TARTU ÜLIKOOL
Arvutiteaduse instituut
Informaatika õppekava

Joosep Hubel

Eestikeelne sõnamäng Ühendused

Bakalaureusetöö (9 EAP)

Juhendaja(d):
Sven Aller, MSc

Tartu 2025

Eestikeelne sõnamäng Ühendused

Lühikokkuvõte:

Selle bakalaureusetöö eesmärk oli luua eestikeelne sõnamäng, mis aitaks õppida uut sõnavara ja mõista sõnade vahelisi seoseid. Töö alguses tutvustati keelemängude rolli keeleõppes ning käsitleti mänge, mis olid antud töö inspiratsiooniks. Praktilises osas loodi automaatselt sõnade vahelised seosed, mille põhjal valmis veebileht, kus saab loodud mängu mängida. Mängu eesmärgiks on leida neli sõnade vahelist seost kuueteistkümne sõna seast. Mängu raskusaste on varieeruv, mistõttu sobib see nii eesti keele õppijatele kui ka emakeelsetele kasutajatele.

Võtmesõnad: keelemäng, keeleõpe, definitsioonid, sõnade seosed

CERCS: P175 Informaatika, süsteemiteooria

Estonian word game Ühendused

Abstract:

The aim of this bachelor's thesis was to create an Estonian word game that would assist with learning new vocabulary and understanding the connections between words. The theoretical part of the thesis introduces the role of language games in language learning and discusses several games that served as inspiration for this project. In the practical part, word connections were generated automatically, forming the basis for a browser game. A website was created where users can play the game. The objective is to identify four word groups from a set of sixteen words. The game's difficulty varies, making it suitable for both Estonian language learners and native speakers.

Keywords: language game, language learning, definitions, word connections

CERCS: P175 Informatics, systems theory

Sisukord

Sissejuhatus

Statistikaameti andmetel rääkis 2021. aastal eesti keelt 84% Eesti elanikkonnast [1]. Kuigi eesti keel on riigi ainus riigikeel [2], viitab see näitaja sellele, et märkimisväärne osa elanikkonnast ei valda seda. Arvestades, et Eesti elanikkond oli samal aastal ligikaudu 1,3 miljon inimest [1], tähendab see, et hinnanguliselt umbes 200 000 inimest Eestis ei räägi eesti keelt. Väike keeleoskajate arvu tõttu peab keskenduma keele püsivuse peale.

Eesti keele püsimise tagamiseks on vaja mitmekesiseid keeleõppe võimalusi ja platvorme. Üks viimasel ajal populaarseks saanud meetod õppida keeli on keelemängud. Klimova ja Kacet [3] töös on kirjas uuringuid selle kohta, kuidas digitaalsete mängude kasutamine aitas mõnes olukorras oluliselt kaasa võõrkeele sõnavara omandamisele.

See bakalaureusetöö on rakenduslik uurimustöö, mille käigus luuakse eesti keele keeleressursse kasutatav sõnamäng Ühendused, mis võtab inspiratsiooni inglise keelsest mängust Connections. Mängu eesmärk on õpetada uusi sõnu, sõnade vahelisi seoseid ja sõnade definitsioone läbi meelelahutusliku mängu.

Töö on jagatud neljaks peatükiks. Esimene peatükk räägib mängust Connections, mis inspireeris projekti ja sõnamängude olulisusest keeleõppes. Teises peatükis on kirjas, kuidas andmeid saadakse Eesti Wordnetist ja töödeldakse mängu jaoks. Kolmandas peatükis keskendutakse veebirakendusele, mille kaudu mäng on kättesaadav. Selles peatükis kirjeldatakse mängu jaoks vajalikku eeltöötlust ja mängu loogikat. Viimases peatükis analüüsitakse mängu kohta kogutud tagasisidet ja edasiarendamise võimalusi. Töö kirjutamises kasutati tehisintellekti GPT-4o grammatika ja vormistuse parandamiseks.

1. Keelemängud

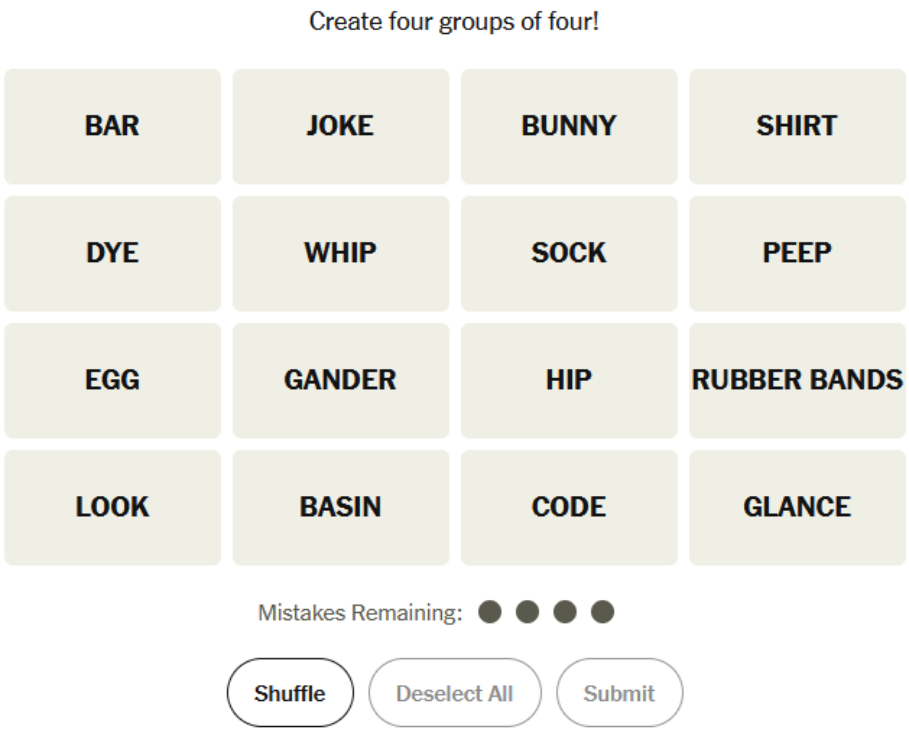
Gupta M., Kumar R., Kumar U. ja Singh A. töös [4] on kirjas, kuidas keeleõppel on sõnade tundmine kriitilise tähtsusega ning keelemängud pakuvad selle saavutamiseks huvitavat ja kaasahaaravat platvormi. Läbi mängude õppimine teeb tegevuse lõbusamaks, mis aitab õppijal olla motiveeritud. Nende töö lõpus oleva uurimuse põhjal on näha, et õpilased suutsid sõnamängu mängides saavutada paremaid tulemusi keeleõppes. Klimova ja Kacet [3] töö toob välja keelemängude kasutamise tugevaid ja nõrkki külgi keeleõppes. Töös vaadatud uuringutest ilmnes ainult ühes negatiivne tagajärg sõnavara omandamisel arvutimängude mängimisest. Autorite arvamusel olid arvutimängud eriti head sõnavara omandamisel ja kuigi teema vajab veel uurimist, siis arvutimängud on üldiselt ikkagi kasulikud keeleõppes.

1.1 Only Connect

Only Connect on briti mängusaade, kus mängijad peavad leidma seoseid erinevates mängu formaatides [5]. Mäng koosneb neljast osast: Connections, Sequences, Connecting Wall ja Missing Vowels [5]. Connecting Wall käigus on antud 16 sõna, mille vahel peab grupp mängijaid leidma neli seost [6].

