firebase Hosting on teenus, mis võimaldab staatilise ja dünaamilise sisu teenindamist. Rakenduses kasutatakse antud teenust programmi failide hoidmiseks ja kasutaja üleslaetud failidele ligipääsu saamiseks.

Cloud Storage võimaldab failide üles ja alla laadimist ning kasutaja loodud sisu hoidmist. Rakenduses kasutatakse antud teenust kasutajate üles laetud sõnastikkude hoidmiseks.

Cloud Functions on raamistik, mis võimaldab automaatset serveri koodi käivitamist vastuseks HTTPS-i päringutele. Rakenduses kasutatakse antud teenust kasutaja laetud sõnastiku Cloud Storage's salvestamiseks ja selles olevatele failidele ligipääsu saamiseks.

2.2. Andmestik

Sõnastikku kasutatakse arvuti käigu leidmiseks ja kasutaja sisestatud sõnade kontrollimiseks. Programmi baassõnastik põhineb Eesti Keeleressursside Keskuse veebilehelt leitaval sagedusloendil "Tasakaalus korpuse lemmad tähestikulises järjekorras" [12]. Failis on 43657 rida, igal real on sõna sagedus ning sõna algses vormis. Sagedusloend sisaldab käändsõnu (nimi-, omadus-, arv- ja asesõnad) ainsuse nimetavas käändes, pöördõnu (tegevusõnad) ma-tegevusnimena ja muutumatuid sõnu (side-, hüüd-, määr- ja kaassõnad).

Sõnastiku eeltöötluste kood asub failis „preprocessing.py“. Programmi baassõnastiku saamiseks on sagedusloendist eemaldatud sõnad, mis sisaldavad vahemärke või mängu tähtede jaotusest puuduvaid tähte ning sõnad, mille pikkus on suurem kui 15. Samuti on EstNLTK vahendite abil eemaldatud võõrkeelsed sõnad. Selleks leiti iga sõna liik ja vorm. Kui üks neist omadustest ei olnud määratud, loeti sõna võõrkeelseks ning jäeti sõnastikust välja. Käändsõnadele on EstNLTK abil sünteesitud mitmuse nimetavad käänded. Sõnastikus sisalduvad käändsõnad ainsuse ja mitmuse nimetavas käändes, pöördõnad ma-tegevusnimena ning muutumatud sõnad. Lõppsõnastik on tähestiku järgi sorteeritud ning koosneb 71205 sõnast.

2.3. Temaatiline sõnastik

Temaatiline sõnastik on kasutaja üles laetud tekstifail. Vältimaks ebasobivate sõnade sattumist faili, mis on kõikidele kasutajatele kättesaadav, faili sisu kontrollitakse. Selleks kasutatakse Filosofti loodud veebiprogrammi - eesti keele HTMLi morfanalüsaatorit [13], sest valitud tehnoloogiatega on mängu ajal veebipäringut lihtsam ja mugavam kasutada. Sõnastiku valideerimise kood asub failis „server.js“.

Esiteks tehakse päring morfanalüsaatori lehele. Negatiivse tulemuse korral päringust saadud vastus on „####“, mis tähendab, et sõna analüüsida ei ole võimalik. Tulemusliku päringu puhul saadud info sisaldab muuhulgas sõnavormi. Seda kasutatakse kontrolliks, kas sõna on vastavalt reeglitele algvormis või käändsõnade puhul mitmuse nimetavas käändes. Järgmisena leitakse sõna pikkus, mis võib maksimaalselt olla 15 tähte. Tingimustele vastavatest sõnadest koostatakse ning salvestatakse uus temaatiline sõnastik.

3. Käigu leidmise algoritm

Arvuti käigu leidmise algoritm põhineb Andrew W. Appeli ja Guy J. Jacobsoni kirjeldatud algoritmil, kuna Steven A. Gordoni algoritm on üsna keeruline arusaamiseks. Programmi algoritm erineb algsest algoritmist selle poolest, et ei kasutata ristkontrolli hulkasid ja DAWG asemel kasutatakse prefiksipuud. Loodud algoritm asub failis „algorithm.mjs“.

Andrew W. Appeli ja Guy J. Jacobsoni töös kirjeldatud algoritm töötab, võttes arvesse ainult ühe rea sisu, selle rea ankruruute ja ristkontrolli hulkasid. See tähendab, et ristkontrolli hulkade leidmisel arvestatakse ainult kahetäheliste risti tekkivate sõnadega. Töös loodud programmi algoritm ei kasuta ristkontrollide hulkasid. Selle asemel kontrollitakse, kui terve sõna on konstrueeritud, kas seda saab leitud positsioonile paigutada. See tähendab, et mittesobivaid sõnu ei lükata kohe tagasi, vaid võetakse seevastu sõna sobivuse hindamisel arvesse mis tahes pikkusega risti tekkinud sõnu. Kuna ei ole antud lahenduse kasutamine ajaliselt märgatav, on see algoritmi jaoks sobiv.

Programmi algoritmi töö alguses leitakse ankruruudud ning edastatakse need ükshaaval funktsioonile, mis leiab sõna vasaku poole. See funktsioon kutsub omakorda välja iga leitud vasaku poole puhul funktsiooni, mis pikendab sõna paremale. Pärast seda, kui sõna on täielikult leitud, tehakse kindlaks, kas seda saab leitud positsioonile paigutada. Selleks kontrollitakse, kas uus sõna moodustab koos juba mängulaua olevate tähtedega horisontaalselt ja vertikaalselt kehtivaid sõnu. Kui uue sõna paigutamine leitud positsioonile on võimalik, kutsutakse välja hindamisfunktsioon, mis leiab käigu eest saadud punktid. Selle põhjal salvestatakse kõik, mille punktisumma on suurim. Kui kõik ankruruudud on läbi vaadatud, uuendatakse mängulaua olekut ja arvuti täheklotse kasutades salvestatud käigu informatsiooni.

