

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Maarja Helena Elisabeth Hoop
194515IADB

TalTechi algoritmilise mõtlemise veebimängu serverrakenduse arendus

Bakalaureusetöö

Juhendaja: Meelis Antoi

Magistrikraad

Kaasjuhendaja: Kerman Saapar

Bakalaureusekraad

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: Maarja Helena Elisabeth Hoop

25.04.2022

Annotatsioon

Käesolevas töös lahendatakse probleemi õpilase algoritmilise taseme hindamine mängu abil. Töö eesmärk on luua algoritmilise mõtlemise mängu serverrakendus, mis on mõeldud Tallinna Tehnikaülikooli IT Kolledžile töötubade läbiviimiseks. Töö tulemusel valmib kasutusvalmis serverrakendus. Serverrakendusel ja vastaval klientrakendusel on võimalik mängida erinevaid hariduslikke mängu. Mängu mängimisel saavad kasutajad hinnata oma algoritmilise mõtlemise taset ning seda arendada.

Töö käigus uuritakse algoritmilist mõtlemist ja pakutakse uuritu põhjal välja lahendus. Planeeritava lahenduse analüüsi käigus kavandatakse rakenduse nõuded, disainitakse andmebaas ja klient—server vaheline suhtlus. Arendusprotsessi käigus järgitakse kaasaegsed tehnoloogiad. Töös kirjeldatakse rakenduse erinevate osade töötamist ja platvormil töötavat näidismängu. Valmivale rakendusele on võimalik tulevikus funktsionaalsust juurde lisada, Serverrakendust saab kasutada erinevatel platvormidel töötavate klientrakendustega.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 33 leheküljel, 6 peatükki, 10 joonist, 7 tabelit.

Abstract

Development of the Backend of TalTech's Computational Thinking Web Game

This paper analyses the subject on assessing a student's computational thinking level through a game. Tallinn University of Technology's IT College would like to use such game to conduct computational thinking workshops to introduce IT specialties to students. The problem was highlighted by the covid-19 pandemic as the lack of adequate solutions became evident. The need for a workshop that could be used in both contact and distance forms was discovered.

The aim of current thesis is to create a game that could be used in workshops that focuses on identifying and developing algorithmic thinking skills. The application is developed as a collaboration between two students. The scope of this thesis is creating a backend application for the game. Modern technologies will be used to create a comfortable and intuitive user experience.

This thesis studies computational thinking and proposes a solution for the application. During the analysis of the solution the requirements of the application are planned and the database and the communication between the client-server is designed. The paper specifies how the different parts of the application work. An example game "Programming of a Space Robot" is also described. This illustrates how the application runs a specific game. It is possible to add functionality to the completed application in the future. The server application can be used with client applications running on different platforms.

The thesis is in Estonian and contains 33 pages of text, 6 chapters, 10 figures, 7 tables.

Lühendite ja mõistete sõnastik

API	<i>Application Programming Interface</i> , rakendustarkvara liides
BLL	<i>Business Logic Layer</i> , äriloogika kiht
DAL	<i>Data Access Layer</i> , andmetele ligipääsu kiht
DevOps	Tarkvaraarenduse kultuur, mis ühendab tarkvaraarenduse ja tarkvaraoperatsioonid
Git	Versioonihaldustarkvara
Id	Identifikaator
IT	Infotehnoloogia
JPA	<i>Java Persistence API</i> , Java ORM spetsifikatsioon
JSON	<i>JavaScript Object Notation</i> , andmevahetusvorming
Klient-	Kliendipoolne, kasutajat või kasutavat süsteemi tagaosaga liidestav
MVC	<i>Model-View-Controller</i> , tarkvara arhidektuurimuster, mis jagab tarkvaraarenduse kolmeks seotud osaks: mudel, vaade, kontrolleri
npm	<i>node package manager</i> , JavaScripti paketi haldur
ORM	<i>Object-relational mapping</i> , objekt-relatsioonvastendus
REST	<i>Representational state transfer</i> , tarkvaarahidektuuri laad
Server-	Kasutajale nähtamatu, töötlev, talletav, käitlev jne põhiosa
spa	<i>single page application</i> , üheleherakendus
SQL	<i>Structured Query Language</i> , päringukeel andmebaasiga suhtlemiseks
TalTech	Tallinna Tehnikaülikool
TIOBE	<i>The importance of being earnest</i> , indeks, mis hindab keele populaarsust otsingumootori aktiivsuse põhjal
URL	<i>uniform resource locators</i> , internetiaadress
Veebilehitseja	Programm interneti kaudu saadud failide tõlgendamiseks ja esituseks