1.2 Connections

Connections on sõnade vaheliste seoste leidmise mäng, kus mängija leiab sõnagruppe, mis keskenduvad semantikale, Ameerika Ühendriigi kultuurile, sõnade sarnasusele tähtede poolest ja poolikutele fraasidele [7]. Mängul Connections on väga palju sarnasusi mänguga Only Connect [6]. Mängu ülesehituseks on 16 sõnaline ruudustik (vaata Joonis 1).



Joonis 1. Mäng Connections[1]

Tornoe kirjutab enda artiklis [8] kuidas Connections on New York Timesi poolt loodud mängude seast populaarsuselt teisel kohal , jäädes ainult nende teise sõnamängu, Wordle, taha. Samamoodi nagu Wordle, on võimalik ka mängus Connections enda lõpptulemust jagada.

1.2.1 Reeglid

Connections[1] mängu alguses on esitatud mängijale 16 sõna, mis on neljas veerus ja neljas reas ning mängija lõppeesmärk on leida neli temaatilist gruppi, mis koosnevad neljast sõnast. Sõnad kuvatakse esialgu juhuslikus järjekorras, kuid kasutaja saab need nupuvajutusega uuesti segada, et mängu saada mängust uus vaatepilt. Sõnarühmad on jagatud värvideks: kollane, roheline, sinine ja lilla. Samuti on iga rühm eri raskusastmega, kus kollane on kõige lihtsam, roheline järgmine, sinine eelviimane ning lilla kõige keerulisem. Mängu jooksul saab kasutaja märkida ühte rühma kuuluvaid sõnu ja peale täpselt nelja sõna valimist on võimalik tal vajutada esitamise nupule, mille puhul mäng kontrollib, kas valitud sõnad kuuluvad õigesse sõnagruppi. Kui kasutaja on neli sõna valinud, mis kõik kuuluvad ühte sõnarühma, siis mäng viib need sõnad kõige tippu, näitab kasutajale, millise teema alla sõnad kuuluvad, ja märgib need vastava

[1] https://www.nytimes.com/games/connections

raskusastme värviga. Mängu jooksul on lubatud valesti arvata neli korda ja kui kasutaja esitab enda vastuse, aga tema valitud sõnad ei kuulu koos ühte rühma, siis läheb kirja üks viga. Kui kasutaja on täpselt kolm sõna ühest rühmast õigesti pakkunud, siis annab mäng vihje, näidates teadet, et kasutajal on ainult üks sõna valesti. Mäng lõpeb võiduga siis, kui mängija on kõik sõnagrupid õigesti ära arvanud. Kui kasutaja teeb neli viga, on mäng läbi ja ta näeb kõiki teemasid koos õigete vastustega. Mängu lõpus saab kasutaja vajutada jagamise nupule, et kopeerida emotikonidest loodud tabel, kus on näidatud kõik kasutaja mängu jooksul tehtud käigud teema värvide järgi. See võimaldab kasutajal jagada mängus saavutatud tulemust üle erinevate sotsiaalmeedia rakenduste.

1.3 Sõnamängu Ühendused loomise nõuded

Mängu Ühendused loomiseks on vaja määrata selle nõuded. Nõuded keskenduvad peamiselt mängu funktsionaalsusele ja kasutajasõbralikkusele. Kuna töö eesmärk on luua hariduslik keelemäng, tuleb nõuetes arvestada ka keeleõppe aspektidega. Seega on mängule seatud järgmised nõuded:

1. Mängus peab olema neli sõnarühma, millest igaüks koosneb neljast omavahel seotud sõnast.
2. Mängija peab saama valida neli sõna ja esitada enda vastuse.
3. Õige vastuse korral peab mäng näitama kasutajale vastuse teemat ja sellega seotud sõnu.
4. Vastus peab olema värvitud selle raskusastme järgi.
5. Mängija peab saama vaadata vastuses esinevate sõnade definitsioone, et õppida uusi sõnu.
6. Pärast nelja vale vastust peab mäng lõppema ja kuvama kõik vastused, mida mängija ei leidnud.
7. Mängul peab olema nupp, millega saab sõnade asukohti segada, et pakkuda kasutajale uut vaatepilti.
8. Vale vastuse korral peab mäng kontrollima, kas vähemalt kolm pakutud sõna on omavahel seotud, ning sellisel juhul andma mängijale vihje.
9. Mängus peab olema vähemalt neli erinevat tüüpi seoseid sõnade vahel.
10. Iga päev peab olema loodud uus mäng, mis koosneb 16 sõnast.
11. Mängus ilmuvad sõnad peavad olema automaatselt genereeritud, et töö autor ei peaks iga päev käsitsi uusi sõnarühme looma.

12. Automaatselt genereeritud informatsioon mängus peab olema korrektne, sest keeleõppes on informatsiooni täpsus väga oluline.
13. Mängu kasutajaliides peab laadima maksimaalselt kahe sekundiga.
14. Mängu lõpus peab kasutajal olema võimalus jagada oma tulemust.

Nõuete järgimine tagab, et mäng on kasutajasõbralik, loogilise ülesehitusega ja toetab mängija keeleõpet.

2. Eeltöötlus

Mängu Ühendused jaoks on vajalik informatsiooni sõnade vahelistest seostest. Selle saavutamiseks on kasutatud Eesti Wordnetti. Eesti Wordnet [9] on eestikeelne kollektsioon sünonüümide hulkadest, mis kasutab Princetoni WordNeti ja EuroWordNeti põhimõtteid. Sünonüümihulk ehk sünohulk on ühe mõistega seotud sõnade rühm. Eesti Wordnetis on ühendatud sünohulgad omavahel semantiliste ja leksikaalsete reeglite põhjal nagu sünonüümid, meronüümid, hüponüümid, antonüümid ja nii edasi. Seega pakub Eesti Wordnet kõike vajalikku informatsiooni mängu Ühendused loomiseks.

Kuna Eesti Wordneti sünohulgad sisaldavad väga palju informatsiooni, millest kõik ei ole mängu loomiseks vajalik, tuleb andmeid esmalt töödelda. Näiteks kui sõnal on vähem kui neli seost, siis me ei saa seda mängu loomiseks kasutada. Sobivate andmete loomiseks on loodud automaatne töötlusprotsess, mis filtreerib välja sobimatud sõnad, loob nelja sõnalised grupid, arvutab seoste raskusastmed ja seob sõnad nende definitsioonidega.

2.1 Sobivate ühenduste leidmine

Algoritmi alguses sobivate sõnade leidmiseks on kasutatud Eesti Wordnetti ja sõnade sagedussõnastikku. Kuna Eesti Wordnetis on kokku umbes 90 000 sõna ning andmete töötlemine on ajamahukas, on ülesande mõistliku aja jooksul lahendamiseks esmalt valitud välja need sõnad, mis esinevad nii Wordnetis kui ka sagedussõnastikus korraga. Selline lähenemine on vajalik, kuna enamik sõnu, mida sagedussõnastikus ei esine on väga spetsiifilised ja seega need ei sobi sõnagruppi ülemsõnaks, sest nendel puuduvad mängu jaoks valitud seosed. Seega kasutades sagedussõnastikku saab vältida suur osa sünohulke, millel puudub rohkem kui neli seost valitud teemadel. Sõnu, mida sagedussõnastikus ei leidu, ei kasutata baassõnadena, kuigi need võivad mängus siiski esineda. Selles projektis on sõnade vaheliste seostena valitud järgmised neli teemat:

- Sõnade sünonüümid.
- Sõnade meronüümid, ehk osad. Näiteks on sõna auto meronüümiks eesiste.
- Sõnade hüponüümid, ehk alamsõnad. Näiteks on sõna auto hüponüümiks diiselauto.
- Sõnad, millele on üks või kaks tähte juurde lisatud. Näiteks sõnaga kiir käib kokku sõna kiirus.