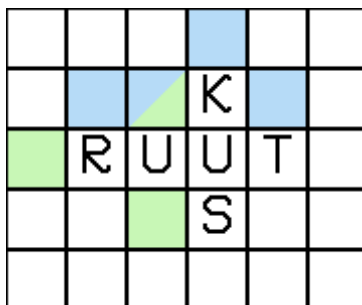
Esialgu leitakse võimalikud horisontaalsed käigud, seejärel transponeeritakse mängulauda ja käivitatakse algoritm uuesti vertikaalsete käikude leidmiseks. Sellise lahendusega ei kulu arvutil umbes 70000 sõnast koosnevast sõnastikust käigu leidmiseks mitutkümmet sekundit, vaid piisab üldiselt ühest või kahest sekundist, seega antud lahendus sobib mängu jaoks.

3.1. Andmestruktuurid

Programmi jaoks vajalikud kaks peamist andmestruktuuri on ankruruudud ja prefiksipuu. Ankruruudud hoiavad sõnade paigutamiseks sobivate võimalike positsioonide infot. Prefiksipuud kasutatakse sõnastike esitamiseks.

3.1.1. Ankruruudud

Ankruruute on kaht tüüpi: horisontaalsed ja vertikaalsed. Horisontaalsed ankruruudud on ruudud, mis asuvad mängulaual olevate tähtede kohal. Vertikaalsed ankruruudud on ruudud, mis asuvad mängulaual olevate tähtede vasakul. Joonisel 1 on eri värvidega näidatud erinevad ankruruudud.

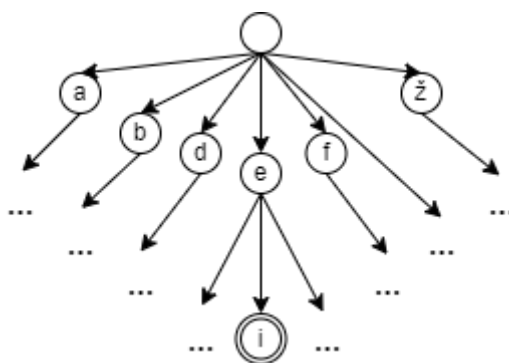


Joonis 1. Vertikaalsed (rohelist) ja horisontaalsed (sinised) ankruruudud

Iga ankruruudu kohta on salvestatud rea ja veeru indeksid ning piir ehk tühjade ruutude arv järgmise mittetühja ruuduni. Kõikide ankruruutude info sisaldub eraldi loendis.

3.1.2. Prefiksipuu

Sõnastik on esitatud prefiksipuuna. Joonisel 2 on osa sõnastiku graafilisest esitusest.



Joonis 2. Sõnastiku graafiline esitus

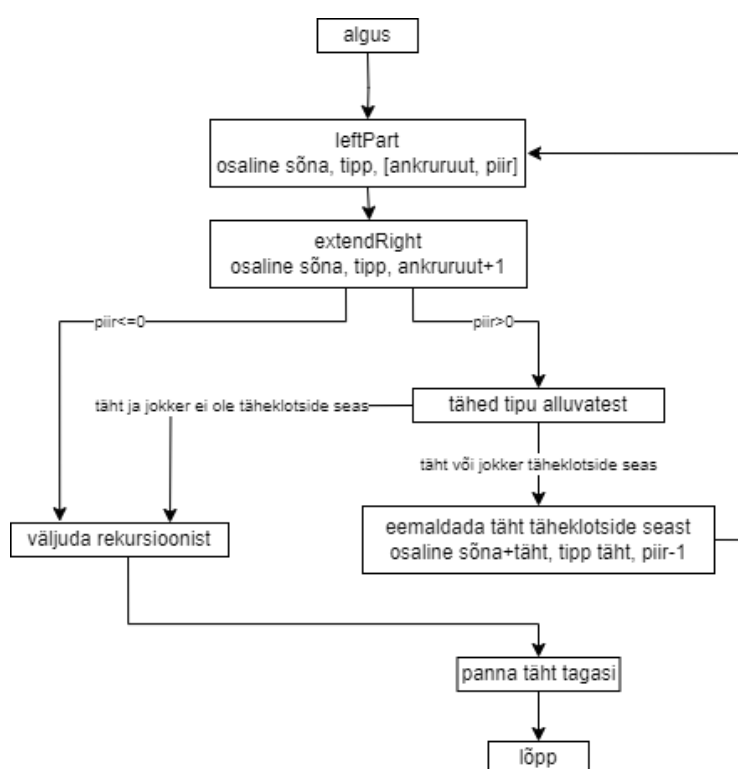
Prefiksipuu iga tipp sisaldab informatsiooni tipus hoitava tähe kohta, kas tipp tähistab sõna lõppu ja alamtippude sõnastikku. Alamtippude sõnastiku võtmed on tähed ja väärtused on vastavaid tähte sisaldavad tipud. Juurtipp on tähistatud tühja sümboliga.

3.2. Algoritm

Käigu leidmiseks kasutatakse rekursiivseid funktsioone sõna vasaku ja parema poole leidmiseks. Terve sõna leidumise korral arvutatakse käigu eest saadav punktide arv ning salvestatakse mängu tingimustele vastav käik.

3.2.1. Vasaku poole leidmine

Algoritmi töö algab sõna vasaku poole leidmisega. Selle jaoks kutsutakse funktsiooni *leftPart*, mille argumendid on osaline sõna (esialgu tühi), sõnastiku juurtipp ja ankruruut. Joonisel 3 on näidatud funktsiooni *leftPart* täitmise skeem.