Sisukord

1 Sissejuhatus	10
2 Ülesande püstitus	11
2.1 Metoodika	11
2.2 Algoritmilise mõtlemise arendamine läbi mängu.....	12
2.3 Töö kitsendused	12
2.4 Pakutav lahendus	13
2.5 Olemasolevad lahendused	14
3 Rakenduse analüüs	15
3.1 Funktsionaalsed nõuded	15
3.2 Mittefunktsionaalsed nõuded.....	16
3.3 Tehnoloogiate valik	16
3.3.1 Teenusepoolse programmeerimiskeele valik.....	17
3.3.2 Arvutitarkvara raamistiku valik.....	19
3.3.3 Arenduskeskkonna valik	20
3.3.4 Andmebaasi valik	21
3.4 Klientrakendus.....	24
3.5 Suhtlus klient- ja serverrakenduste vahel	24
3.6 Andmebaasi disain.....	28
3.7 Analüüsi kokkuvõtte.....	31
4 Rakenduse arendus	32
4.1 Projekti loomine	32
4.2 Andmebaasiga ühendamise	34
4.3 Rakenduse osad	35
4.3.1 Andmete edastamise osad.....	35
4.3.2 Mängu loogika.....	37
4.4 Näidismäng “Kosmoseroboti programmeerimine”	38
4.5 Klientrakenduse arendus.....	40
5 Hinnang valminud rakendusele	41
5.1 Valminud veebirakenduse nõuetele vastavus	41

5.2 Edasiarendus võimalused	42
6 Kokkuvõte	43
Kasutatud kirjandus	44
Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks	49
Lisa 2 – Funktsionaalsed nõuded.....	50
Lisa 3 – Infosüsteemi skeem	53
Lisa 4 – Kasutajakogemuse disaini joonised.....	54
Lisa 5 – Andmebaasi täielik olemi-suhte diagramm	57
Lisa 6 – Mängu kosmoseroboti programmeerimine vaated	58
Lisa 7 – Link rakenduse versioonihaldusele.....	60
Lisa 8 – Link rakenduse veebilehele	61
Lisa 9 – tALGOtech kasutusjuhend töötoa läbiviijale.....	62

Jooniste loetelu

Joonis 1. Kõikide mängude edastamise JSON.	26
Joonis 2. Mängu info edastamise JSON.	27
Joonis 3. Kasutatud kastide edastamise JSON.	27
Joonis 4. Mängu vastuse edastamise JSON.	28
Joonis 5. Andmebaasi lihtsustatud olemi-suhte diagramm.	29
Joonis 6. Spring initializr kasutajaliides.	33
Joonis 7. Docker-compose andmebaasi loomiseks.	34
Joonis 8. Application.properties faili sisu.	34
Joonis 9. Koodinäide Function klassi mudel.	36
Joonis 10. Koodinäide Function klassi repositoorium.	37

Tabelite loetelu

Tabel 1. Programmeerimiskeelte võrdlus.....	18
Tabel 2. Java Raamistike võrdlus.....	19
Tabel 3. Andmebaaside võrdlus.....	22
Tabel 4. Kasutatavad API päringud.....	25
Tabel 5. Kastide tüübid.....	30
Tabel 6. Lüngatäidete tüübid.....	31
Tabel 7. Kosmoseroboti programmeerimise mängu tasemed.....	39

1 Sissejuhatus

Käesoleva töö raames lahendatakse mängu abil õpilase algoritmilise taseme hindamise probleemi. Probleemi tõstis esile koroonapandeemia ja olemasolevate lahenduste vähesus ja sobimatus. Selle jooksul oli Tallinna Tehnikaülikoolil IT kolledžil väga raske viia läbi töötubasid, mis aitaksid tutvustada IT erialasid võimalikult paljudele õpilastele. Avastati vajadus universaalse töötoa järele, mida saaks kasutada nii kontakt- kui ka distantsvormis. Töötoas keskendutakse algoritmilise mõtlemise oskuste tuvastamisele ja arendamisele. Algoritmiline mõtlemine on üks programmeerimise alustaladest. Töötoa läbimisel saab kasutaja teada, kas tal on eeldusi IT erialade õppimiseks.