Teemade valik selles projektis sai tehtud eesmärgiga tagada, et mäng oleks keeleliselt rikas ja küllalt keeruline ka kogenud sõnamängude mängijatele. Viimane teema nimekirjas sai valitud

just selleks, et tekitada suuremat varieeruvust mängus, mis teeb mängus sõnade vaheliste seoste leidmise keerulisemaks.

Algoritmi järgmiseks etapiks on leida kõik sõnade vahelised seosed, mis mängu jaoks on kasulikud. Samuti ei ole kasulikud seosed, mis koosnevad vähem kui neljast sõnast. Iga sõna jaoks on loodud massiiv kõigist ühendatud sõnadest ja eraldi järjendid igat erinevat tüüpi seose jaoks. Nende loomiseks on kõigepealt iga sõna juures loetud sisse kõik selle sõnaga seotud sünohulgad Eesti Wordnetist. Sünohulkadest on eraldatud informatsioon sünonüümidest, meronüümidest ja hüponüümidest. Sõnu millel on paar tähte lõppu lisatud saab leitud võrreldes sõnu kõigi teiste võimalike sõnadega kasutades järgmist tekstitöötlust:

- Kontrollitakse, kas üks sõna on üks või kaks tähte pikem kui teine.
- Lõigatakse pikemalt sõnalt viimased paar tähte maha ja vaadatakse, kas alles jäänud sõnad on võrdsed.

Järjend kõigist ühendatud sõnadest on kasutusel peamiselt programmi optimeerimiseks, kuna kui sõnal on vähem kui neli seost, siis ei ole võimalik seda kasutada mängus oleva neliku loomisel ja seda sõna võib ignoreerida. Loodud sõnastik salvestatakse JSON faili, kuna genereerimine on pikk protsess, siis on modulaarne disain ohutum, kui midagi vahepeal valesti läheb.

Algoritmi jaoks on vaja eestikeelset sagedussõnastikku, kus sõnad on järjestatud esinemissageduse alusel kahanevas järjekorras. Projekti jaoks valiti sagedussõnastik, mis põhineb tasakaalus korpusel ja sisaldab lemmade sagedusi koos esinemiskordade arvuga. Tasakaalus korpus hõlmab kolme tekstiliiki: ajakirjandust, ilukirjandust ja teaduskirjandust, mis tagab eesti keele laiapõhjalise esindatuse. Eesti Keeleressursside Keskuse veebilehel[2] on olemas ka spetsiifilisemad sagedussõnastikud, mis keskenduvad üksnes ajakirjandusele, ilukirjandusele või teaduskirjandusele, kuid lõppvalik langes tasakaalus korpuse põhjal koostatud sagedussõnastiku kasuks, kuna see katab kõige mitmekesisemat keelekasutust. Projekti alguses oli kasutusel ka Eesti kirjakeele sagedussõnastiku[3], mis valdas väiksemat sõnade arvu ja seetõttu lubas kiiremat programmi arendamist.

[2] https://keeleressursid.ee/et/256-sagedusloendid
[3] https://www.cl.ut.ee/ressursid/sagedused/

2.2 Sõnaühenduste loomine

Sõna ühenduste loomiseks on kõigepealt loetud sisse eelnevalt salvestatud JSON fail, mis sisaldab sõnastiku kõigist sõnadest ja nende ühendustest. Ühenduste leidmiseks programm käib läbi kõik sõnastikus leiduvad sõnad. Kõigepealt programm teeb kindlaks, et sõna ei leidu juba olemasolevates nelja sõnalistes komplektides ja sõna seoseid on piisavalt, et luua nelik. Järgmisena algoritm käib läbi kõik seoste tüübid ja seose alla jäävad sünohulgad ja kontrollib, kas leidub kindel seos, mis on vähemalt neli sõna pikk. Kui seos on leitud siis saab ühenduse tüüp kirja pandud ja märgitud, et sobiv sõna on leitud. Protsessi lõpus on uuritud sõna kustutatud võimalike sõnade hulgast. Kui sobivat sõna ei leitud ja sõnad on otsas, siis ühenduste genereerimine lõpetab töö.

Peale sobiva sõna leidmist läheb algoritm edasi sõna nelikute genereerimise etapile. Kõigepealt käib kood läbi kõik varasemalt leitud seoste tüübid, millel oli vähemalt nelja sõnaline seos mingil sünohulgal. Seejärel saab konstrukteeritud teema järgi sobiv pealkiri, mis kirjeldab nelikus olevate sõnade vahelist seost. Peale pealkirja loomist on vaja leida kõik võimalikud neljasõnalised alamsõnade seosed sellel sõnal. Selle jaoks käib programm läbi kõik seoste teemad. Teemad millel on vähem kui neli sõnad saavad vahele jäetud kuna nendest pole võimalik luua nelikut. Kuna me leiame kõik nelja sõnalised kombinatsioonid, kus järjekord ei loe ja kordused ei ole lubatud, siis on kasutusel valem $C(n,4) = \frac{n(n-1)(n-2)(n-3)}{4!}$. Valemi pealt saame lugeda, et programm on $O(n^4)$ ajalise keerukusega.

Seetõttu on väiksemaks lõigatud sünonüümi rühmade seoste järjendid, mis sisaldavad rohkem kui 6 sõna, et vältida polünoomse kasvu kiiret tõusu, mis võimaldab nelikuid genereerida mõistliku ajaga ja säilitada nende varieeruvust.

2.2.1 Raskusastme leidmine

Mängu jaoks on teemad märgistatud raskusastme värviga, seega tuleb eeltöötluse käigus määrata igale sõnade nelikule numbriline raskusaste. Raskusastme määramiseks on mitmeid võimalusi. Kõige lihtsam variant oleks arvutada nelikus olevate sõnade keskmine pikkus ja võtta see raskusastmeks, kuna pikemad sõnad on tavaliselt keerulisemad. See meetod ei sobinud aga mängu loomisel, sest see ei olnud piisavalt täpne sõnade raskusastme määramisel.

Alternatiivina saab võrrelda sõnade sünohulkade seoseid. Sünohulgad, millel on rohkem seoseid, on tõenäoliselt tuntumad ja seetõttu ka arusaadavamad. Lõpuks valiti siiski kolmas lähenemine: kasutusel võeti sagedussõnastikku, mille abil arvutati sõna esinemissageduse

põhjal normaliseeritud väärtus. See osutus kõige tõhusamaks meetodiks, kuna sagedamini esinevad sõnad on eesti keeles üldiselt tuntumad ja inimestele paremini mõistetavad. Seega on nende vahelisi seoseid lihtsam luua.

Seega järgmine samm sõna nelikute genereerimise algoritmis on arvutada igale variatsioonile raskusastme valitud meetodi järgi. Kõigepealt on kasutusel valem $gradient = 1.0 + (3.0 - 1.0)(1 - \frac{x-min}{max-min})$ [10]. Valem põhineb *Min-Max Scaling* normaliseerimismeetodil [10], mille üldkuju on valem $X_{\text{norm}} = \frac{(X - X_{\text{min}})}{(X_{\text{max}} - X_{\text{min}})}$.

Käesolevas töös on seda valemit kohandatud. Esmalt pööratakse väärtuste kasvusuund ümber, lahutades normaliseeritud tulemus ühest. Raskusastme arvutamisel on määratud tulemuse piirideks konstantsed väärtused: minimaalne väärtus 1.0 ja maksimaalne väärtus 3.0. Need määravad, et normaliseeritud raskusaste jääb vahemikku 1 kuni 3.