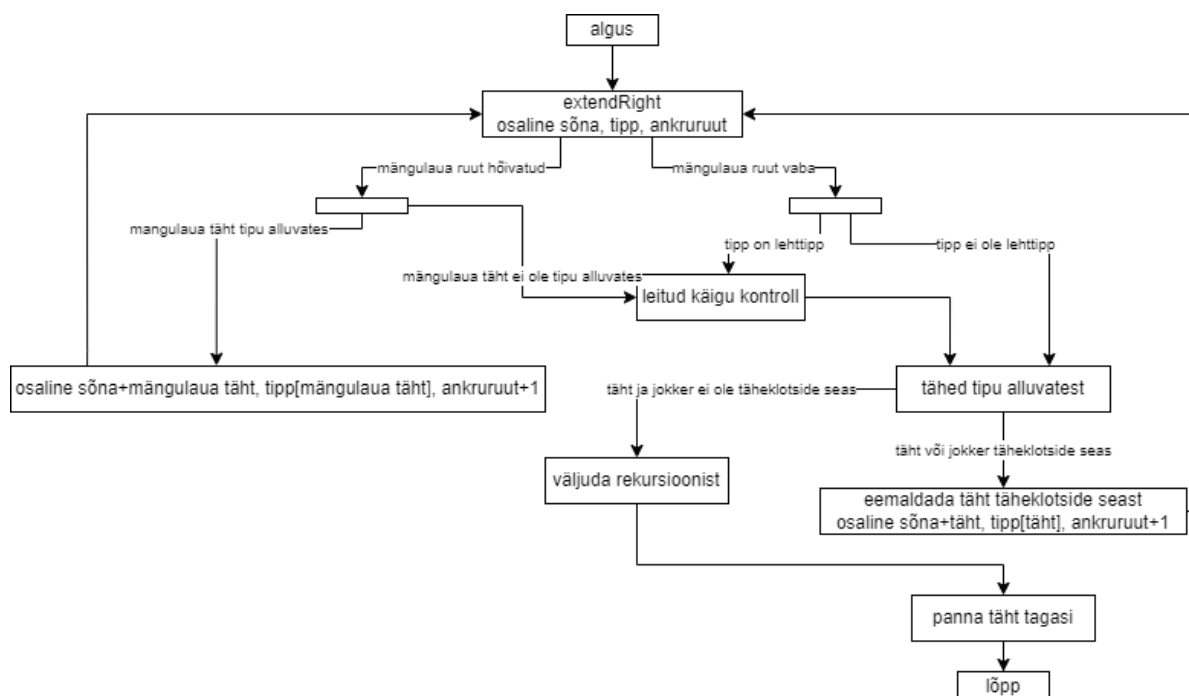
Joonis 3. Vasaku poole leidmise skeem

Selles funktsioonis edastatakse leitud osaline sõna funktsioonile *extendRight*, mis olemasoleva vasaku poole jaoks leiab sõna parema poole. Edasi kontrollitakse, kas sõna saab vasakule pikendada. Kui ei, funktsioon lõpetab töö. Kui jah, siis vaadatakse ükshaaval tähti sõnastiku tipu alamtippudest. Iga tähe jaoks kontrollitakse, kas täht või jokker (klots, mis võib tähistada mistahes tähte) on mängija täheklotside seas olemas. Kui ei, siis funktsioon lõpetab töö. Kui jah, siis täheklotside seast eemaldatakse vaadeldav täht. Seejärel tehakse rekursiivne kutse funktsioonile *leftPart* muudetud argumentidega. Nimelt lisatakse osalisele sõnale vaadeldav

täht, tipuks määratakse tähte sisaldavat alamtippu ning piiri vähendatakse ühe võrra. Funktsiooni töö lõppemisel lisatakse täht tagasi täheklotside hulka.

3.2.2. Pikendamine paremale

Pikendamine paremale on tehtud funktsiooniga *extendRight*, mille argumendid on osaline sõna, sõnastiku tipp ja mängulaua ruut. Joonisel 4 on näidatud funktsiooni *extendRight* täitmise skeem.



Joonis 4. Paremale pikendamise skeem

Kui funktsioonile argumendina edastatud ruut on vaba, siis on kaks võimalust. Kui sõnastiku tipp märgib sõna lõppu, siis edastatakse tekkinud sõna ja selle sõna indeks mängulaua funktsioonile *legalMove*, mis skoorib sõna ning salvestab suurima punktisummaga käigu. Kui sõnastiku tipp ei märgi sõna lõppu, siis vaadatakse ükshaaval tähti sõnastiku tipu alamtippudest. Iga tähe jaoks kontrollitakse kas täht või jokker on mängija täheklotside seas olemas. Kui ei, siis funktsioon lõpetab töö. Kui jah, siis täheklotside seast eemaldatakse vaadeldav täht. Seejärel tehakse rekursiivne kutse funktsioonile *extendRight* muudetud argumentidega. Nimelt lisatakse osalisele sõnale vaadeldav täht, tipuks määratakse tähte sisaldavat alamtippu ning võetakse järgmine ruut paremale. Funktsiooni töö lõppemisel lisatakse täht tagasi täheklotside hulka.

Kui mängulaua ruut ei ole vaba, siis kontrollitakse kas ruudul olev täht on tipu alamtippude seas. Kui ei, siis funktsioon lõpetab töö. Kui jah, siis tehakse rekursiivne kutse funktsioonile

extendRight muudetud argumentidega. Nimelt lisatakse osalisele sõnale mängulaua ruudul olev täht, tipuks määratakse ruudu tähte sisaldavat alamtippu ning võetakse järgmine ruut paremale.

3.2.3. Hindamisfunktsioon

Käigu väärtuse leidmiseks leitakse eraldi iga uue moodustatud sõna väärtus ning seejärel liidetakse väärtused kokku lõppskoori saamiseks. Sõna väärtus sõltub selle tähtede väärtustest ning mängulaua ruutudest millel sõna paikneb. Tähtede väärtused põhinevad ametliku eestikeelse Scrabble väljaande andmetel [\[14\]](#).

Iga tähe väärtus korrutatakse ruudu teguriga, mis võib olla üks, kaks või kolm, kui täht asub vastavalt tavalisel, tähe väärtuse kahe- või kolmekordistaval ruudul. Erand on täht, mida kujutab jokker - selle väärtus on alati null. Kõik saadud väärtused liidetakse kokku ning korrutatakse teguriga, mis suurendab kogu sõna väärtust. Iga tähe korral, mis asub tavalisel, sõna väärtuse kahe- või kolmekordistaval ruudul korrutatakse see tegur vastavalt ühe, kahe või kolmega. Tähe või sõna kahe- või kolmekordistavad ruudud mõjutavad ainult ühe käigu väärtusele. Kui vaadeldav sõna on temaatiline, lisatakse sõna väärtusele punktide arv, mis on võrdne selle sõna pikkusega.