Töö eesmärk on luua algoritmilise mõtlemise mäng, mis on mõeldud Tallinna Tehnikaülikooli IT Kolledžile töötubade läbiviimiseks. Töö käigus valmib kasutusvalmis rakendus, millega on võimalik mängida erineva sisuga hariduslikke mänge. Mängu mängimisel saavad kasutajad hinnata ning arendada oma algoritmilise mõtlemise taset. Rakendus arendatakse kahe tudengi koostööna. Käesoleva töö raames valmib serverrakendus. Klientrakendus valmib Triinu Malvi töö „Klientrakenduse loomine algoritmilist mõtlemist arendavale TalTechi veebimängule tALGOtech” raames.

Valmiv rakendus loob kasutajale mugava ja intuitiivse kasutajakogemuse. Selle tagamiseks kasutatakse rakenduse arendusel kaasaegseid tehnoloogiaid. Töö raames kasutatakse peamiselt internetiallikaid, sest need on antud teema puhul kõige kaasaegsemad ja asjakohasemad. Töö on jaotatud neljaks suuremaks peatükiks. Ülesandepüstituse peatükis uuritakse probleemi ja pakutakse sellele lahendus. Rakenduse analüüsi peatükis tuuakse välja rakenduse nõuded, valitakse tehnoloogiaid, kirjeldatakse klient- ja serverrakenduse vahelist suhtlust ning kirjeldatakse andmebaasi disaini. Rakenduse arenduse peatükis kirjeldatakse arendusprotsessi, kirjeldatakse ära erinevate rakenduse osade töötamis põhimõtted ja tuuakse välja näidismäng. Peatükis „Hinnang valminud rakendusele“ analüüsitakse valminud rakendust ja tuuakse välja edasiarendus võimalused.

2 Ülesande püstitus

Algoritmiline mõtlemine ei ole ainult arvutiteaduste valdkonna alustala, vaid üldine probleemide lahendamise viis. Kristin Parve kirjeldab oma artiklis algoritmilist mõtlemist ehk raalmõtlemist inglise teadlaste Cynthia Selby ja John Woollardi abiga, kelle sõnul koosneb raalmõtlemine viiest osaoskusest: „a) ülesandest olulise info eristamine, b) ülesande jagamine väiksemateks osategevusteks, c) esialgse lahenduskäigu ehk algoritmi kavandamine, d) tulemuste hindamine ja tõlgendamine ning e) lahenduskäigu üldistamine muude sarnaste ülesannete lahendamiseks.” [1]

Algoritmilise mõtlemise abil on võimalik lahendada paljusid kompleksseid probleeme ja selle tõttu on algoritmiline mõtlemine tarkvaraarenduse üks olulisematest aspektidest. Kuigi programmeerimise õppimiseks pole üldjuhul vaja eelnevaid teadmisi, teeb heal tasemel algoritmilise mõtlemise oskus õpingud lihtsamaks. Hoolimata IT-maailma kasvavast populaarsusest, on palju inimesi, kes pole programmeerimisega varem kokku puutunud. Nende inimeste algoritmilise mõtlemise oskust on efektiivne kontrollida mängu abil. Nii ei tõrju neid programmeerimise esialgu keerulisena näiv tehniline pool eemale edasisest huvist infotehnoloogia vastu.

2.1 Metoodika

Käesoleva töö metoodika peatükk kirjeldab serverrakenduse arendamise metoodikat. Paralleelselt toimub ka klientrakenduse arendus, mille metoodikat saab lugeda Triinu Malvi tööst „Klientrakenduse loomine algoritmilist mõtlemist arendavale TalTechi veebimängule tALGOtech“..

Probleemi lahendamiseks uuritakse esmalt algoritmilist mõtlemist ja olemasolevaid lahendusi. Nende põhjal pakutakse välja lahendus ja pannakse paika lahenduse skoop.

Seejärel analüüsitakse planeeritavat rakendust. Selle jaoks kaardistatakse rakenduse funktsionaalsed ja mittefunktsionaalsed nõuded. Nõuete piltlikustamiseks luuakse planeeritava infosüsteemi skeem. Rakenduse tehnoloogia valikul tuuakse välja erinevad

tehnoloogiad, mis sobiksid rakenduse arendamiseks, võrreldakse neid omavahel ja valitakse välja kõige sobivam. Analüüsi käigus planeeritakse ka klient- ja serverrakenduse vaheline suhtlus ja kirjeldatakse seda. Nõuete ja infosüsteemi skeemi põhjal disainitakse sobiv andmebaas.

Analüüsile järgneb rakenduse arendus. Arenduse käigus kirjeldatakse arendusprotsessi, rakenduste eriosade töötamist ja näidismängu. Töö lõpuosas antakse hinnang valminud rakendusele ja pakutakse välja võimalused edasiarenduseks.