Minimaalseks sageduseks kasutatakse sagedussõnastikus esinevat väikseimat väärtust (min), maksimaalseks aga väärtust 100 (max). Maksimaalseks sageduseks on valitud väärtus 100, et vähendada ebaproportsionaalset mõju, mida väga kõrge sagedusega sõnad avaldaksid raskusastme arvutusele. Kuna sagedussõnastikus on enamus sõnade sagedus alla 100, ning samal ajal on ka sõnu, mille sagedus ületab seda piiri kordades, aitavad selline piirmäär ja normaliseerimine saavutada ühtlasema jaotuse. Kõigile sõnadele, mille sagedus on suurem kui 100, määratakse raskusastmeks automaatselt minimaalne väärtus 1.0.

Ülejäänud sõnade puhul rakendatakse eespool toodud valemit. Muutuja x tähistab iga sõna sagedust, mille alusel arvutatakse vastav raskusaste. Selle tulemuseks saab igale sõnale määratud ühtlaselt kasvav raskusaste.

Peale sõnade raskusastme määramist saab lõpuks arvutada neliku raskusastme. Selle jaoks käib programm üle kõik neli sõna antud variatsioonis ja arvutab nende raskusastme keskmise. Kui mõnel sõnal ei leidunud sagedust ja seetõttu puudub ka raskusaste, siis saab selle raskusastmeks automaatselt määratud maksimaalne raskus, ehk kolm. Puuduvate sõnade raskusastmeks saab pandud just selline tase, sest enamus sõnad, mis puuduvad sagedussõnastikust olid väga spetsiifilisel teemal ja seega ka tüüpiliselt väga keerulised. Lõpuks saab keskmine raskusaste korrutatud läbi neliku teemaga kokku käiva raskusastme kordajaga (vaata lisa 1).

Kuna eesti keeles esineb palju liitsõnu, peab programm nendega samuti arvestama. Liitsõnad esinevad tüüpiliselt sagedussõnastiku põhjas ning seetõttu määratakse neile kõrge raskusaste,

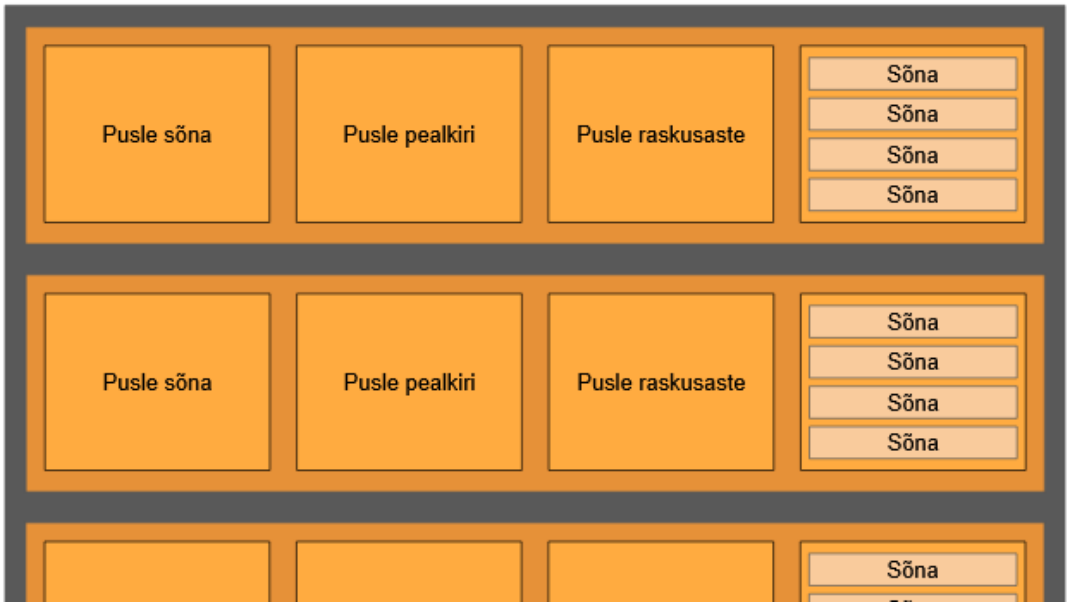
kuigi kahe liitsõna vahelise seose leidmine on tüüpiliselt lihtne. Selleks kasutatakse algoritmi, mis arvutab kahe sõna sarnasuse: kui sõnad on piisavalt sarnased, jagatakse raskusaste kahega, juhul kui seose tüübiks on sõna hüponüümid. Algoritm töötab, lõigates sõnad kõikvõimalikeks tähekombinatsioonideks ja otsides neist vähemalt kolme tähe pikkuseid ühiseid juppe.

Alternatiivina saab kasutada ka EstNLTK tööriista CompoundWordTagger, mis võimaldab tuvastada liitsõnu ja nende osi. Kuigi see annaks mitu korda täpsema tulemuse võrreldes töös kasutusel oleva meetodiga, ei sobinud see siiski kasutamiseks. Põhjuseks on see, et raskusastme arvutamine toimub väga suure hulga sõnade puhul ning EstNLTK liitsõnade tuvastamise võimalus aeglustaks protsessi liigselt võrreldes tekstitöötlusega.

2.2.2 Tulemuse salvestamine

Peale raskusastmete arvutamist saab andmed kirjutatud neliku objekti, kus leidub sõna mille järgi nelik loodi, pealkiri, raskusaste ja nelja sõnaline kombinatsioon. Peale seda kui algoritm on kõik võimalikud ühendused läbi vaadanud on vaja andmed salvestada faili, et hiljem nendest mäng luua.

Faili salvestamiseks muudetakse kõik nelikute objektid järjendiks ja lisatakse omakorda järjendisse, mis saab kirjutatud JSON faili, et hiljemalt veebilehes sisse lugeda. JSON faili salvestatud andmed omavad sama struktuuri kui nelikute objektid (vaata Joonis 2).



Joonis 2. Illustratsioon JSON failis hoitud andmete ülesehitusest.

See algoritm genereerib eelnevalt uuritud 19684 sõnast 226116 nelikut, mis võimaldab 32 aastat järjest mängida täiesti unikaalset mängu, kui iga päev kasutada nelja uut sõnade nelikut.

