

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Janar Velleste 193648IAIB

Mitme inimesega koos lahendatav eestikeelne Wordle

Bakalaureusetöö

Juhendaja: Peeter Ellervee
PhD

Tallinn 2022

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: Janar Velleste

01.05.2022

Annotatsioon

Selle töö eesmärk on luua sõnamäng, kus on võimalik ära arvata erineva pikkusega sõnu mitme inimese koostööna. Selle jaoks loodi programm, mis võimaldab inimestel ühes Discordi serveris koos mängida läbi sõnumite saatmise.

Programm kirjutati Pythonis, algul discord.py teegiga ning hiljem disnake teegiga. Töö esimene pool uurib olemasolevaid lahendusi ning võimalikke valikuid suhtlusprogrammi, programmeerimise keele, raamistiku ning sõnastiku suhtes. Töö teine pool kirjeldab programmi täpsest implementatsioonist (sealhulgas tähtede värvimise loogika ning kasutajaliidese mured) ning kasutajatelt saadud tagasisidet.

Töö tulemuseks valmis mäng nimega „Sõnam“, mis võimaldab Discordis käskude abil sõnamängu mängida. Üldiselt vastasid kasutajad, et see mäng pakub originaalset rohkem võimalusi ning et nad soovitsid seda ka oma tuttavatele.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 20 leheküljel, 6 peatükki, 15 joonist, 9 tabelit.

Abstract

Multiplayer Wordle in Estonian

The aim of this Bachelor's thesis is to create a word game, where it is possible to guess words cooperatively. The main considerations for this project were that the game would allow people to play together asynchronously and that it would allow variable length Estonian words. For that a program was written, that allows people in one Discord server to play games together by sending commands to the bot account.

The first half of the thesis looks into existing solutions to play Wordle together and possible choices for which instant messaging program, programming language, library and word list to use. The second half talks about this specific implementation (including things like logic for colouring in letters and problems that cropped up with the user interface) and feedback received from users.

While there are many existing solutions that fulfill some requirements, there aren't any that do everything at once, or which a non-programmer could trivially change to work as needed. The program was written in Python, first using the library discord.py and later moving to disnake.

The result of this work is the program „Sõnam“, which fulfills every requirement that was previously set forth and which makes it possible to play this word game on Discord via slash commands.

The feedback for this game was gathered from volunteers who joined the test server for the game. In general, feedback was positive. Testers said that this game offers more options than the original game of Wordle and that they would recommend this game to their friends.

The thesis is in Estonian and contains 20 pages of text, 6 chapters, 15 figures, 9 tables.

Lühendite ja mõistete sõnastik

API	<i>Application Programming Interface</i> ehk Rakendusliides
EKI	Eesti Keele Instituut
ID	Identifitseeriv kood
TTÜ	Tallinna Tehnikaülikool

Sisukord

1 Sissejuhatus.....	10
2 Taust ja olemasolevad lahendused.....	12
2.1 Olemasolevad lahendused	12
2.1.1 Veebipõhised lahendused	12
2.1.2 Discordipõhised lahendused	12
3 Tehnoloogiate valik	14
3.1 Kiirsuhtluse programm	14
3.2 Programmeerimiskeel	14
3.3 Sõnastik.....	15
4 Realisatsioon	16
4.1 Sõnastiku genereerimine.....	16
4.1.1 Faili sisselugemine ja kirjutamine	16
4.1.2 Otsingu funktsioonid	16
4.1.3 Filtreerimise funktsioonid.....	17
4.2 Mängu loogika	17
4.2.1 Pakkumise meetod	17
4.3 Mängude organiseerimine	20
4.4 Ühendus Discordiga.....	21
4.4.1 Kasutajaliides.....	23
4.5 Edasiarenduse võimalused	24
5 Testimine ja tulemused	26
5.1 Vastajate taust.....	26
5.2 Arvamus mängimiskogemusest	27
5.3 Mängu vajalikkus.....	28
6 Kokkuvõte.....	30
Kasutatud kirjandus	31
Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks.....	32
Lisa 2 – Näidismäng	33

Lisa 3 – K simustik 35

Jooniste loetelu

Joonis 1 Lauamäng Mõttemeister (Mastermind), mis kasutab sarnaseid reegleid.	10
Joonis 2 Näidis tähtede värvimise järjekorrast.	18
Joonis 3 Tähtede värvimise loogika sõna ümberpööramisega.	19
Joonis 4 Tähtede värvimise loogika kahe tsükliga.	20
Joonis 5 Kuidas ruudud peaksid välja nägema vs kuidas ruudud nägid välja ühe kasutaja jaoks.	21
Joonis 6 Kalkkriipskaskude soovitused.	22
Joonis 7 Veateade kui programm ei vasta käsule.	23
Joonis 8 Näidis mängu tulemusest.	23
Joonis 9 Näidis suvalise sobiva sõna küsimisest.	24
Joonis 10 Käsk „uus“.	33
Joonis 11 Käsk „paku karma“.	33
Joonis 12 Käsk „paku aldis“.	33
Joonis 13 Käsk „paku traat“.	34
Joonis 14 Käsk „paku sügav“.	34
Joonis 15 Käsk „paku tänav“ ja lõplik tulemus.	34

Tabelite loetelu

Tabel 1 Eelnev kogemus Wordle'iga	26
Tabel 2 Eelnev kogemus tehnikaga	26
Tabel 3 Eelnev kogemus Discordiga	27
Tabel 4 Mängu mängimisviis	27
Tabel 5 Reeglite arusaadavus	27
Tabel 6 Väljatoodud probleemid	28
Tabel 7 Huvi mängu edasi mängida	29
Tabel 8 Mängu võrdlus Wordle'iga.....	29
Tabel 9 Huvi mängu teistele soovitada.....	29

1 Sissejuhatus

Wordle¹ on veebipõhine sõnamäng, kus mängija ülesandeks on ära arvata salajane sõna. Selleks teeb mängija pakkumisi, mille kohta annab arvuti talle tagasisidet. Tavaliselt on see salasõna üks ingliskeelne viietäheline sõna, mille arvamiseks on mängijal kuus võimalust. Iga pakkumise tulemusena annab arvuti teada iga tähe kohta, kas võrreldes salasõnaga on täht õiges kohas, vales kohas või ei esine üldse salasõnas. Raske režiimi puhul on veel lisareegel, mille kohaselt peab iga järgmine pakkumine sisaldama eelmises leitud õigeid tähti.



Joonis 1 Lauamäng Mõttemeister (Mastermind) [1], mis kasutab sarnaseid reegleid.

Wordle on mängitav üksinda veebilehe kaudu ning hiljem on võimalik mängu tulemust jagada oma tuttavatega. Kuid sellel mängul on mitmed piirangud. Seda ei ole võimalik mugavalt mängida koos sõpradega, on võimalik mängida vaid üks mäng päevas ning ei ole võimalik mängida eri pikkusega eestikeelsete sõnadega. Eesmärk on teha programm, mis lahendab kõik need piirangud korraga ning sobib kasutamiseks ka inimeste poolt, kes ei ole arvutite kasutamise eksperdid.

Käesoleva töö ülesandeks on teha valmis Discordi serveris töötav programm, mille abil on võimalik inimestel ühes serveris koos mängida seda mängu läbi sõnumite saatmise.

¹ <https://www.nytimes.com/games/wordle/index.html>

Selles mängu versioonis on põhirõhk erinevate pikkustega eestikeelsete sõnade ära arvamisel, ning ühte mängu on võimalik mängida pikemalt kui ainult kuus käiku.

Pärast esialgse versiooni valmimist on plaan koguda tagasisidet ja selle põhjal teha parandusi ja täiendusi mängule, milleks võivad olla näiteks lahendamist mugavamaks tegevad käsud või mängude üle statistika pidamine. Seejärel kogutakse uuesti tagasisidet, et teada, kas uus versioon on parem ning kas mäng üldiselt on vajalik.