Selleks, et mängijal oleks võimalus mängida vastavalt oma võimetele, saab valida kolme raskusastme vahel: kerge, keskmine ja raske. Vastavalt valitule salvestatakse kindlatele tingimustele vastav käik. Kergel raskusastmel salvestatakse esimese kümme käigu puhul käik, mille skoor on suurem kui viis; edasi kõige vähem punkte teeniv käik. See on tehtud selleks, et mängu alguses ei tekiks ainult kahetähelisi sõnu, mida ei saa pikendada. Keskmisel raskusastmel salvestatakse viimasena leitud käik, mida võib vaadelda juhusliku käigu valikuga. Raskel raskusastmel salvestatakse kõige rohkem punkte teeniv käik.

4. Programm

Valminud programmi lähtekood on leitav GitHubi repositooriumis <https://github.com/OlgaP21/scrabble-like-game-est.git> ning valminud rakendus asub aadressil <https://play-scrabblest.web.app/>.

4.1. Reeglid

Loodud mängu reeglid on samad, mis ametlikus eestikeelses Scrabble lauamängus [1], kuid muudatustega temaatilise mängu osas. Kuna temaatilise mängu versioon on kirjeldatud algse Scrabble lauamängu reeglites, siis on loodud rakenduses temaatilise mängu võimalus kasutajatele kättesaadav mängu huvitavamaks muutmiseks.

Mängu jaoks on võimalik valida temaatiline sõnastik, kuhu on koondatud mõne kindla teemaga seotud sõnad. Kasutajal on samuti võimalus ise enda sõnastikku failina üles laadida. Pärast faili valideerimist lisatakse see serverile ning sõnastiku saavad kõik kasutajad mänguks valida.

4.2. Struktuur

Üks tähtis programmi osa on kaust nimega „public“. See sisaldab mänguks vajalikke ressursse (tähtede ja nende väärtuste pildid, mänguvälja ruutude pildid, põhisõnastik, temaatilised sõnastikud, mängu vaadete ja loogika kirjeldust sisaldavad failid) ning faile (mängu sisenemispunkt – „index.html“, peamine skript – „main.js“).

Teine tähtis programmi osa on fail nimega „server.js“, mis on serveri rollis. Selles failis luuakse rakendus, mille abil teenindatakse staatilisi faile ning teostatakse suhtlust kliendiga. Sealhulgas kuulub loodud rakenduse funktsioonide hulka kasutaja üles laetud faili valideerimine ning salvestamine.

4.3. Mänguloogika

Mänguloogika on jagatud kuueks failiks: mängulaud („gameboard.mjs“), laud („board.mjs“), tähtede kott („bag.mjs“), sõnastik („dictionary.mjs“), algoritm („algorithm.mjs“) ja mängija („player.mjs“). Iga fail vastutab kindla mängu komponendi eest.

Mängulaua fail sisaldab klassi mängulaua visuaalse esituse hoidmiseks. Mängulaud koosneb ruutudest, millest igaühel on X- ja Y-koordinaadid ning pilt tähistamiseks kas ruut on tavaline või preemiaruut.

Laua fail sisaldab mängulauaga seotud muutujaid, mida kasutatakse selle oleku hoidmiseks: laud vabade ruutude märkimiseks, laud preemiaruutude märkimiseks, mängulaua olek (transponeeritud või mitte), kõik mängulaua asetsevad vertikaalsed ja horisontaalsed sõnad. Lisaks sisaldab fail ka funktsioone mängulaua initsialiseerimiseks, transponeerimiseks ja uuendamiseks ning ankruruutude leidmiseks.

Tähtede koti fail sisaldab täheklotsidega seotud muutujaid, mida kasutatakse tähtedega seotud informatsiooni hoidmiseks: tähtede kott ja tähtede väärtused ning mängijate käes olevad täheklotsid. Lisaks sisaldab fail ka funktsioone tähtede koti initsialiseerimiseks ja uuendamiseks, mängijatele täheklotside jagamiseks ja mängijate käes olevate täheklotside uuendamiseks.

Sõnastiku fail sisaldab klassi sõnastiku esitamiseks ning muutujaid sõnastiku info hoidmiseks: sõnastik, temaatiline sõnastik ja valitud mänguteema. Lisaks sisaldab fail ka funktsioone nii baas- kui ka temaatilise sõnastiku initsialiseerimiseks.

Algoritmi fail sisaldab arvuti käigu jaoks vajalikke muutujaid: parim leitud käik, käiguga tekitatud sõnad, parim skoor, käigu tüüp (esimene või mitte) ja valitud mängu keerukus. Lisaks sisaldab fail ka funktsioone muutujate lähtestamiseks, arvuti käigu sooritamiseks, leitud käigu kontrolliks, skoorimiseks ja salvestamiseks ning mängu oleku uuendamiseks.

Mängija fail sisaldab mängija käigu jaoks vajalikke muutujaid: pakutud sõna tüüp (vertikaalne või horisontaalne), mängija käik, käiguga tekitatud sõnad, mängija täheklotside hulgast kasutatud tähed ning nende asukohad mängulaua visuaalsel esitusel. Lisaks sisaldab fail ka funktsioone muutujate lähtestamiseks, mängija käigu sooritamiseks, leitud käigu kontrolliks, skoorimiseks ja salvestamiseks ning mängu oleku uuendamiseks.

4.4. Mänguvaated

Kasutajaliides koosneb kuuest vaatest: eellaadimine („preloader.js“), peamenüü („mainMenu.js“), reeglid („rules.js“), sõnastikud („options.js“), mänguseaded („gameOptions.js“) ja mäng („game.js“). Iga vaade täidab kindlat eesmärki.