2.3 Definitsioonide kirjapanemine

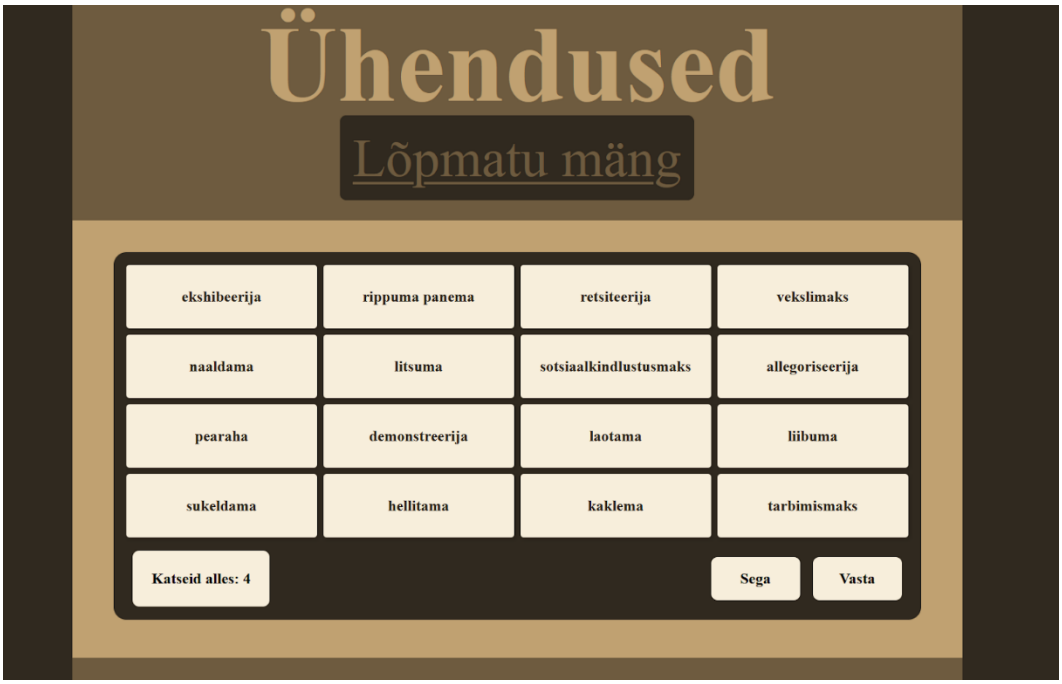
Kuna mängu üks peamine ülesanne on keeleõpe ja täpsemalt, uute sõnade õppimine, siis selle jaoks on vajalik pakkuda igal sõnal definitsiooni, mida kasutaja saab vaadata peale mängu lõpu. Selle saavutamiseks on kasutatud Eesti Wordneti, kust on käidud läbi kõik võimalikud sõnad. Need sõnad on lisatud sõnastiku, kus iga sõna nimi on kaardistatud sõna definitsioonile. See sõnastik on järgnevalt kirjutatud JSON faili, mida mäng saab hiljem sisse lugeda.

2.4 Algoritmide implementeerimine

Eeltöötluse algoritmid on implementeeritud programmeerimiskeeles Python, kasutades Jupyter Notebook faili Google Colab keskkonnas. Python valiti, kuna selle jaoks on olemas EstNLTK teek, mis võimaldab mugavalt töödelda Eesti Wordneti andmeid. Eeltöötluse programm loodi Jupyter Notebook failis, kuna see võimaldab koodi osade kaupa käivitada. Selline lähenemine on oluline, kuna sõnade töötlus on ajamahukas protsess ning osade kaupa käivitamine võimaldab vigade ilmnemisel protsessi pooleli jätta ja hiljem jätkata. Keskkonna Google Colab kasutamine võimaldas programmi aktiivselt arendada ja käivitada otse veebipõhiselt, ilma vajaduseta kohalikuks seadistamiseks. Samuti võimaldab see teistel kasutajatel programmi mugavalt rakendada.

3. Veebipõhine mäng Ühendused

Mängu mängimiseks on vajalik luua kasutajaliides, mis suudaks töödeldud informatsioonist mäng luua ja lasta kasutajal vastust pakkuda. Selle loomiseks on peamised võimalused luua kas töölauarakendus, mobiilirakendus või veebirakendus. Selle projekti juures on kasutusel programmeerimiskeeles JavaScript kirjutatud veebirakendus. Tegu on veebirakendusega, kuna erinevalt teiste variantidega, ei pea kasutaja veebirakendust alla laadima, et seda proovida. Hea ligipääsetavus mängule on tähtis, sest see lihtsustab tagasiside kogumist ja võimaldab ka huvilistel, kes pole veel kindlad, mängu kohe katsetada. Mängu veebilehe disain sarnaneb mängu Connections lehekülje disainile, kuigi proovib kasutada ka unikaalsed disaini elemente (vaata Joonis 3).



Joonis 3. Ühendused mängu veebileht.

Valminud mäng on kätte saadaval Github veebilehel https://drsirknight.github.io/Loputoo/ ja selle repositooriumi leiab lingilt https://github.com/DrSirKnight/Loputoo.

3.1 Segamise algoritm

Mitmes mängu osas on vajalik massiivis hoitud elementide asetamist suvalisse järjekorda. Selle saavutamiseks on kasutusel Fisher–Yates segamise algoritmi. Töös on kasutusel Mike Bostocki [11] loodud implementatsioon algoritmist. See implementatsioon on Bostocki sõnul O(n) ajalise keerukusega. Algoritmi töö käib järgmiselt:

1. Algoritm paneb kirja antud massiivi pikkuse kui m
2. Algoritm valib suvalise elemendi massiivist, mis asub enne m indeksil olevat elementi.
3. Algoritm vähendab numbrit m ühe võrra.
4. Algoritm vahetab m indeksil oleva elemendi suvaliselt valitud elemendiga
5. Algoritm kordab tööd sammust 2 kuni numbri m väärtus jõuab nullini.

Kuna töö jaoks oli vajalik luua täpselt sama mäng kõigil kasutajatel samal päeval, siis oli vaja algoritmi natuke muuta. Peamiseks muudatuseks on suvalisuse funktsioonile seemne andmise võimaluse lisamine. Seemnega suvalisus tähendab, et suvaliselt genereeritud number on alati sama, kui sellele antud seeme on sama. Kuna seemnega suvalisus on vajalik ainult kuupäev järgi mängu loomisel, siis nendel juhtudel kui seemneks ei ole spetsifitseeritud kindel arv, siis pannakse seemne väärtuseks suvaline number.

3.2 Mängu eeltöötlus

Veebileht loeb kõigepealt sisse eeltöötlusel loodud JSON faili. Kuna veebileht kasutab programmeerimiskeelt JavaScript, siis on võimalik meil kasutada *fetch* meetodit, et lugeda andmete faili ja siis kasutada JSON meetodid, et parsida failis olev informatsiooni kui JSON. See JSON informatsioon saab edasi pandud andmete töötlemise meetodisse.

3.2.1 Suvaliste nelikute valimine

Mängu loomise esimene etapp on leida neli suvaliselt valitud nelikut, milles ei ilmu korduvaid sõnu. Kuna mängul on kaks lehekülge, peamine lehekülg mille mäng uuendatakse kord päevas ja teine lehekülg, kus kasutaja saab alati uut mängu mängida, siis algoritm kõigepealt kontrollib, kummal leheküljel me oleme. Selle saavutamiseks on olemas nähtamatu objekt peamisel leheküljel, mida me otsime ID järgi. Kui objekti ei leidu siis teab algoritm kasutada tänast mängu. Edasi peab algoritm nüüd suvaliselt segama kogu nelikute massiivi. Selle jaoks on kasutatud peatükis 3.1. seletatud algoritmi. Sealjuures on kasutusel ka seemnetatud suvalisus, kus suvalisuse seemneks on tavaliselt pandud segamise meetodi poolt leitud suvaline arv. Kui tegu on mänguga, mis vahetub ainult kord päevas, siis on suvalisuse seemnena hoopis kasutatud tänast kuupäeva valemiga, päev 1000000 + kuu 10000 + aasta. Selle valemiga saab loodud iga päeva jaoks unikaalne number. Sellega saab kindlaks teha, et kõik mängijad saavad mängida ühel päeval sama mängu.

Järgmisena saame võtta esimesed neli nelikut segatud massiivi algusest. Siin tekib ka probleem, et nelikutes võivad olla kattuvad sõnad. Selle vältimiseks kontrollime korduvalt kõik sõnad üle ja valime korduste esinemise puhul massiivis järgmise neliku, mis sobiks.