Antud töö koosneb neljast osast. Esmalt uuritakse Wordle mängu ennast ja olemasolevaid lahendusi probleemile. Teiseks uuritakse, milline tehnoloogia on kõige sobivam ülesande lahendamiseks. Edasi antakse ülevaade projekti kasutajaliidesest ja struktuurist. Viimane osa keskendub projekti tulemusele ja tagasisideküsitlusele.

2 Taust ja olemasolevad lahendused

Wordle sai suures osas populaarseks, kuna iga mängu tulemusena on mängijal võimalik jagada värvilist *emoji*'dest koosnevat ruudustikku oma tulemusest, mille üle on võimalik arutada teiste mängijatega. Mängu vohikutes tekitavad need ruudustikud uudishimu ja kutsuvad mängima. Mängu mängimise teeb huvitavaks see, kuidas see mängib keelega ning pakub võimaluse näha huvitavaid ühendusi sõnade vahel. [2]

2.1 Olemasolevad lahendused

Selle töö eesmärkide täitmiseks on olemas erinevat sorti osalised lahendused, mõned neist ka vabavaralised. Põhiliselt on saadaval kahte sorti lahendusi: veebipõhised ja Discordipõhised.

2.1.1 Veebipõhised lahendused

Veebi kaudu on üksinda võimalik mängida mitmeid erinevaid mängu variante. Muidugi on olemas originaalne Wordle¹, mis on pakub juba sissejuhatuses mainitud reeglistikku. On saadaval ka mängu nagu hello wordl² (eri pikkusega sõnad), Squabble³ (üksteise vastu mängimine) ja Sõnur⁴ (eestikeelsed sõnad). Siiski ei leidu sellist versiooni, mis pakuks mitme mängijaga koos eesti keeles mängimist.

2.1.2 Discordipõhised lahendused

Discordi kaudu on võimalik lisada olemasolevaid programme oma serverisse, ilma et peaks ühtegi programmi ise käima panema. Selline on näiteks Co-ordle⁵, mis küll pakub palju võimalusi, kuid ei ole avatud lähtekoodiga ning ei ole võimalik mängida piiramatult palju mängu.

¹ <https://www.nytimes.com/games/wordle/index.html>

² <https://helloworld.net>

³ <https://squabble.me>

⁴ <https://mountwoolgames.itch.io/ekss>

⁵ <https://top.gg/bot/941726067125784616>

Vabavaralised versioonid, mida on võimalik muuta ja ise jooksutada, on näiteks Discord Wordsy¹ ja discordle², mis on piisavalt piiratud, et tundub lihtsam alustada nullist ja teha ise paremini. Näiteks Discord Wordsy kasutab alati viietähelisi sõnu, mis tähendab, et see loogika on vaja üldistada ükskõik mis pikkusega sõnade jaoks. Discordle pakub küll eri pikkusega sõnu, kuid selle käsuformaat ja tulemuste kuvamine tundub autorile pigem ebamugav.

Kokkuvõttes on inglise keeles võimalik mängida tavalist mängu (Wordle), erinevate sõnapikkustega mängu (hello wordl) või läbi Discordi mängu (discordl). Eesti keeles on ainult võimalik mängida tavalist mängu läbi veebi, kus iga päev on üks sõna (Sõnur). Aga ei ole võimalik mängida eesti keeles eri pikkustega mängu koos mitme inimesega.

¹ https://github.com/Donciop/Discord_Wordsy/blob/main/game.py

² <https://github.com/siddhu33/discordle>

3 Tehnoloogiate valik

Selles peatükis analüüsitakse, milline kiirsuhtluse programm valida, millises programmeerimiskeeles see programm teha ning kuidas valmistada sobilik sõnastik.

3.1 Kiirsuhtluse programm

Suuremad kiirsuhtluse programmid, millega võimalik bot teha: Discord, Facebook Messenger, Slack, Telegram, Viber, Whatsapp. [3]

Facebook Messenger ei ole sobilik selliseks projektiks, sest Facebook kasutab botte pigem teistsugusteks kasutusalaudeks. Facebook eeldab, et igal sellisel programmil on oma leht Facebookis ning üldiselt pigem räägib üks-ühele kasutajatega. [4]

Slack ei ole sobilik, sest see on mõeldud pigem firmasiseseks vestlemiseks ja seal on vaja iga uue grupi jaoks luua eraldi kasutaja. [5]

Ülejäänutest on Discord kõige sobivam mitmel põhjusel. Discordil on head võimalused sellise programmi lisamiseks ning mitmed teegid sellega ühendumiseks. [6] Samuti on seal tõenäoline leida rohkem kasutajaid, kelle peal testida. Näiteks on Discord üks kolmest põhilisest vestluskanalist, mis on autori kogemusel kasutusel TTÜ õppeainete info edastamiseks (teised on Microsoft Teams ja Slack).

3.2 Programmeerimiskeel

Discordiga ühendumiseks on neli põhilist varianti: Python, Java, C# ja JavaScript. [6] Neist sai valitud Python, kuna võrreldes C# või Javaga pakub Python paremini loetavat koodi ning ei vaja eraldi koodi kompileerimist.

Võrreldes JavaScriptiga on Pythonil palju sisseehitatud teeke, millest võib olla sellise projekti puhul kasu. Lisaks on JavaScripti puhul murekohaks selle allalaaditavate teekide usaldusväärsus. Üks suurem teekide haldaja, npm¹, on hiljuti mitmel korral sattunud uudistesse, kuna selle erinevaid teeke on saboteeritud mitmetel põhjustel. [7]

¹ <https://www.npmjs.com>

Spetsiifilise Discordi teegina valiti esialgu discord.py¹ kuna see tundus olevat kõige populaarsem selline teek. Programmi kaldkriipskäskude lisamisel mindi üle teegile disnake², kuna ilmnis asjaolu, et discord.py oli viimati uuendatud eelmise aasta juunikuus [8] ning vahepeal on Discord lasknud välja uue API versiooni (ja on vana välja lülitamas). [9] Üks soovituslikest asendustest oli disnake, mis sai valitud kuna see oli sobiv eelneva koodiga ning oli võimalik nimevahetusega otse sisse asendada.

3.3 Sõnastik

Mängu mängimiseks on vaja sõnade loendit, mille põhjal salasõnu valida ning pakkumisi kontrollida. Selleks on Eesti Keele Instituudi lehel pakkuda mitmeid erinevaid lehekülgi, millest iga üks sisaldab erinevaid loendeid. Nendest lehtedest tundus autorile kõige sobivam sõnaloendite leht³.

Kuna algses mängus on siamaani olnud kõik salasõnad algvormis (ning see tundub ka kõige ausam), siis lähtuti ka eestikeelse sõnastiku valimisel sellest, et oleks võimalusel algvormis sõnad. Seda arvestades tundus lemmade sõnaloend, mis on koostatud sõnastike alusel ja peaks sisaldama põhiliselt algvorme, kõige sobivam. Kuigi soned ja vormid sõnaloendid sisaldasid rohkem sõnu, siis need sisaldasid ka eri käändeid ja pööordeid.

Valitud failis *lemmad2013.txt* oli algselt 178 000 sõna. Sealt filtreeriti välja sellised sõnad, milles esines tähti, mis ei esinenud ülejäänud loendis väga tihti. Need ära jäetud märgid olid sidekriips, numbrid, tähed „c“, „q“, „w“, „x“, „y“, „z“, ning mitmed eritähed nagu „é“. Järgnevalt sorteeriti sõnad pikkuste järgi ning jäeti alles kõik sellised, kus oli vähemalt 1000 sõna. Selliselt saadi 4-18-täheliste sõnadega loendid. Kõige rohkem sõnu oli üheksatäheliste sõnade loendis – 26 211.