2.2 Algoritmilise mõtlemise arendamine läbi mängu

Üks parimatest võimalustest algoritmilise mõtlemise õpetamiseks ja õppimiseks on selle asetamine teise teema konteksti. Näiteks saab algoritmilist mõtlemist arendada mõistatuste ja tekstülesannete abil, mis nõuavad õppijatelt nende algoritmilise mõtlemise oluliste aspektidega tegelemist. Lisaks on algoritmilist mõtlemist võimalik arendada ka ilma matemaatika või infotehnoloogiliste vahenditeta. Algoritmilise mõtlemise jaoks vajalikke aspekte leiab näiteks ka keeleõppes ja muusikateoorias. [2]

Efektive variant algoritmilise mõtlemise arendamiseks on mängu kasutamine. Seeläbi muutub õppimisprotsess lõbusaks ning see motiveerib mängijaid ja aitab neil olla tähelepanelik ja keskendunud. Hea mäng võib mängijale huvi pakkuda mitmeid tunde, lisaks võivad õpilased õppemängu vastu huvi tunda ka väljaspool töötuba. Samuti on tõendeid selle kohta, et mängud võimaldavad õpilastel paremini keskenduda ja paremini õppida. Uuringutes on leitud, et koolitunni mänguks muutmine parandab oluliselt laste õpitulemusi. [3]

2.3 Töö kitsendused

Töö on tehtud koostöös Tallinna Tehnikaülikooli IT-teaduskonnaga, mille eesmärk on tutvustada IT-erialasid potentsiaalsetele tulevastele tudengitele. Käesoleva töö käigus valmiv lahendus on eelkõige suunatud õpilastele, kes on hiljuti alustanud õpinguid gümnaasiumiastmes. Nende õpilaste näol on tegemist inimestega, kes peavad varsti mõtlema erialasele spetsialiseerumisele. Samuti on nad sobivas vanuses, et jõuda enne erialavalikut iseseisvalt infotehnoloogilisi oskusi põhjalikumalt uurida ja seeläbi oma valikus veenduda.

2.4 Pakutav lahendus

Lahenduseks probleemile pakuti välja interaktiivne veebimäng, mille koostamisel võetakse arvesse osaoskusi ning lahendataks järgnevad kaks küsimust: kuidas mängu abil aru saada, kui heal tasemel on õpilase algoritmiline mõtlemine ning kuidas arendada erinevatel tasemetel olevate õpilaste algoritmilist mõtlemist? Nende probleemide lahendamiseks on loodavas süsteemis dünaamiline tasemete süsteem, mis korrigeerib ülesande raskust vastavalt mängija tasemele. Seeläbi saab õpilane teada oma algoritmilise oskuse taseme ning tal on võimalik harjutades progresseeruda kõrgematele tasemetele. Ühe mängu jooksul peab mängija lahendama üht ülesannet. Ülesande kirjeldus on mängijale ette antud pikema kirjeldusena. Mängija peab ülesande teksti analüüsima ja vajaliku info välja otsima. Ülesande täitmiseks on mängijal ette antud sobiv arv tegevuste kaste ja kastide täitmiseks lüngatäited. Mängija peab ülesande kirjelduse järgi tegevused väiksemateks osadeks jaotama ja tegevuste kastide ja lüngatäidetega tegevused kirja panema. Mängija saab oma lahendust teatud arv kordi proovida ja vajadusel muuta. Mängu läbimisel on mängijal võimalik raskemate tasemete juures omandatuid teadmisi rakendada.

Loodav mäng peab olema ka esinduslik, et see sobiks kasutamiseks akadeemilises asutuses. Mängu on võimalik saata läbi veebi ka ülikoolist välja ja mängu mängimiseks ei ole vaja eraldi juhendajat. Nii on tulevasi õpilasi võimalik leida ka distantsilt.

Probleemi ja pakutava lahenduse keerukus on suur, seejuures on oluline ka rakenduse suur praktiline väärtus Tallinna Tehnikaülikooli IT kolledži jaoks. Võimalikult terviklikuks lahenduseks arendati rakendus kahe tudengi koostööna. Selgeks piiritlemiseks koosneb rakendus kahest osast: serverrakendus ja klientrakendus. Klientrakendus valmib Triinu Malvi lõputöö „Klientrakenduse loomine algoritmilist mõtlemist arendavale TalTechi veebimängule tALGOtech” raames, mille skooopi jääb rakenduse disain ja mängu tagasiside mängijale animatsiooni näol. Käesoleva töö raames valmib mängule serverrakendus, mis hoolitseb mängu info salvestamise ja ärioloogika eest. Serverrakenduse planeerimise käigus arvestati sellega, et loodav rakendus oleks võimalikult universaalne. Serverrakendusele on võimalik arendada erinevaid kasutajaliideseid, et mängu oleks tulevikus võimalik mängida erinevatel platvormidel. Käesolevas töös kasutatakse ainult üht mängu, kuid serverrakendus suudab salvestada ja käidelda mitmeid erinevaid mängu erinevate ülesannetega.