3.2.2 Neliku raskusastme ühendamine värviga

Mängus on märgitud kõik neli nelikut erineva raskusastmega, mis on märgitud värvide järgi. Igal nelikul on määratud raskusaste, mis tähistab, kui keeruline on mängijal selle neliku sõnu üksteisega seostada. Need raskusastmed said määratud eeltöötluses numbrilisele skaalale nullist kolmeni, kus 0 tähistab kõige lihtsamat ja 3 kõige raskemat nelikut. Visuaalselt kuvatakse raskusastet mängus värvikoodide kaudu. Iga raskusaste on seotud kindla värviga sinise gradiendist, kus helesinine käib kõige lihtsama ja tumesinine kõige raskema puhul. Selle jaoks on programmeerimiskeele JavaScript *Map* objekt, millel on igale raskusastme indeksile pandud vastavusse kindel värvikood (#E3F2FD, #BBDEFB, #90CAF9, #64B5F6) (vaata Joonis 4).

Joonis 4. Raskusastmete värvid

Seetõttu on järgmine etapp mängu loomisel sorteerida nelikud eeltöötlusel saadud raskusastme järgi kasvavas järjekorras. Peale sorteerimist saame luua JavaScript *Map* objekti, kus on ühendatud neliku nimi selle raskusastme indeksiga.

3.2.3 Sõnade märgendamine ja mänguvälja loomine

Mängu jaoks on vajalik teada sõnade ja nende kuuluvuse vahelist seost. Selle teostamiseks on kasutatud jällegi JavaScript *Map* tüüpi muutujat, mille kaudu on seotud iga sõna ja sellele

sõnale vastava neliku nimi. See lahendus võimaldab hiljem kontrollida, kas mängija valitud sõnad kuuluvad samasse nelikusse.

Samal ajal kui sõnu ja neliku nimesid kaardistatakse, paneb algoritm kirja ka kõik sõnad ühte massiivi. Kuna igas nelikus on 4 sõna tuleb massiiv lõpuks 16 sõna pikk. Ennem kui sõnad on valmis mängijale kuvamiseks, tuleb need läbi segada, et vältida mustreid, mis tekivad nelikute järjest sisse lugemisest. Selle jaoks on kasutusel peatükis 3.1. seletatud algoritm.

Peale sõnade märgendamist on mänguväli valmis kuvamiseks. Selle jaoks on kirjelduskeeles HTML loodud ruudustik, mis algselt ei sisalda mingit teksti. Sõnade töötluse lõpus käib programm läbi kõik ruudud antud ruudustikus ja paneb nende tekstiks eelnevalt massiivi pandud sõnad.

3.3 Mängija poolne interaktsioon

Mängija interaktsioon veebilehega toimub peamiselt läbi järgmiste tegevuste:

1. Sõnade valimine, kus kasutaja saab valida neli sõna, mis ta arvab, et sobiksid vastusena.
2. Vastuse esitamine, mille juures kasutaja saab pakkuda, et praegu valitud neli sõna on omavahel ühendatud.
3. Definitsioonide kuvamine, mille kaudu kasutaja saab peale vastuse teada saamist vaadata, mis on ühendatud sõnade definitsioonid.
4. Sõnade segamine, millega kasutaja saab sõnade järjekorda ruudustikus muuta, et uue perspektiiviga aidata seoste leidmisega.
5. Tulemuse jagamine, kus mängija saab mängu lõpus jagada enda saavutatud tulemust mängus.

Nende interaktsioonide saavutamiseks on loodud nupud kasutades programmeerimiskeeles JavaScript olevaid sündmuse kuulajaid. Sündmuse kuulajad võimaldavad jälgida kasutaja vajutusi HTML elementide peale ja nendele reageerida koodiga.

3.3.1 Sõnade valimine

Sõnade valimine on üks mängu peamistest interaktsioonidest. Mängija saab sõna peale vajutades valida ruudustikust sõnu, mis ta arvab kuuluvad samasse gruppi. Korraga on võimalik valida kuni neli sõna, mida programm jälgib valitud sõnade massiiviga. Sõna peale vajutamisel saab toimuda kaks erinevat sündmust.

1. Kui sõna ei ole juba valitud ja kasutaja vajutab selle peale, siis sellele lisatakse valitud sõnadega kokku käiv klass, mis määrab sõna visuaalse stiili ja seejärel saab sõna kirja pandud valitud sõnade massiivi.
2. Kui sõna leidub valitud sõnade massiivis ja kasutaja vajutab selle peale, siis kustutatakse sõna sealt ja seejärel kustutatakse valitud sõna klass. Peale klassi kustutamist naaseb sõna tagasi ruudustikus asuvate sõnade vaikestiili.

3.3.2 Vastuse esitamine

Kui mängija on enda sõnade valikud teinud, saab ta vajutada "Vasta" nuppu. Ka sellel nupul leidub sündmuse kuulaja, mis kutsub välja koodi, et kontrollida praegu valitud sõnu ja seejärel uuendada mängu seisu. Selle jaoks käib kood läbi järgmised sammud:

1. Kontrollib, kas kasutaja on valinud neli sõna ja katkestab tegevuse, kui mängija pole valinud küllalt sõnu.
2. Paneb kirja kõik valitud sõnade tekstilise sisu.
3. Paneb kirja kõik sõnade vahelised ühendused, mis kuuluvad praegu valitud sõnade hulka kasutades valitud sõnade tekstilist sisu.
4. Võrdleb kõiki ühendusi ja leiab selle, mille alla jääb kõige rohkem valitud sõnu. Paneb kirja leitud ühenduse ja mitu sõna sellega kokku käis.
5. Kontrollib, kas kõige rohkem valitud sõnadega ühendusel leidus neli sõnu ja vähendab arvamise võimaluste arvu, kui ei leidu. Konstrueerib ja näitab kasutajale vastust, kui ühendus leiti.
6. Kui kasutaja arvas õigesti ainult kolm sõna, siis mäng näitab teksti "Kolm õiget!", et edasist arvamist kergemaks teha.
7. Peale protsessi lõpu märgib mäng kõik sõnad mitte valituks, uuendab teksti, mis näitab kui palju katseid mängijal alles on ja lõpuks kontrollib, kas mäng on läbi või mitte.

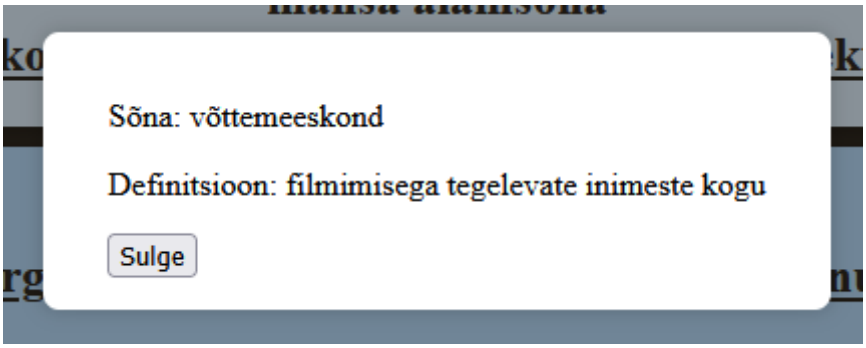
Vastuse konstrueerimisel eemaldatakse kõigepealt õigesti valitud sõnad ruudustikust. Seejärel valib mäng vastuse elemendid nende indeksi järgi ja konstrukteerib vastuse järgmiste etappidega.

1. Vastuse pealkirjaks pannakse valitud ühenduse pealkiri.
2. Vastuse tekstiks kirjutatakse sõnad HTML elementidena, mis on vajutatavad ja lisatakse nende vahele komad.
3. Vastus tehakse nähtavaks ja selle taustavärv määratakse neliku raskusastme indeksi järgi. (Vaata peatükki 3.2.2)

Peale vastuse konstrueerimist saab kirja pandud vastus, suurendatud vastuste indeks ühe võrra ja uuendatud mängu ruudustik, et selles oleks üks rida vähem.