¹ <https://discordpy.readthedocs.io/en/stable/>

² <https://docs.dsnake.dev/en/stable/>

³ <https://www.eki.ee/tarkvara/wordlist/>

4 Realisatsioon

Mängu programm on üldiselt jaotatud neljaks: sõnastiku genereerimine, mängu loogika, mängude organiseerimine ja ühendus Discordiga. Siin peatükis mainitakse ka vana ja uut versiooni mängust, kus uuem versioon valmis vana versiooni tagasiside põhjal.

4.1 Sõnastiku genereerimine

Sõnastiku generaator koosneb üheksast funktsioonist, mille abil on võimalik luua algsest sõnade listist lõplikud sobivad failid mängu jaoks. Nende funktsioonide loogika on tavaliselt järgmine: esimesel jookstuamisel loe sisse fail ja leia sobivad parameetrid sõnastike jaoks, teisel jooksutamisel loe fail uuesti sisse, filtreeri vastavate parameetritega ning kirjuta välja uued listid.

4.1.1 Faili sisselugemine ja kirjutamine

Funktsioon *read_in_wordlist* loeb sisse esialgse sõnade nimekirja. See peab olema tekstifail, kus iga sõna on eraldi real. Kuna sõnad loetakse sisse hulk (*set*) andmestruktuuri, siis sõnade järjekord lõplikus nimekirjas ei ole alati sama kui esialgses.

Funktsioon *write_wordlist_to_file* kirjutab sõnastiku etteantud faili samasugusel kujul nagu see sisse loeti.

4.1.2 Otsingu funktsioonid

Kõik funktsioonid, mille nimi algab sõnaga *find*, võtavad sisendiks sisseloetud sõnastiku (välja arvatud *find_all_unused_letters*, mis otsib ise kõik *index.txt* failis mainitud failid läbi) ja aitavad leida sobivad parameetrid filtreerimiseks.

Funktsioonid *find_unique_letters* ja *find_letter_counts* leiavad kõik erinevad tähemärgid, mis esinevad vastavas sõnastikus. See aitab leida üksikuid veidraid tähti nagu „é“, mis esinevad ainult üksikutes sõnades.

Funktsioonid *find_unused_letters* ja *find_all_unused_letters* leiavad üles tähed, mis esinevad muutujas *ALL_LETTERS* ja ei esine üheski sõnastiku sõnas. See on kasulik kui on kasutusel erinevate tähtedega sõnastikud. Näiteks eestikeelses sõnastikus on liiga

vähe sõnu, milles esineb täht „x“, et need sõnad sisse jätta, aga ingliskeelses sõnastikus ei esineks ühtegi täpitähtedega sõna. Need kasutamata tähed tuleb kirjutada *index.txt* faili ning mängu mängides neid tähti ei kuvata võimalike pakkumistena.

Funktsioon *find_word_length* loeb kokku, mitu tükki esineb iga pikkusega sõna. See aitab otsustada, milliste pikkustega sõnastikud on võimalik luua.

4.1.3 Filtreerimise funktsioonid

Kõik funktsioonid, mille nimi algab sõnaga *filter*, võtavad sisse sõnastiku ja filtreerimise parameetri, ning tagastavad filtreeritud sõnastiku(d).

Funktsioon *filter_wordlist_by_character* jätab sõnastikku alles vaid sellised sõnad, mis ei sisalda etteantud tähti. Need tähed on võimalik leida, kui analüüsida eelnevalt funktsiooni *find_letter_counts* väljundit.

Funktsioon *filter_wordlist_by_length* loob sõnastiku (*dict*) üksuse, kus võtmeteks on sõnapikkused ning iga võtme all on hulk (*set*), mis sisaldab vastava pikkusega sõnu.

4.2 Mängu loogika

Mängu loogika paikneb klassis *SMGame*. Selles sisaldub pakkumise meetod *guess* ning erinevad meetodid mängu info kuvamiseks. Iga mäng on erinev *SMGame* objekt.

4.2.1 Pakkumise meetod

Meetod *guess* on üles ehitatud esialgse Wordle mängu jälgimise põhjal. See kontrollib üle, kas pakkumine on korrektne, leiab igale tähele vastava värvi, teeb vajadusel raske režiimi kontrolli ning lõpuks ehitab valmis vastuse teksti.

Värvide loogika on järgmine: hall täht ei esine sõnas, kollane täht esineb sõnas kuskil mujal ning roheline täht on täpselt õigel kohal. Märkimisväärne on veel see, et kui täht esineb pakkumises rohkem kordi kui salasõnas, siis on mitte-halli värvi ainult salasõnaga sama arv tähti. Näiteks kui salasõna on „tuvi“ ning pakkumine on „kiri“, siis saab ainult üks „i“ täht olla värvitud, siin juhul neist viimane, sest rohelised tähed saavad eelise.



Joonis 2 Näidis tähtede värvimise järjekorrast.

Esialgne tähtede värvimise loogika proovis ühe tsükliga käia läbi kõik tähed. See oli keeruline, sest tähe värv võib sõltuda sellest, kas sama täht esineb ka kuskil sõna hilisemas osas. Võttes näiteks jälle eelneva „tuvi“ ja „kiri“, siis alustades sõna algusest võiks naiivselt värvida esimese „i“ tähe kollaseks (kuna see on vales kohas) ja teise halliks (kuna rohkem seda tähte ei esine sõnas). Et seda vältida, luges esialgne lahendus kokku, mitu tükki esines igat tähte salasõnas ning mitu tükki pakkumises. Iga tähe korral said rohelised tähed kindlalt värvitud, kuid kollased said värvitud ainult siis, kui neid oli edaspidi võrdselt salasõnas ja pakkumises. Selline lahendus töötas kollaste arvu suhtes, aga paigutuselt algasid kollased pakkumise paremalt poolt.

Paigutuse parandamiseks sai värvimise loogika muudetud niimoodi, et pakkumine käiakse läbi tagurpidi. Kuid selle tööle saamiseks ei piisanud vaid pakkumise tagurpidi pööramisest. Et kogu loogika töötaks, pidi pöörama ümber pakkumise ja salasõna enne kontrolli, kontrolli sees pidi kõik positsiooni arvutused tegema ümber ning lõpuks pidi tulemuse jälle õiget pidi pöörama.

```

rev = ''.join(reversed(guess))
result = ''

for i, letter in reversed(list(enumerate(guess))):
    self.known_letters['Unused'].discard(letter)
    if letter in letters: # yellow or green
        letters.remove(letter)
        if self.password[i] == letter: # green
            result += self.GREEN
            self.known_letters['Yellow'].discard(letter)
            self.known_letters['Green'].add(letter)
        else: # yellow
            if rev.count(letter, self.word_len-i-1)-1 > letters.count(letter):
                letters.append(letter)
                result += self.GREY
                continue
            result += self.YELLOW
            if letter not in self.known_letters['Green']:
                self.known_letters['Yellow'].add(letter)
    else: # grey
        result += self.GREY
        if letter not in self.password:
            # If letter is guessed more times than it appears, then it's grey
            self.known_letters['Grey'].add(letter)

result = ''.join(reversed(result)) # each result bit MUST be 1 character

```

Joonis 3 Tähtede värvimise loogika sõna ümberpöramisega.