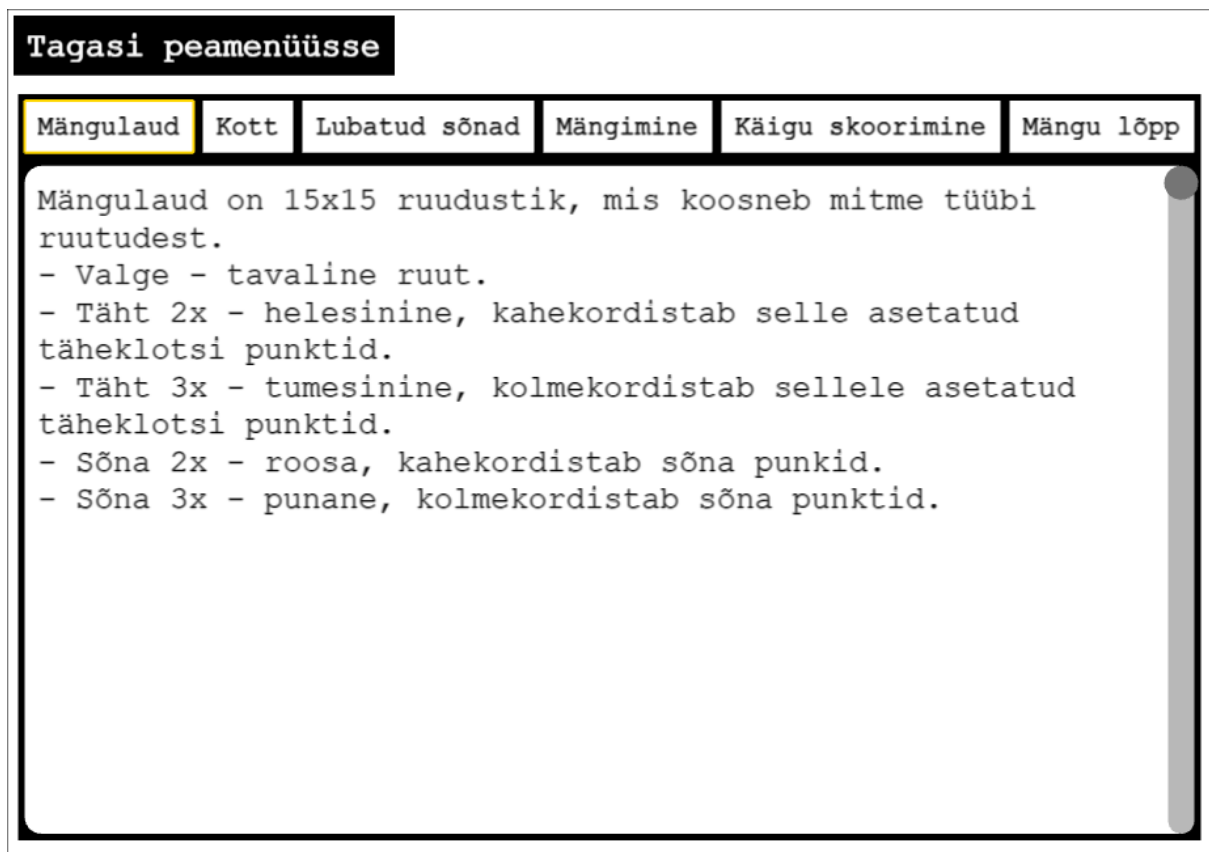
Eellaadimise vaade on mängijale nähtamatu, kuid see on vajalik mängu toimimiseks. Selles vaates toimub vajalikkude piltide (tähed, tähtede väärtused, ruudud) eellaadimine. Selle lõpus avatakse peamenüü vaade.

Peamenüü on esimene vaade, mida mängija näeb. Vaade koosneb kolmest nupust, mis võimaldavad minna reeglite, sõnastiku üleslaadimise ning mänguseadete vaadettesse. Joonisel 5 on näidatud peamenüü vaade.



Joonis 5. Peamenüü

Reeglite vaates näeb kasutaja mängureegleid. Vaade koosneb kuuest vahekaardi lehest, millest igal lehel kirjeldatakse ühte mängu aspekti. Siit saab minna tagasi peamenüüsse. Joonisel 6 on näidatud reeglite vaade.



Joonis 6. Reeglid

Sõnastike vaates saab kasutaja enda temaatilist sõnastikku üles laadida. Siin on antud lühike üles laetava faili struktuuri kirjeldus ning nupp enda sõnastiku lisamiseks. Faili valimata jätmise, ebasobiva nime või üleliigse suuruse puhul näidatakse kasutajale veateadet. Siit saab minna tagasi peamenüüsse. Joonisel 7 on näidatud sõnastike vaade.

Tagasi peamenüüsse

Siin saab oma temaatilise sõnastiku üles laadida
Laetava faili formaat - .txt
Failis võivad sõnad olla eraldatud kas
komaga, tühikuga või asuda igaüks omal real
Laetava faili maksimaalne suururs - 1 kilobait
Sõnad kontrollitakse Filosofti morfanalüsaatori abil
Ebasobivad sõnad eemaldatakse

Vali sõnastiku fail

Joonis 7. Sõnastikud

Mänguseaded on mängueelne vaade mängu seadete valimiseks. Esimesena saab valida mängu raskusastme - kerge, keskmine või raske. Raskusastme juures on lühidalt kirjas, kuidas arvuti valitud tasemel käitub. Teisena saab kasutaja valida temaatilise või tavalise mängu. Pärast valikut näidatakse paremal olevas tekstialas mängu jaoks valitud sõnastiku sisu. Vaatest saab minna tagasi peamenüüsse või alustada mängu. Joonisel 8 on näidatud mänguseadete vaade.

Tagasi peamenüüsse

Vali mängu keerukus

Kerge - Arvuti teeb väheskoorivaid käike

Keskmine - Arvuti teeb juhuslikke käike

Raske - Arvuti teeb parimaid käike

teemata

Baasõnastik:

- aa
- aabits
- aabitsad
- aade
- aadel

Mängima!

Joonis 8. Mänguseaded

Mängu vaates toimub mäng. Alguses initsialiseeritakse mängu jaoks vajalikud komponendid (laud, tähtede kott, sõnastikud). Seejärel näidatakse mängulauda, kasutaja tähti, mõlema mängija punktide arvu ning tehtud käikude infot. Kasutajal on võimalus teha käik, vahetada tähte või jätta käik vahele. Reeglite rikkumise korral näidatakse kasutajale veateadet. Mängu lõpus näidatakse mõlema mängija lõplikku punktide summat ning kasutajal on võimalus minna tagasi peamenüüsse või alustada uut mängu. Mängu saab ka varem lõpetada vajutades vastavale nupule, mis viib kasutaja peamenüüsse. Joonisel 9 on näidatud mängu vaade.