3.3.3 Definitsioonide kuvamine

Peale ühenduse nägemist on võimalik kasutajal vajutada ühenduse alla kuuluvate sõnade peale, et näha nende definitsioone. Peale sõna peale vajutamist avaneb kasutajal hüpikaken, millel on kuvatud sõna ja selle definitsioon. Selle jaoks kontrollib rakendus peale iga elemendi peale vajutamist, kas see on vajutatav sõna ja seejärel kuvab programmeerimiskeele JavaScript *Dialog* elementi. *Dialog* element võimaldab lehekülje kohal informatsiooni kuvada [12]. *Dialog* elemendi sees on kirjutatud sõna ja selle definitsioon (vaata Joonis 5). Definitsioon on võetud eeltöötluses loodud definitsioonide JSON failist.



Joonis 5 . Sõna definitsiooni näitamine kasutajale

Definitsioonide kuvamine on oluline komponent tööst, kuna see toetab mängija sõnavara arengut, andes võimaluse õppida uute sõnade kohta, mis mängu jooksul on ette tulnud.

3.3.4 Sõnade segamine

Mõnikord võib mängides tekkida probleem, et kasutaja ei märka ühendusi sõnade vahel, selle jaoks on kasulik saada uut perspektiivi. Selle saavutamiseks on mängus “Sega” nupp, mis laseb kasutajal sõnade järjekorda suvaliselt muuta. Selle jaoks lisame massiivi kõik mängu ruudustikus asuvad sõnad ja kasutame peatükis 3.1. kirjeldatud segamise algoritmi, et sõnad suvalisse järjekorda panna. Peale seda asendame kõik ruudustikus olevate elementide teksti iga massiivis oleva elemendi tekstiga ja kustutame praegu valitud sõnad mälust. Sellega on saavutatud suvaliselt segatud elemendid mängu ruudustikus.

3.3.5 Tulemuse jagamine

Peale mängu lõppu muutub nähtavaks mängu jagamise nupp, mis lubab kasutajal jagada enda tulemust, näidates mitu korda ta arvas õigest ja mitu korda ta arvas kokku. Samuti on nähtaval,

kas mängija mängis iga päev uuenduvat mängu versiooni või lõpmatut versiooni. See on saavutatud mängija lõikelauale teksti kirjutamisega. Tekst esineb kujul:

- Igapäevase mängu puhul: 🟦⬛⬜ Ma leidsin 4 ühendust 6 arvamisega tänases mängus! 🟦⬛⬜
- Lõpmatu mängu puhul: 🟦⬛⬜ Ma leidsin 1 ühendust 5 arvamisega suvalises mängus! 🟦⬛⬜

Jagamisel väljastatud sõnum sai ümbritsetud Eesti lippu värviliste emotikonidega, et näidata mängu fookust eesti keelele.

3.4 Mängu lõpp

Mäng lõpeb kahel erineval juhul: kas mängija on leidnud kõik neli ühendust või on tema katsete arv otsa saanud. Mängu lõpu tingimusi kontrollitakse iga kord, kui kasutaja esitab uue vastuse. Kui üks kahest tingimusest on täidetud, kutsub programm välja mängu lõpetamise funktsiooni.

Mängu lõpetamisel kõigepealt peidetakse kõik mänguga seotud elemendid. Nende elementide hulka kuuluvad sõnade ruudustik, käikude arv, segamise nupp ja vastuse esitamise nupp. Seejärel tehakse nähtavaks mängu lõppu informatsiooni ja jagamise nuppu. Seejärel käib programm läbi kõik võimalikud ühendused. Kui ühendus on juba lahendatud siis me ignoreerime seda, aga kui ühendus on veel lahendamata siis me kustutame alles olevad sõnad, mis on sellega seotud. Seejärel kasutamine 3.2.2. peatükkis kirjeldatud vastuse konstrukteerimise meetodit, et luua vastus. Lõpuks kuvame, mitu ühendust kasutaja suutis leida mängu lõppu informatsiooni all.

4. Tagasiside

Mängu tagasiside kogumine toimus vabas vormis, kus mängijad said mängu proovida ja siis, kas suuliselt või teksti kaudu jagada mängu kohta arvamusi. Mängijatelt samuti küsiti täpsustavaid küsimusi, mis nad arvasid mängust ja kas mõni probleem segas mängimist. Mängu testimise käigus koguti tagasisidet kokku viieteistkümnelt inimeselt, kus kõik olid autori tuttavad või sõbrad. Testijatest küsiti kuuelt tagasisidet näost näkku. Ülejäänute testijatega suheldi üle interneti. Testijate hulgas olid nii kogenud sõnamängude mängijad kui ka need, kes selliseid mänge tihti ei mängi. Mitmekesine testijate grupp võimaldas saada väärtuslikku tagasisidet nii mängumehaanika kui ka kasutajakogemuse kohta. Tagasiside oli üldiselt positiivne, kuigi selle jooksul leiti ka mitmeid probleeme, mida pidi parandama.

Üheks esimeseks probleemiks, mis testimisel esile kerkis, oli mängu sobimatus väiksemate ekraanide jaoks. Nimelt ei olnud võimalik esialgset mängu mugavalt mängida ekraanidel, mille horisontaalne laius oli alla 1300 piksli. Selle probleemi lahendamiseks loodi eraldi stiilid, mis rakenduvad juhul, kui kasutaja seadme ekraani laius jääb alla 1300 piksli. See parandus oli hädavajalik, kuna see võimaldas mängu mängida ka mobiilil.

Tagasiside käigus avastati ka üks programmiviga. Mitmete test mängude läbiviimisel ilmnes, et mäng võis genereerida olukordi, kus mõnes mängus esinesid korduvad sõnad ehk sama sõna kuulus mitmesse erinevasse ühendusse. See tegi lahenduse leidmise võimatuks, sest algne kood eeldas, et kõik sõnu esineb ainult üks kord. Peale koodis sõnade korduse kontrolli implementeerimist testimine jätkus ja viga enam ei ilmunud.

Veel üks probleemi koht, mis ilmnes testimisel oli see, et alguses definitsiooni näitamiseks kasutatud programmeerimiskeele JavaScript *alert* funktsiooni oli võimalik kasutaja poolt ära peita. Peale selle peitmist oli väga keeruline see tagasi tuua ja kasutaja enam definitsioone ei näinud. Seetõttu oli kasutusele võetud spetsiaalselt disainitud JavaScript Dialog, mida kasutaja ei saanud enam ära peita.

Tagasisidet kogudes leiti ka, et algselt valitud värvid raskusastme jaoks tekitas segadust. Algsed raskusastme värvid algasid sinisest ja muutusid punaseks. Sealjuures aga tekkis probleem, et kasutaja oli vastuse leidnud ja nägi selle juures punast tausta, mis tekitab mulje nagu midagi oli valesti. Selle lahendamiseks prooviti mitmeid erinevaid värve ja küsiti testijatelt nende kohta tagasisidet. Lõplikult valiti raskusastme märkimiseks tumenevad sinise toonid.

Kuigi kogenud sõnamängijate tagasiside oli hädavajalik, osutus ka uute mängijate panus väga väärtuslikuks. Nende tähelepanekute põhjal ilmnes, et mängu reeglid ei olnud kohe piisavalt arusaadavad. Selle probleemi lahendamiseks lisati mängu leheküljele lühike tekst, mis tutvustab mängu reegleid.