Kuna see loogika oli üsna keeruliseks muutunud, otsustas autor selle uuesti kirjutada, seekord kahe tsükliga. Uus loogika kontrollib esimese tsükliga vaid seda, kas täht on roheline (asub õigel kohal). Teise tsükliga värvitakse värvid vastavalt kollaseks või halliks kui neid esineb veel sõnas kuskil.

```

result = [''] * len(guess)

for i, letter in enumerate(guess):
    self.known_letters['unknown'].discard(letter)

    if self.password[i] == letter: # right letter, right place
        result[i] = GREEN
        letters.remove(letter)
        self.known_letters['yellow'].discard(letter)
        self.known_letters['green'].add(letter)

for i, letter in enumerate(guess):
    # mark everything else as wrong place or not in the puzzle
    if result[i]:
        continue
    if letter in letters: # right letter, wrong place
        result[i] = YELLOW
        letters.remove(letter)
        if letter not in self.known_letters['green']:
            self.known_letters['yellow'].add(letter)
    else: # wrong letter
        result[i] = GREY
        if letter not in self.known_letters['green'] and \
            letter not in self.known_letters['yellow']:
            self.known_letters['grey'].add(letter)

result = ''.join(result)

```

Joonis 4 Tähtede värvimise loogika kahe tsükliga.

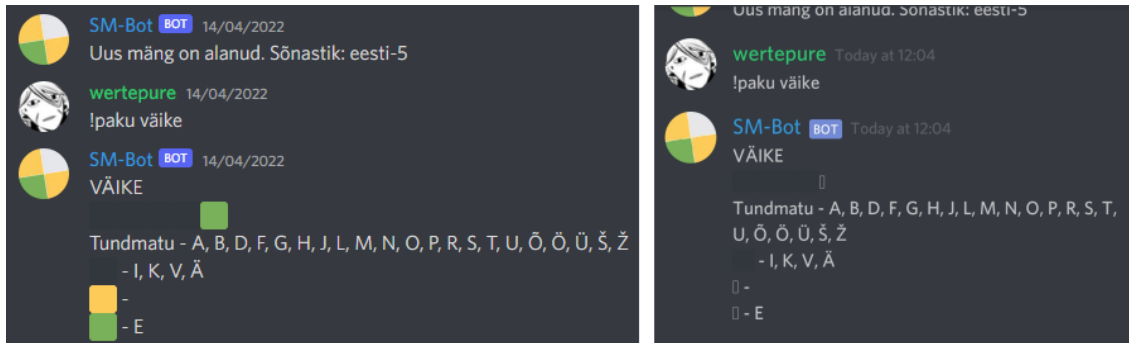
4.3 Mängude organiseerimine

Klass *SMOrganizer* käitub mängu loogika välise kihina. See klass loeb sisse sõnalistid, alustab uusi mängu ning hoiustab poololevaid mängu ja nende sätteid. Mängud ja nende sätted on kättesaadavad täisarvuliste võtmete kaudu, milleks siin programmis on Discordi kanalite ID väärtused. Igal kanalil on eraldi ID ja seega eraldi mäng ja sätted.

Sõnalistide info loeb programm failist *index.txt*, kus on iga sõnastiku kohta kirjas selle asukoht, kasutamata tähed ning selle nimi.

Mängu sätted on salvestatud *pickle* teegi abil faili *SMDdata.bin*. Sätete abil on võimalik muuta järgnevaid mängu osi: kas pakkumist kontrollitakse sõnastikust; kas mäng on raskes režiimis; millise sõnastikuga uut mängu alustada (kui pole ette antud); kas kasutada *emoji*'sid, milliste *emoji*'dega väljendada tähtede värve (tegelikult sobib värvi väljendamiseks ükskõik milline vähem kui 10 tähe pikkune sõne, kuna üks *emoji* võib olla mitu tähemarki pikk). Kui ühtegi sätet pole muudetud, kasutatakse tavasätteid (sõnastiku kontroll, tavaline raskus, suvaline sõnastik, südamate *emoji*'d).

Esialgu olid tavalisteks *emoji*'deks samad värvilised ruudud, mis esialgses mängus, aga ühel mängijal ei kivanud mobiilirakendus neid korrektselt. Peale mõningast testimist leiti, et värvilised südamed on lähim värviline kujund, mis töötab ka tolle mängija telefonis.



Joonis 5 Kuidas ruudud peaksid välja nägema vs kuidas ruudud nägid välja ühe kasutaja jaoks.

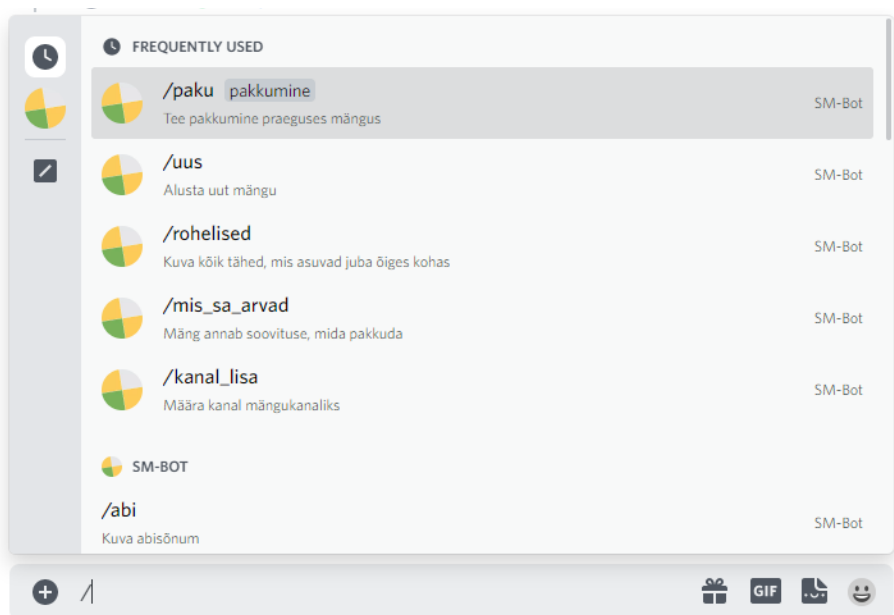
4.4 Ühendus Discordiga

Esialgne versioon ühendusest kasutas teeki `discord.py` ning töötas sõnumite läbilugemisega (*text command*). Iga saadetud sõnumi kohta serveris vaatas programm, kas sõnum algas korrektse käsuga. Kui ei alanud, siis sõnumit ignoreeriti. Kui algas mõne käsuga, siis kutsuti välja vastavad mängu meetodid ning saadeti sõnumiga samasse kanalisse vastus.

Kuna aga Discord on sellist sorti käske vähehaaval välja juurimas [10], siis otsustati mängu teise versiooni jaoks ümber kolida ametlikumale kaldkriipskäskude formaadile (*slash command*). Sellega seoses vahetati ka välja esialgne teek `discord.py` uuema disnake teegi vastu, mis on eelmisest välja hargnenud ning võimaldab kasutada uuemaid võimalusi.

Kui esialgne lahendus oli suures osa teksti parsimine, siis uues versioonis oli võimalik käsud valmis deklareerida ning teha seeläbi kasutaja elu lihtsamaks. Nimelt kaldkriipskäskude puhul on näha otse Discordi aknas kõiki võimalikke käske ja nende kirjeldusi, lisaks on võimalik deklareerida milliseid parameetreid käsk vajab. Seetõttu ei ole (teoorias) isegi võimalik saata vales formaadis käsku programmile. Samuti ei jää kasutaja saadetud sõnum kanalisse eraldi sõnumina. Positiivne on veel see, et kui programmis tuleb ette mõni probleem, siis erinevalt esimest versioonist ei jookse kogu

programm kokku, vaid programm suudab end töös hoida, printides vaid konsooli veateate, mille abil on võimalik viga parandama hakata.



Joonis 6 Kaldkriipskäskude soovitused.