Mängija punktid:

Arvuti punktid:

K ₁	Sõna 3X			Täht 2X				Sõna 3X				Täht 2X			Sõna 3X
S ₁		Sõna 2X				Täht 3X				Täht 3X				Sõna 2X	
I ₁			Sõna 2X			Täht 2X		Täht 2X				Sõna 2X			Täht 2X
T ₁				Sõna 2X						Sõna 2X					
E ₁			Täht 2X			Täht 2X		Täht 2X		Täht 2X			Täht 2X		
U ₁					Sõna 2X					Sõna 2X					
D ₂			Sõna 2X				Täht 2X		Täht 2X			Sõna 2X			
		Sõna 2X			Täht 3X				Täht 3X			Sõna 2X			
	Sõna 3X			Täht 2X			Sõna 3X					Täht 2X			Sõna 3X

Esimesena käib mängija

Jäta kõik vahele

Vaheta tähed

Valmis

Lõpeta mäng

Joonis 9. Mäng

5. Testimine

Mängu testimine toimus kahe etapina. Esimese etapi kasutajad mängisid mängu ning andsid tagasisidet tekstifaili või edastasid suuliselt. Selle põhjal olid parandatud vead ning uut rakenduse versiooni testisid teise etapi kasutajad.

5.1. Esimene etapp

Esimene rakenduse testimine toimus aines “Lõputöö seminar”, kus kaastudengid proovisid mängida ning kas kirjutasid tagasisidet eraldi tekstifaili või ütlesid suuliselt.

Testimisel esines kaks viga. Esimese vea tõttu oli võimalik risti tekkivate sõnade korral olematuid sõnu moodustada. Teine viga tekkis mängija tähtede näitamisel, kui nende hulgas oli jokker. Esimese probleemi lahenduseks oli kasutajasisendi kontrolli funktsiooni täiendamine. Teise probleemi põhjus seisnes mänguobjekti omaduse muutmises, mis takistas selle objekti hävitamist. Selle probleemi lahenduseks oli muudetud omaduse nullimine.

Lisaks vigade parandamisele oli uuendatud sõnastik ja infopaneeli käitumine vastavalt kasutajate kommentaaridele. Sõnastikust puudus sõna “ma”, mis sai sinna hiljem lisatud. Infopaneel jäi algselt asendisse, kuhu kasutaja selle jättis, kuid see sai tirimise vastu välja vahetatud, et näidata kõige uuemat käiku.

5.2. Teine etapp

Teises testimise etapis osales viis inimest. Selle jaoks oli koostatud küsimustik, millele testijad vastasid pärast mängu proovimist. Küsimustik koosnes kuuest osast: küsimused mängu raskusastme, temaatilise mängu, mängu reeglite, mängu kujunduse, mängu kasutusmugavuse ning mängu toimimise kohta. Lõpus oli üks küsimus üldiste kommentaaride või soovitude jaoks. Testimise jaoks kasutatud küsimustik asub lisas 1.

Testimistulemustest selgus, et kolme testija jaoks vastas valitud raskusaste ootustele arvuti käikude suhtes. Üks mängijatest ei nõustunud selle väitega ning kirjutas, et keskmisel raskusastmel mängides sai arvuti vähem punkte kui kergel raskusastmel. Neljale testijale meeldis temaatilise mängu valimise võimalus, kuigi kuna ühtegi teemasõnastikku testimise ajal ei olnud, siis temaatilist mängu keegi ei proovinud. Sõnastiku üles laadimise juhiste kohta saadud tagasiside oli pigem neutraalne ning üks testijatest märkas, et juhistele saaks lisada järjekorda, kuidas faili üles laadida. Mängureeglid olid nelja testija jaoks arusaadavad ning üks testija kirjutas, et ei ole selge, mis mängu idee on.

Kujunduse ja kasutusmugavuse kohta saadud tagasiside oli vastuoluline. Testijad tõid välja, et raskusastme valikul ei olnud arusaadav, et raskusastet saab kohe valida, kuna valikud ei näe välja nagu nupud. Samuti tekkisid ebamugavused tähtede liikumisega, sest tähed läksid üksteise peale, ning eemaldamisega, sest kindla tähe haaramise jaoks oli vaja enne ümbritsevad tähed liigutada. Testijate arvates on kujundus lihtne ning seda võiks huvitavamaks teha. Üks testijatest kirjutas, et must-valge toon on masendav ning mängul võiks olla rohkem värve.

Testimise käigus tuli välja kaks viga. Esimene jokkeriga seotud viga oli sama, mis tekkis esimesel testimise etapil, kuid erinevate tingimustega. Teine viga on sõnastiku üles laadimisega seotud ning oli ka varem avastatud. Selle vea korral ei tuvasta mõnikord programm faili, kui see oli failisüsteemis valitud. Kuna selle põhjus ei ole teada, on viga parandamata jäetud ning võib esineda mängu praeguses versioonis. Võimalik lahendus kasutaja jaoks on lehe värskendamine ja uuesti proovimine.

5.3. Võimalikud edasiarendused

Saadud tagasisidest selgus, et mäng on hea ja huvitav, kuid vajaks parandamist nii kasutusmugavuse kui ka kujunduse osas. Kasutajad sooviksid, et sõna mittesobivuse korral oleksid mängulauale viimasena lisatud tähed kas näidatud teise värviga või oleks võimalus neid topeltklõpsuga või eraldi nupuga mängulaualt eemaldada. Antud juhul on kõige lihtsam lahendus lisada eraldi nupp. See on kergelt realiseeritav ning oleks ka kasutajate jaoks mugav, kuna saaks tüütu tähtede liigutamise tegevuse ühe klõpsuga asendada.

Teine ettepanek oli lisada mängu vaatesse tähtede koti seisu, et mängija saaks selle põhjal enda strateegiat arendada. Klassikalises lauamängus ei ole mängijatel teada koti seisu ning nad peavad ise, tuginedes olemasolevale infole (tähtede jaotus, käes olevad täheklotsid, mängulaual olevad tähed), ära arvama, mis tähed on kotti jäänud. Antud võimalust saab teha valikuliseks sõltuvalt mängija soovist.