Kokkuvõttes osutus tagasiside kogumine ja mängu testimine mitmete kasutajatega äärmiselt oluliseks. Ilma selleta oleks lõppversiooni jäänud mitte ainult visuaalse disaini puudused, vaid ka olulised vead koodis.

4.1 Edasiarendamise võimalused

Edasiste arenduste keskmes võiks olla raskusastme määramise algoritmi täpsustamine, et see suudaks paremini arvestada liitsõnadega, mis võivad olla kasutajale siiski lihtsad. Samuti võiks parandada nelikute laadimise loogikat. Praegu laeb programm sisse terve JSON faili jagu nelikuid, iga kord kui uus mäng saab loodud.

Veel üks edasiarendusvõimalus oleks ka mängu oleku salvestamine, võimaldades tuvastada, kas kasutaja on tänase neliku juba lahendanud ning mitu võitu järjest on saavutatud. Seda infot saaks kasutada ka tulemuste jagamisel.

4.1.1 Tehisintellekti kasutus

Töö loomise käigus uuriti ka tehisintellekti kasutust sõnade genereerimisel. Testiti erinevaid keele mudeleid nagu DeepSeek-V3 ja GPT-4o. Protsessi käigus selgus, et kuigi suured mudelid suutsid antud teemade kohta leida sõnade vahelisi seoseid, siis see protsess oli väga aja ja ressursimahukas. Seetõttu prooviti väiksemaid mudeleid ja mudelite kvantifitseeritud vorme, mis lasi sõnu kiiremini ja efektiivsemalt genereerida, aga sellega nõrgenes mudelite nii eesti keele arusaam kui ka sõnade vaheliste seoste leidmise oskus. Väiksemate mudelite juures leidus kõige efektiivsemaks eestikeelne keelemudel Llammas, mis suutis ülesandest aru saada, kuid sellel tekis raskusi vastuse väljastamisel kasutatavas formaadis. Samuti ei olnud mudeli loodud seosed küllalt keerukad, et asendada Eesti Wordneti abil loodud seoseid. Kõige tähtsam asi, mis silma paistis oli see, et mudelid ei olnud täiesti usaldusväärsed ja võisid vahepeal pakkuda sõnu, mis ei ole omavahel seotud. See oleks läinud projekti nõuete vastu, kuna töö keskendub keeleõppele ja selle juures on tähtis täpsus.

Seetõttu jäi lõpp rakenduses kasutusele ainult Eesti Wordnet, mis tagas usaldusväärsema tulemuse. Kuid tehisintellekt ei saanud implementeeritud siis see on ikkagi väärtuslik uurimis

punkt edasiarendamises, et lisada sõnade ühendusi, mis on seotud keerulisemate teemadega, nagu Eesti kultuur.

Kokkuvõte

Selle bakalaureusetöö käigus tutvustati keelemänge, keelemängude kasulikkust keeleõppes, Eesti Wordneti andmete töötlust, sõnaühendustele raskuse arvutamist ja veebirakenduse loomist keeleõppe mängu jaoks. Mängu eesmärgiks oli õpetada uusi sõnu, sõnade vahelisi seoseid ja sõnade definitsioone läbi meelelahutuse. See saavutati kasutades eestikeelseid keeleressursse nagu Wordnet ja sagedussõnastikud.

Mäng Ühendused võttis suurelt osalt inspiratsiooni mängust Connections. Mängu lõppeesmärk on leida kuueteist sõna seast neli sõnagruppi. Sellise mängu saavutamiseks implementeeriti veebilehel sõnagruppidest mängude loomine, sõnade valimine, vastuse esitamine ja sõnade järjekorra segamine. Samuti lisati veebilehele võimalus vaadata sõnade definitsioone. Andmete töötlus tehti eraldi Jupyter Notebook failis, kus eelnevalt leiti kõik sõnade ühendused ja nende raskusastmed.

Mäng saavutab haridusliku eesmärgi näidates kasutajale haruldasemaid eestikeelseid sõnu koos nende definitsioonidega ja osaliselt sarnaste sõnadega. Mängu tagasiside oli üldiselt positiivne ja kõik vead, mis tagasiside kogumise ajal leiti said parandatud.

Viidatud kirjandus

[1] Trasberg T. Rahvaloenduse tulemused on avaldatud. https://www.stat.ee/et/uudised/rahvaloenduse-tulemused-avaldatud (20.04.2025)

[2] Keeleseadus. https://www.riigiteataja.ee/akt/835593 (20.04.2025)

[3] Klimova B, Kacet J. Efficacy of Computer Games on Language Learning. Turkish Online Journal of Educational Technology - TOJET 2017; 16: 19–26. https://eric.ed.gov/?id=EJ1160637 (20.04.2025)

[4] Gupta M., Kumar R., Kumar U., & Singh A. (2023). Use of Digital Word Games as a Tool for Improving Vocabulary Skills. 2023 International Conference on Advancement in Computation & Computer Technologies (InCACCT), Advancement in Computation & Computer Technologies

[5] Only Connect. https://www.imdb.com/title/tt1294009/ (28.04.2025)

[6] Abdul G. What connects Only Connect and New York Times? Game's similarity puzzles host. https://www.theguardian.com/media/2023/jun/15/connections-new-york-times-puzzle-only-connect-victoria-coren-mitchell (28.04.2025)

[7] Samadarshi P., Mustafa M., Kulkarni A., Rothkopf R., Chakrabarty T., & Muresan S. (2024). Connecting the Dots: Evaluating Abstract Reasoning Capabilities of LLMs Using the New York Times Connections Word Game. https://arxiv.org/abs/2406.11012 (28.04.2025)

[8] Tornoe R. How the New York Times is making connections with puzzles and games. Editor and Publisher, 2023. https://www.editorandpublisher.com/stories/new-york-times-makes-a-connection-with-puzzles-and-games,246051 (07.12.2024)

[9] Orav H., Vare K., Parm S., Eesmaa L., Taremaa P., Reile M., Alekand K., Jaska I., Türk H., Aedma E., Lohk A. Eesti Wordnet. https://keeleressursid.ee/et/265-eesti-wordnet (08.12.2024)

[10] Wei D. Demystifying Data Normalization in Machine Learning. https://medium.com/@weidagang/demystifying-machine-learning-normalization-0cdb8b281234 (15.05.2025)

[11] Bostock M. Fisher–Yates Shuffle. https://bost.ocks.org/mike/shuffle/ (18.04.2025)

[12] <dialog>: The Dialog element. https://developer.mozilla.org/en-US/docs/Web/HTML/Reference/Elements/dialog (12.05.2025)

Lisad

1. Raskusastme kordajad

sünonüümid = 1.0

osad = 1.5

alam sõnad = 2

alam sõnad ja leitud liitsõna = 0.5

sõnad, millel on üks täht lõppu juurde lisatud = 1.7

Litsents

Lihtlitsents lõputöö reprodutseerimiseks ja üldsusele kättesaadavaks tegemiseks

Mina, Joosep Hubel,

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) minu loodud teose

   Eestikeelne sõnamäng Ühendused,

   mille juhendaja(d) on Sven Aller,

   reprodutseerimiseks eesmärgiga seda säilitada, sealhulgas lisada Tartu Ülikooli digitaalarhiivi kuni autoriõiguse kehtivuse lõppemiseni;

2. annan Tartu Ülikoolile loa teha punktis 1 nimetatud teos üldsusele kättesaadavaks Tartu Ülikooli veebikeskkonna, sealhulgas digitaalarhiivi kaudu Creative Comptoni litsentsiga CC BY NC ND 4.0, 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.

Joosep Hubel

15.05.2025