Kaldkriipskäskudega kaasnesid ka mõned negatiivsed küljed. Esiteks pidi programm alati vastama käsule. Kui programm ei vasta, siis Discord kuvab veateate, nagu oleks programm midagi valesti teinud, ning laseb kasutajal uuesti proovida käsku. See ei sobinud kokku mõttega, et seda mängu peaks saama mängida vaid teatud kanalites, seega on igal käsul lisatud juurde rida, mis saadab vajadusel veateate kui kasutajal pole luba vastavat käsku kasutada. Veel ilmselt ühe korra probleem sellega, et programm peab vastama käsule kolme sekundi jooksul (või teatama, et läheb aega). Kui programm ei vasta piisavalt ruttu, siis kasutaja saab taaskord veateate ning isegi kui programm vastab pärast seda, ei tule vastus enam läbi. See probleem ilmselt ühe korra, ilmselt erinevate väiksemate põhjuste kombinatsioonina, kuid on siiski mure, millega võiks edaspidi arvestada. Viimane asi millega arvestada, on see, et käskude muutmisel on vaja anda ette kindlad serverid või oodata mõnda aega, et käsud globaalselt propageeriks Discordis.



Joonis 7 Veateade kui programm ei vasta käsule.

4.4.1 Kasutajaliides

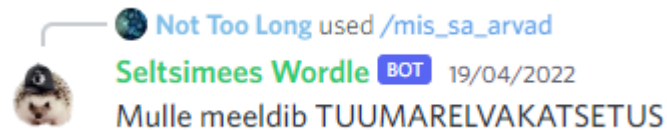
Mängimiseks saadab kasutaja õiges Discordi kanalis korrektse sõnumi, ning neile vastab programm läbi oma Discordi kasutaja mingisuguse vastusega.

Esialgse kasutajaliidese käskude kohta oli võimalik lugeda ainult abikäsust („!abi“). Iga käsk algas hüüumärgiga. Tavamängijale oli saadavalt kolm käsku: loetle saadaval olevad pakid (ehk siin eri sõna pikkused), alusta uut mängu ja tee pakkumine mängus. Kui käsk oli korrektne, saatis programm mingi vastuse. Pakkumise vastus koosnes järgnevatest osadest: pakkumine ise, selle all värve kujutavad *emoji*’d, info tähtede teadaolevatest värvidest. Kui pakkumine oli korrektne vastus, kuvati ka ridade kaupa eelnevate pakkumiste tulemused nagu originaalne mäng annab tulemuseks.



Joonis 8 Näidis mängu tulemusest.

Uus versioon ei muutnud algsete käskude ülesehitust, kuid lisas uusi vastavalt kasutajate tagasisidele. Põhilised uued käsud on *mis_sa_arvad* (mäng annab ette ühe suvalise sõna, mis sobiks pakkumiseks), *ajalugu* (kuvab kõik eelnevad pakkumised ühes sõnumis) ning *rohelist* (kuvab kõik tähed mille asukoht on kindlalt teada ehk mis on rohelist värvi).



Joonis 9 Näidis suvalise sobiva sõna küsimisest.

Oli ka mitmeid väiksemaid muudatusi. Näiteks ei lase uus versioon enam pooleliolevat mängu uuega üle kirjutada enne kui vana on läbi. Samuti sai üle vaadatud kasutaja abistamise võimalused, kuna esialgne abisõnum ei olnud üheselt arusaadav (näiteks mitte-tehnikahuvilised ei tea, et kantsulgude vahel on käsu valikuline osa) ja asus vaid ühes kanalis. Teises versioonis paigutati abisõnumi algusesse näidised põhilistest käskudest. Veel on abiks asjaolu, et kaldkriipskäskude tõttu kuvab Discord käsu trükkimise ajal käsu ja selle parameetrite kirjeldusi.

4.5 Edasiarenduse võimalused

Mäng täidab hetkel kõik etteantud nõuded, kuid arenguruumi veel on. Praegune mängu versioon ei kustuta kunagi kanali sätteid, kuid selline funktsioon oleks väga kasulik. Samuti ei tea programm kui mõni kanal või terve server ei ole enam saadaval ja oleks vaja kustutada. Selle jaoks on hetkel vaja manuaalselt vastavad bin failid ära kustutada, aga programm võiks ise Discordi käest pärida, kas need kanalid on veel saadaval. Veel võiks käskude ja abisõnumite sõnastust kohendada, et need oleks kompaktsemad ja arusaadavamad.

Samuti on võimalik üle vaadata mängu sõnastikud. Mängimise käigus on ilmnenud, et praegune sõnastik sisaldab ka mingil määral sõnu, mis ei ole algvormis või on kuidagi teisiti veidrad. Parem oleks, kui salasõna saaks olla ainult algvormis sõna. Samuti saaks anda sõnade seletusi paremal kujul. Hetkel annab mäng lingi „Eesti keele seletav

sõnaraamat“ otsingule¹, kuid ilmselt oleks ka võimalik otse mingi API vastu päring teha. Peale eesti keele on mängu vabalt võimalik mängida ka mõnes muus keeles, on vaja vaid õigel kujul sõnastik luua.

Edasi vaadates on võimalik palju uusi võimalusi lisada mängule. Näiteks võiks üks mängija valida teistele salasõna arvamiseks. Samuti saaks teha mingisugust statistikat peetud mängude üle. Veel oleks võimalik lisada võimalus mängida muid režiime nagu näiteks Dordle² režiim, kus mängitakse kahte mängu korraga.

¹ <https://www.eki.ee/dict/ekss/>

² <https://zaratustra.itch.io/dordle>

5 Testimine ja tulemused

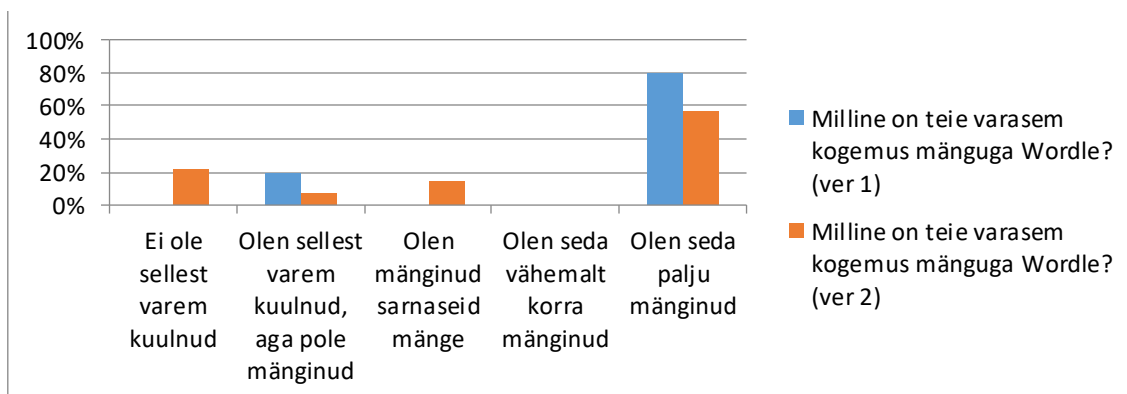
Mängu tagasiside saamiseks loodi testimise Discordi server, kuhu said liituda vabatahtlikud testijad. Kokku liitus serveriga 18 kasutajat. Esimesele versioonile andis tagasisidet 10 vastajat ning teisele versioonile 14 vastajat.