Lisaks pakkus üks testijatest, et temaatilise sõnastiku välja mõtlemine oleks automaatne, kuna enda sõnastiku koostamine võib olla keeruline ülesanne. Selleks on kaks võimalust: mängija valib esitatud teemade hulgast või pakub oma teema välja. Esimesel juhul on sõnastikud eelnevalt kokku pandud, kuid teemade hulk on piiratud. Teisel juhul on teemade hulk piiramata, kuid on vajalik lisatehnoloogiate kasutamine kasutaja pakutud teema mõistmiseks ja sellega seotud sõnade genereerimiseks. Mõlemal juhul on antud ettepaneku arendamine pikemat mõtlemist ja aega nõudev ülesanne.

Tagasisides toodi välja, et tuleks mängu mobiiliseadmete jaoks sobivaks teha, kuna hetkel saab mängida ainult arvuti veebilehitsejas. Samuti kirjutati ka mängu kujunduse arendamise võimalustest. Seda võiks huvitavamaks muuta näiteks värvide, elementidele interaktiivsuse ja menüüdesse temaatiliste piltide lisamisega.

Kokkuvõte

Töö tulemusena valmis veebirakendus, mis võimaldab kasutajal mängida eestikeelset versiooni Scrabble mängust. Mäng on mõeldud ühele mängijale, kelle vastaseks on arvuti. Selle käigu genereerimise algoritm põhineb kohandatud Andrew W. Appeli ja Guy J. Jacobsoni algoritmil. Programmi algoritm kasutab ankruruute sõnade jaoks võimalike positsioonide leidmiseks. Algoritmi teostamisel on loobutud ristkontrolli hulkade kasutamisest, mis võimaldasid sobimatuid sõnu kohe tagasi lükata. Nende puudusuuus seisnes selles, et sobimatute sõnade kontroll oli piiratud. Ristkontrolli hulkade kasutamise asemel on funktsioon, mis kontrollib sõna sobivust, kui see on täiesti leitud. Algoritm kasutab sõnastiku struktuuri esitamiseks suunatud atsüklilise sõnagraafi asemel, mis on oma olemuselt minimeeritud prefiksipuu, tavalist prefiksipuud. See võib olla aeglasem, kuid üldjuhul ei esine käigu genereerimisel pikemaid viivitusi. Algoritmi töö jaoks vajalik baassõnastik põhineb Eesti Keeleressursside Keskuse veebilehelt leiduval lemmade sagedusloendil ning koosneb sõnade algvormidest, käänduvate sõnade puhul ainsuse ja mitmuse nimetavast käändest. Tähtede jaotus ja nende väärtused, mis on vajalik käigu eest saadud punktisumma arvutamiseks, põhinevad eestikeelse Scrabble lauamängu andmetel.

Arvuti käitumist saab muuta, valides ühe kolmest raskusastmest. Sõltuvalt valikust teeb arvuti väheskooriva, juhusliku või parima käigu vastavalt kerge, keskmise ja raske astme puhul. Samuti on lisafunktsionaalsusega lisatud võimalus mängida temaatilist mängu, mille sõnastiku saab kasutaja ise üles laadida. Sõnastiku üleslaadimisel on mõned piirangud: faili formaat ja faili suurus, mis on seatud lihtsamaks faili valideerimiseks. Kõik sõnad kontrollitakse Filosoofi morfalanüsaatori abil ning ebasobivad sõnad eemaldatakse. Kontrollist läbi pääsenud sõnadest koostatakse uus temaatiline sõnastik ning lisatakse hoidlasse, mis tähendab, et fail on kõikidele teistele mängijatele kättesaadav. Temaatilise mängu jaoks saab kasutada ka teiste kasutajate lisatud teemasõnastikke.

Mängu testimise tulemusena said parandatud tõsisemad vead. Kuigi on rohkelt võimalusi rakenduse parandamiseks ja täiendamiseks, on see mängitav ning seega töö eesmärk sai täidetud.

Viidatud kirjandus

- [1] Mattel, Inc. How to play Scrabble, 2006. https://service.mattel.com/instruction_sheets/53639-ENG.pdf (19.03.2024)
- [2] Luuk A., Eestikeelne ristsõnamäng Scrabble'i kujul. Tartu Ülikooli arvutiteaduse instituudi bakalaureusetöö, 2005
- [3] Shapiro S. C., Smith H. R. A Scrabble Crossword Game Playing Program. *Technical Report 119, Department of Computer Science, State University of New York at Buffalo*, 1977, p. 403-419. <https://cse.buffalo.edu/faculty/shapiro/Papers/ubtr119.pdf> (19.11.2023)
- [4] Appel A. W., Jacobson G. J. The World's Fastest Scrabble Program. *Communications of the ACM*, 1988, 31(5), p.572-578. <https://doi.org/10.1145/42411.42420> (19.11.2023)
- [5] Gordon S. A. A Faster Scrabble Move Generation Algorithm. *Software: Practice and Experience*, 1994, 24(2), p. 219-232. <https://doi.org/10.1002/spe.4380240205> (19.11.2023)
- [6] MDN Web Docs. Getting started with the web https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web (28.02.2024)
- [7] OpenJS Foundation. About Node.js®. <https://nodejs.org/en/about> (28.02.2024)
- [8] Phaser Studio Inc. Phaser. <https://phaser.io/> (28.02.2024)
- [9] Notes of Phaser 3. Plugin list. <https://rexrainbow.github.io/phaser3-rex-notes/docs/site/plugin-list/> (28.02.2024)
- [10] EstNLTK. <https://github.com/estnltk/estnltk> (24.04.2024)
- [11] Firebase. <https://firebase.google.com/> (12.04.2024)
- [12] Eesti Keeleressurside Keskus. TÜ arvutilingvistika keeleressurssid. Ressurssid. Uus sagedussõnastik 15 miljoni sõna suuruse korpuse baasil. <https://keeleressursid.ee/et/256-sagedusloendid> (18.03.2024)
- [13] Filosoft. Eesti keele morfanalüsaator. https://www.filosoft.ee/html_morf_et/ (24.04.2024)
- [14] Scrabble Letter Distributions. Estonian https://en.wikipedia.org/wiki/Scrabble_letter_distributions (16.03.2024)

Lisad

I. Küsimustik

Veebimängu testimise küsimustik

Küsimustik on loodud Olga Puksberg informaatika eriala bakalaureusetöö "Eestikeelne veebipõhine Scrabble taoline mäng" testimise jaoks.

olyapuksberg@gmail.com [Switch account](#)



Not shared

* Indicates required question

Testimise juhised

Palun testige veebirakendust. Tegemist on Scrabble taolise mänguga eesti keele jaoks. Mäng asub [siin](#).