5.1 Vastajate taust

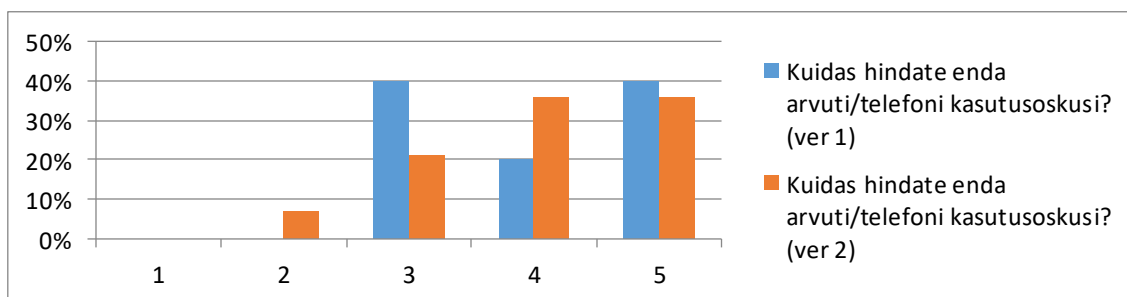
Programmi esimese versiooni kohta andsid tagasisidet üldiselt originaalse mängu fännid, kes on tehnikaga üsna tuttavad, kuid kelle Discordi oskused on seinast sein. Mängu mängiti veidi rohkem mobiilirakenduse kaudu kui arvuti kaudu.

Teise versiooni tagasisides oli rohkem võhikuid originaalmängule, nende tehnikateadmised olid umbes samasugused. Samuti oli mobiili ja arvuti tasakaal veidi rohkem mobiilide poolel.

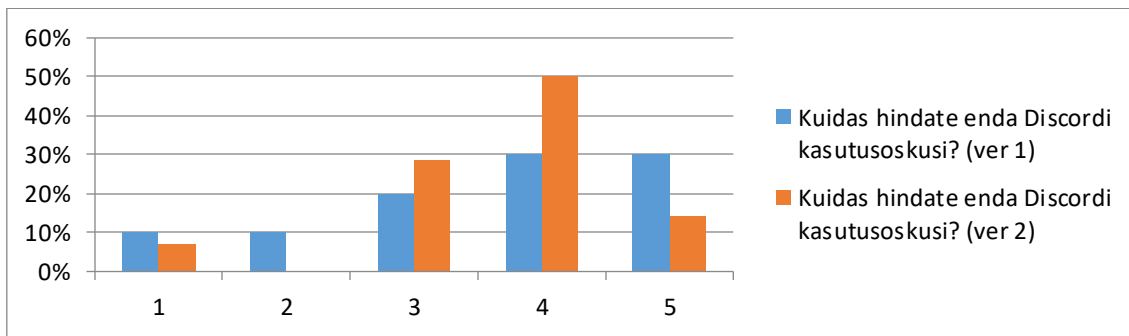
Tabel 1 Eelnev kogemus Wordle'iga



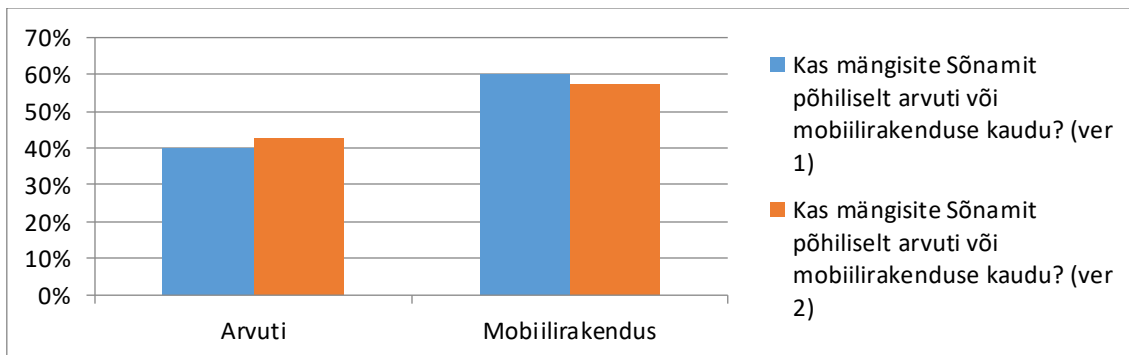
Tabel 2 Eelnev kogemus tehnikaga



Tabel 3 Eelnev kogemus Discordiga



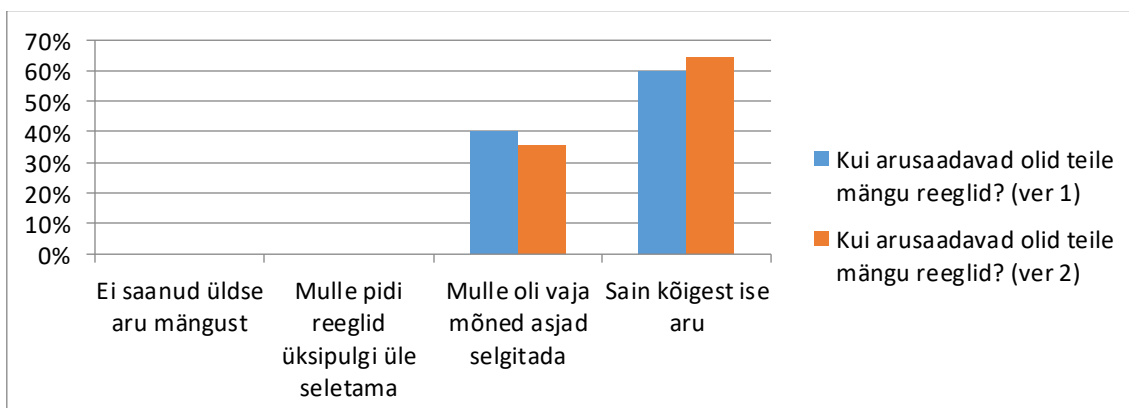
Tabel 4 Mängu mängimisviis



5.2 Arvamus mängimiskogemusest

Mõlemil korral ütles umbes 60% vastajatest, et said kõigest ise aru ning ülejäänud vajasis mõningast selgitamist. Siinkohal peab märkima, et inimesed, kes ei saanud mängust üldse aru, tõenäoliselt ei jõudnud nii kauguele, et vastata küsimustikule.

Tabel 5 Reeglite arusaadavus

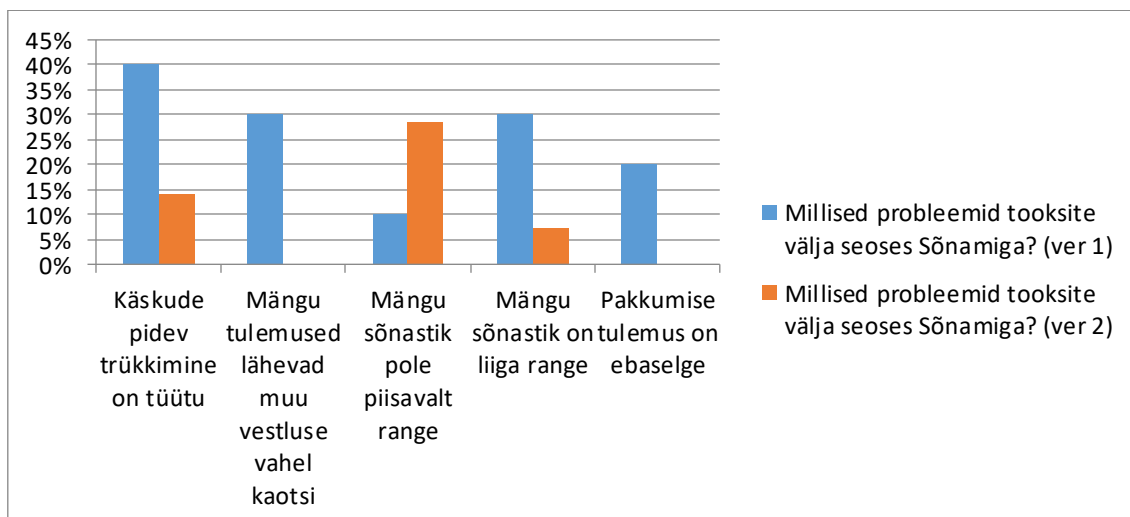


Väljatoodud probleemidest olid esimeses osas kõige tähtsamad käskude pidev trükkimine, mängu tulemuse kaotamine ning mängu sõnastiku rangus.

Nendele probleemidele vastuseks olid kaldkriipskäsud ja *mis_sa_arvad* käsk. Kaldkriipskäskudel on olemas automaattäite võimalus ja seega ei pea iga kord tervet käsku välja trükkima. Samuti kaldkriipskäskude puhul ei jää algne käsu sõnum nii suurelt sõnumite ajalukku ja seega on mängu tulemused rohkem nähtavad. *Mis_sa_arvad* käsk annab mängijale suvalise sobiva sõna, mis aitab siis kui mängijal ei tule ühtegi sobiva pikkusega sõna mõttesse ja ta hakkab suvalisi liitsõnu kokku sobitama.

Teise poole tagasisides kõik need mured vähenesid, aga toodi välja, et mängu sõnastik pole piisavalt range. Seda ilmselt tuli ette rohkem kuna jõuti mängida rohkem mängu ja salasõnana tuli ette rohkem käändes sõnu.

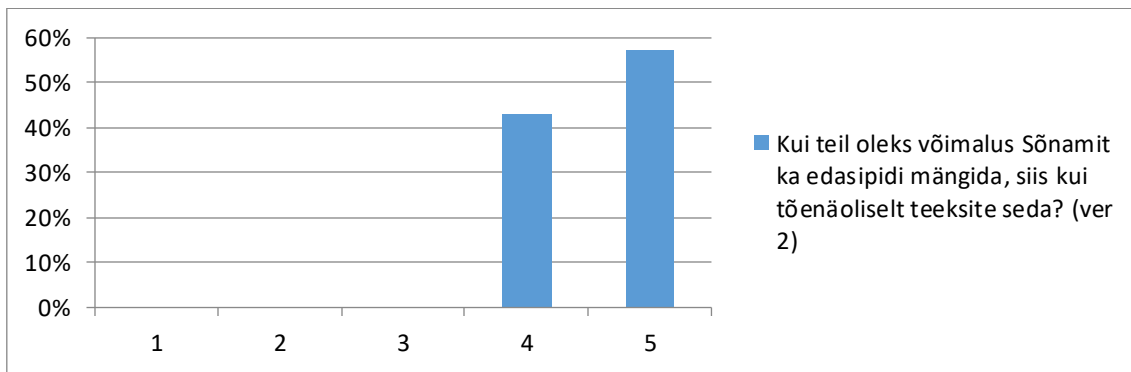
Tabel 6 Väljatoodud probleemid



5.3 Mängu vajalikkus

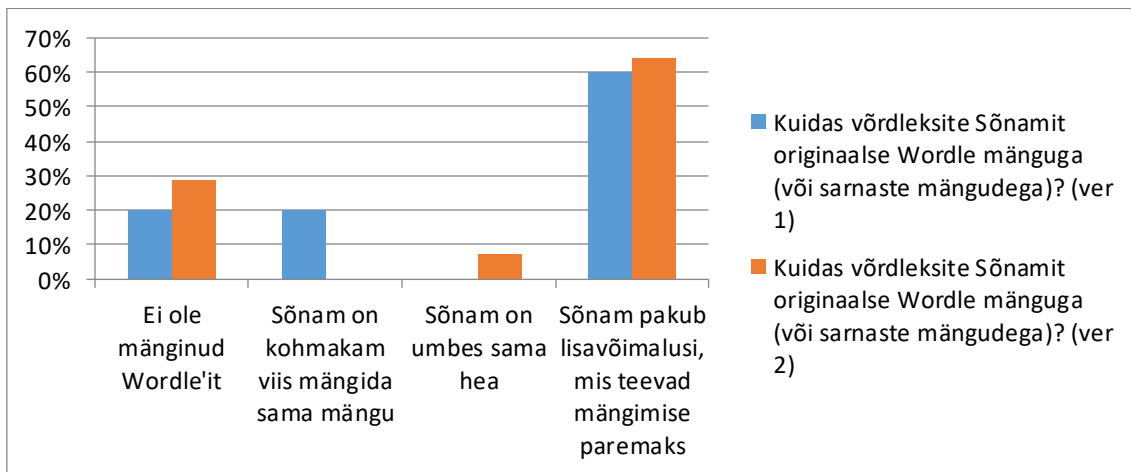
Kõik vastajad ütlesid, et oleks huvitatud selle mängu mängimisest ka edaspidi (esimeses küsimustikus oli see küsimus jah/ei küsimus, millele kõik vastasid jah).

Tabel 7 Huvi mängu edasi mängida



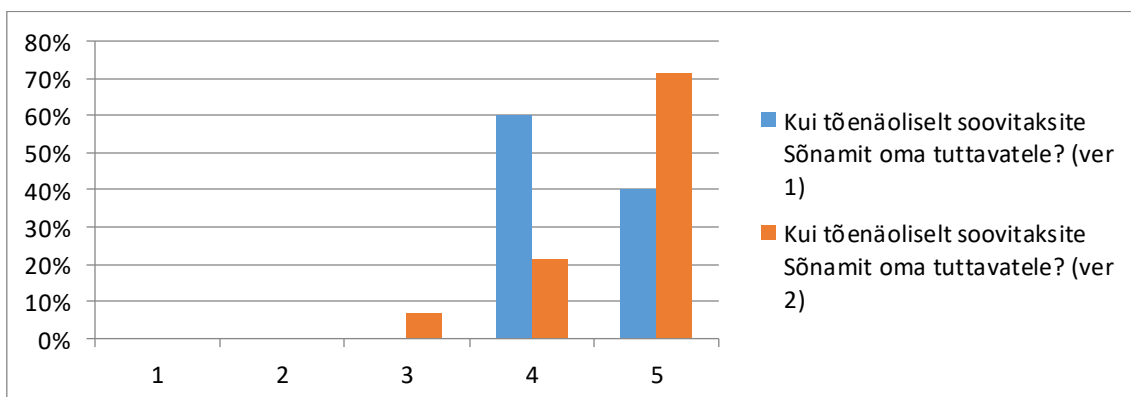
Võrreldes originaalse mänguga oli esimeses pooles tagasiside pigem positiivne, kuid teises pooles arvasid kõik vastajad, et uus versioon on sama hea või parem kui originaal.

Tabel 8 Mängu võrdlus Wordle'iga



Mõlema versiooni puhul vastasid mängijad, et üsna tõenäoliselt soovitaksid seda mängu oma tuttavatele.

Tabel 9 Huvi mängu teistele soovitada



6 Kokkuvõte

Töö eesmärgiks oli luua programm, mille abil oleks võimalik mängida läbi mõne kiirsuhtlusprogrammi sõnamängu, mille reeglid põhinevad mängul Wordle.

Töö käigus uuriti välja, millised võimalused on sellise programmi tegemiseks ning otsustati teha programm Pythoni keele abil Discordi suhtluskeskkonna jaoks. Valmis programm kasutab EKI lemmade nimekirja, mida on muudetud mängule sobivaks.

Programm võtab vastu Discordi kaldkriipskäske (*slash command*), ning reageerib neile vastavalt. Mängu jaoks võtab programm suvalise sõna vastava pikkusega sõnade loendist. Iga pakkumise peale arvutab programm välja värvidest koosneva tulemuse, millest see konstrueerib vastuse ja saadab selle mängijale.

Esimese programmi versiooni tagasiside põhjal tehti programmi juhendit selgemaks, tehti käskude kasutamise lihtsamaks ning selgemaks ja lisati võimalusi teha mängu lihtsamaks. Tänu tehtud muudatustele oli teisel korral tagasiside parem ning kaebusi vähem.