Lugege küsimustik enne testimise asumist. Seejärel sooritage mängus järgmised tegevused:

- lugege läbi reeglid;
- mängige tavalist mängu;
- lugege läbi sõnastiku üles laadimise juhised.

Soovi korral saab sooritada ka mõni järgmistest tegevustest:

- mängida tavalist mängu ülejäänud raskusastmetega;
- laadida üles enda temaatiline sõnastik (õnnestumise juhul jääb fail kättesaadavaks kõikidele kasutajatele);
- mängida temaatilist mängu.

Pärast testimist täitke küsimustik. Küsimustiku täitmine võtab aega umbes 10 - 15 minutit.

Testimiseks kasutage arvutit ning soovituslikult Google Chrome veebilehitsejat.

Küsimustik on avatud kuni 03.05.24 (kaasa arvatud).

Mängu raskusaste

Järgnevad küsimused on mängu raskusaste kohta

Valisin raskusastet *

- ☐ Kerge
- ☐ Keskmine
- ☐ Raske

Valitud raskusaste vastas ootustele arvuti käikude suhtes *

- ☐ Nõustun täielikult
- ☐ Nõustun
- ☐ Neutraalne
- ☐ Ei nõustu
- ☐ Ei nõustu üldse

Kuidas võiks raskusastet sobivamaks muuta?

Your answer

Temaatiline mäng

Järgnevad küsimused on temaatilise mängu kohta

Olen temaatilist mängu proovinud *

- ☐ Jah
- ☐ Ei

Kasutasin temaatilise sõnastikuna

- ☐ Teiste kasutajate poolt üles laetud sõnastik
- ☐ Enda üles laetud sõnastik

Mulle meeldis temaatilise mängu valiku võimalus *

- ☐ Nõustun täielikult
- ☐ Nõustun
- ☐ Neutraalne
- ☐ Ei nõustu
- ☐ Ei nõustu üldse

Olen enda sõnastiku üles laadimise võimalust kasutanud *

- ☐ Jah
- ☐ Ei

Sõnastiku üles laadimise juhised on arusaadavad *

- ☐ Nõustun täielikult
- ☐ Nõustun
- ☐ Neutraalne
- ☐ Ei nõustu
- ☐ Ei nõustu üldse

Mis jäi sõnastiku üles laadimise juhistest arusaamatuks? Kuidas saaks juhiseid parandada?

Your answer

Mul tekkisid vead/probleemid enda sõnastiku üles laadimisel

- ☐ Jah
- ☐ Ei

Mis vead/probleemid tekkisid enda sõnastiku üles laadimisel?

Your answer

Mängu reeglid

Järgnevad küsimused on mängu reeglite kohta

Mängureeglid on arusaadavalt kirjutatud *

- ☐ Nõustun täielikult
- ☐ Nõustun
- ☐ Neutraalne
- ☐ Ei nõustu
- ☐ Ei nõustu üldse

Mis jäi mängureeglite juures arusaamatuks? Kuidas võiks reegleid paremini kirjutada?

Your answer

Mängu kujundus

Järgnevad küsimused on mängu kujundusliku poole kohta

Jäin rahule mängu kujundusega *

- ☐ Nõustun täielikult
- ☐ Nõustun
- ☐ Neutraalne
- ☐ Ei nõustu
- ☐ Ei nõustu üldse

Kommentaariid mängu kujunduse kohta

Mis on hästi tehtud? Mis on halvasti tehtud? Mida saaks parandada?

Your answer

Jäin rahule mängu kasutusmugavusega *

- ☐ Nõustun täielikult
- ☐ Nõustun
- ☐ Neutraalne
- ☐ Ei nõustu
- ☐ Ei nõustu üldse

Kommentaari mängu kasutusmugavuse kohta

Millised kohad tekitasid segadust? Kuidas saaks kasutusmugavust paremaks muuta?

Your answer

Mängu toimimine

Järgnevad küsimused on mängu tehnilise poole kohta

Mängu jooksul tekkisid tehnilised vead *

- ☐ Ei tekkinud üldse
- ☐ Mõnikord
- ☐ Sageli

Mis vead tekkisid?

Your answer

Mäng töötab ootuspäraselt *

- ☐ Nõustun täielikult
- ☐ Nõustun
- ☐ Neutraalne
- ☐ Ei nõustu
- ☐ Ei nõustu üldse

Mis ebatavalist juhtus mängu ajal?

Näiteks mõne tegevuse oodatud tulemus erines tegelikust tulemusest

Your answer

Kas soovite veel midagi öelda?

Kommentaari või soovitusi rakenduse kohta

Your answer

Submit

Clear form

II. Litsents

Lihtlitsents lõputöö reprodutseerimiseks ja üldsusele kättesaadavaks tegemiseks

Mina, Olga Puksberg,

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) minu loodud teose „Eestikeelne veebipõhine Scrabble taoline mäng“, mille juhendaja on Krista Liin, reprodutseerimiseks eesmärgiga seda säilitada, sealhulgas lisada digitaalarhiivi DSpace kuni autoriõiguse kehtivuse lõppemiseni.
2. Annan Tartu Ülikoolile loa teha punktis 1 nimetatud teos üldsusele kättesaadavaks Tartu Ülikooli veebikeskkonna, sealhulgas digitaalarhiivi DSpace kaudu Creative Commons'i litsentsiga CC BY NC ND 3.0, mis lubab autorile viidates teost reprodutseerida, levitada ja üldsusele suunata ning keelab luua tuletatud teost ja kasutada teost ärieesmärgil, kuni autoriõiguse kehtivuse lõppemiseni.
3. Olen teadlik, et punktides 1 ja 2 nimetatud õigused jäävad alles ka autorile.
4. Kinnitan, et lihtlitsentsi andmisega ei riku ma teiste isikute intellektuaalomandi ega isikuandmete kaitse õigusaktidest tulenevaid õigusi.

Olga Puksberg

15.05.2024