Töö tulemus täidab kõik esialgsed nõuded ning sai tagasisidena positiivsed arvustused. Edaspidi võiks otsida välja parema sõnade loendi, mis paneks vastuseks vaid algvormis sõnu. Samuti on ruumi mitmesuguste muude mängude jaoks, mis töötaksid algse mängu peal.

Kasutatud kirjandus

- [1] ZeroOne, Artist, *A sample game of Mastermind*. [Art]. 2005.
- [2] K. Chayka, „Why You Can't Resist Wordle,“ *The New Yorker*, 20 Jaanuar 2022. [Võrgumaterjal]. Available: <https://www.newyorker.com/culture/infinite-scroll/why-you-cant-resist-wordle>. [Kasutatud 1 Mai 2022].
- [3] discover.bot, „A Comprehensive Guide to Chatbots in Messaging Apps,“ discover.bot, 30 Aprill 2019. [Võrgumaterjal]. Available: <https://discover.bot/bot-talk/guide-to-messaging-apps-chatbot/>. [Kasutatud 1 Mai 2022].
- [4] Meta, „Introduction to the Messenger Platform,“ Meta, 2022. [Võrgumaterjal]. Available: <https://developers.facebook.com/docs/messenger-platform/introduction/>. [Kasutatud 1 Mai 2022].
- [5] Slack, „Join a Slack workspace,“ Slack Technologies, 2022. [Võrgumaterjal]. Available: <https://slack.com/help/articles/212675257-Join-a-Slack-workspace>. [Kasutatud 1 Mai 2022].
- [6] WriteBots, „How to Make a Discord Bot in 2022,“ WriteBots, 14 Veebruar 2022. [Võrgumaterjal]. Available: <https://www.writebots.com/how-to-make-a-discord-bot/>. [Kasutatud 1 Mai 2022].
- [7] A. Sharma, „Third npm protestware: 'event-source-polyfill' calls Russia out,“ *Bleeping Computer*, 11 Aprill 2022. [Võrgumaterjal]. Available: <https://www.bleepingcomputer.com/news/security/third-npm-protestware-event-source-polyfill-calls-russia-out/>. [Kasutatud 1 Mai 2022].
- [8] rapptz, „discord.py Release history,“ Python Package Index, 2022. [Võrgumaterjal]. Available: <https://pypi.org/project/discord.py/#history>. [Kasutatud 1 Mai 2022].
- [9] Discord, „API Reference,“ Discord, 2022. [Võrgumaterjal]. Available: <https://discord.com/developers/docs/reference#api-versioning>. [Kasutatud 1 Mai 2022].
- [10] Weh, „Message Content Privileged Intent FAQ,“ Discord, Aprill 2022. [Võrgumaterjal]. Available: <https://support-dev.discord.com/hc/en-us/articles/4404772028055-Message-Content-Privileged-Intent-FAQ>. [Kasutatud 1 Mai 2022].

Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks¹

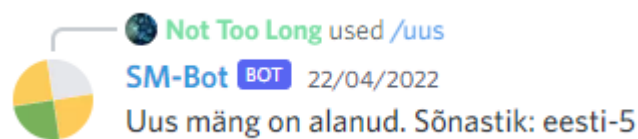
Mina, Janar Velleste

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

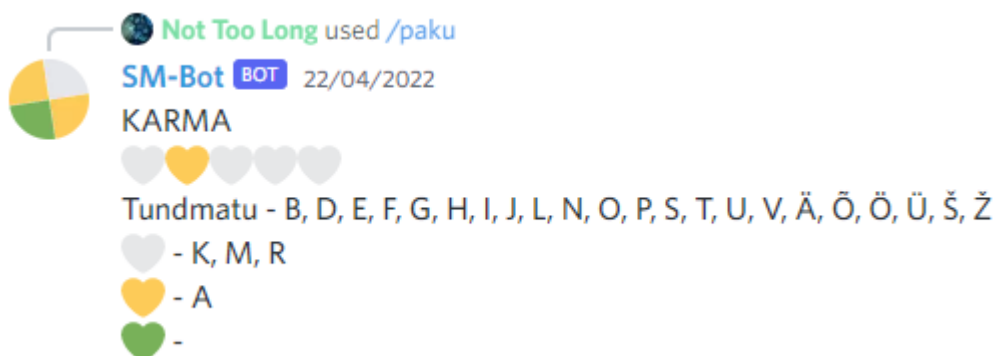
01.05.2022

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

Lisa 2 – Näidismäng



Joonis 10 Käsk „/uus“.



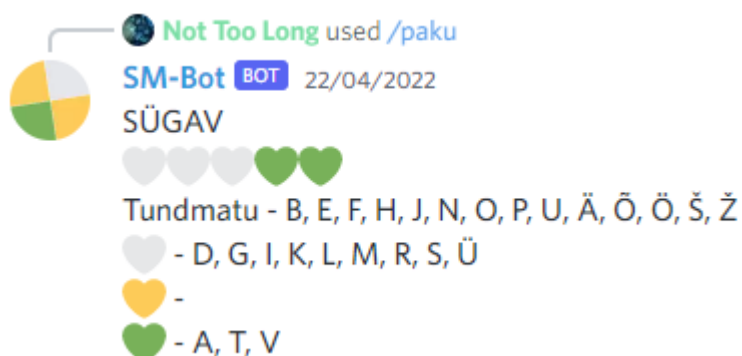
Joonis 11 Käsk „/paku karma“.



Joonis 12 Käsk „/paku aldis“.



Joonis 13 Käsk „/paku traat“.



Joonis 14 Käsk „/paku sügav“.



Joonis 15 Käsk „/paku tänav“ ja lõplik tulemus.

Lisa 3 – Küsimustik

Milline on teie varasem kogemus mänguga Wordle?

Sarnased mängud on näiteks Mõttemeister (ehk Mastermind), Jotto, Lingo jms

- Ei ole sellest varem kuulnud
- Olen sellest kuulnud, aga pole mänginud
- Olen mänginud sarnaseid mängu
- Olen seda vähemalt korra mänginud
- Olen seda palju mänginud

Kuidas hindate enda arvuti/telefoni kasutusoskusi?

- (Variandid 1-5 (Väga halb – Väga hea))

Kuidas hindate enda Discordi kasutusoskusi?

- (Variandid 1-5 (Väga halb – Väga hea))

Kas mängisite sõnamit põhiliselt arvuti või mobiilirakenduse kaudu?

- Arvuti
- Mobiilirakendus

Kui arusaadavad olid teile mängu reeglid?

- Ei saanud üldse aru mängust
- Mulle pidi reeglid üksipulgi üle seletama
- Mulle oli vaja mõned asjad selgitada
- Sain kõigest ise aru

Millised probleemid tooksite välja seoses Sõnamiga? (sai valida mitu tükki)

- Käskude pidev trükkimine on tüütu
- Mängu tulemused lähevad muu vestluse vahel kaotsi
- Mängu sõnastik ei ole piisavalt range
- Mängu sõnastik on liiga range
- Pakkumise tulemus on ebaselge
- Muu...

Kui teil oleks võimalus Sõnamit ka edaspidi mängida, siis kui tõenäoliselt teeksite seda?

- (Variandid 1-5 (Kindlasti mitte – Väga tõenäoliselt))

Kuidas võrdleksite Sõnamit originaalse Wordle mänguga (või sarnaste mängudega)?

- Ei ole mänginud Wordle'it
- Sõnam on kohmakam viis mängida sama mängu
- Sõnam on umbes sama hea
- Sõnam pakub lisavõimalusi, mis teevad mängimise paremaks

Kui tõenäoliselt soovitaksite Sõnamit on tuttavatele?

- (Variandid 1-5 (Kindlasti mitte – Väga tõenäoliselt))

On teil soovitusi mängu veelgi paremaks tegemisel?

- (Vaba tekst)

On mingeid muid kommentaare?

- (Vaba tekst)