Lõputöö eesmärk on Tallinna Tehnikaülikooli IT kolledži vahendite täiendamine akadeemilise mänguga. Töö eesmärk ei ole saada materiaalet tulu. Töö materjalid ja rakenduse lähtekood on avalikuks kasutamiseks kooli inimestele, lootuses, et tööd täiendatakse ja arendatakse vastavalt Tallinna Tehnikaülikooli IT kolledži vajadustele.

2.5 Olemasolevad lahendused

Algoritmilise mõtlemise baasil mängud on IT maailmas väga aktuaalsed, eriti viimastel aastatel on hakatud teemat nii Eestis kui ka mujal maailmas sügavamalt uurima. Olemas on erinevaid võimalusi, mille abil saab algoritmilist mõtlemist kontrollida. Kõige lähedasem lahendus käesolevale lõputööle on Tartu Ülikooli uurijate poolt loodud mäng AutoThinking. Viimases on hästi lahendatud mängu kohanemine erinevate mängijate tasemetele. Sellest hoolimata on mängu juures puudused, näiteks pole mängus arvestatud algoritmilise mõtlemises juures olulise ülesande väiksemateks tükkiideks jagamisega. [4] Teine sarnane võimalus algoritmilist mõtlemist õpetada on graafiline programmeerimiskeel Scratch. Ka sellel on omad puudused, nimelt on Scratchi süsteem liiga keeruline, et õpilased seda intuiivselt kasutada saaks, ja programmeerimiskeele puhul on mängu ülesanne ja mäng ise lahus. [5] Sarnaseid probleeme lahendavad ka mängud Hopscotch ja The Pack. Hopscotchi abil on võimalik õppida mobiilirakenduste arendamist. Sellel puudused sarnanevad Scratchi omadega, kõige suurem probleem on kõrge õppimiskeerukus. [6] The Pack on mäng, mida peab mängima pikema aja jooksul, et jõuda väljakutsuvamate algoritmimiste probleemide lahendamiseni. [7] Täpsemalt saab olemasolevate lahendustega tutvuda Triinu Malvi töös „Klientrakenduse loomine algoritmilist mõtlemist arendavale TalTechi veebimängule tALGOtech“.

3 Rakenduse analüüs

Nõuete määramisel on arvesse võetud eelkõige IT Kolledži töötajate soove ja infosüsteemi arendajate täiendusi. Kirjeldatud nõuded sisaldavad endas kogu valmiva rakenduse skoopi, sealhulgas klient- ja serverrakenduse skoopi.

3.1 Funktsionaalsed nõuded

Funktsionaalsed nõuded on kirjeldatud epikutena, mis sisaldavad endas väiksemaid kasutajalugusid. Infosüsteemi kasutajad on jagatud erinevatesse rollidesse. Õpilane tähistab mängu mängivat gümnaasiumiõpilast, kasutaja tähistab kõrgel tasemel algoritmilise mõtlemisega gümnaasiumiõpilast, kes soovib uusi mängu luua ja Administraator kirjeldab süsteemi muutmisõigustega kasutajat, kes saab süsteemi hallata. Autorid on hinnanud kasutajalugude tähtsust, et välja selgitada infosüsteemi olulisemad punktid. S tähistab väiksema, M tähistab keskmise ja L suurema olulisusega kasutajalugu. Administraatori ja kasutaja rollide kasutajalugude tähtsus on jäetud määramata, sest nende teostamise tähtsus on vähem oluline kui õpilasega seotud kasutajalood. Olulisemaks on peetud mängu mängimise funktsionaalsuse valmisarendamist õpilase jaoks, et see oleks piisav kasutamiseks õpitoas.

Nõuete illustreerimiseks on infosüsteemi skeem, kus on joontega tähistatud eelmainitud kasutajate suhtlus infosüsteemiga.

Kasutajalugudega on võimalik tutvuda Lisas 1. Skeemiga on võimalik tutvuda Lisas 2

3.2 Mittefunktsionaalsed nõuded

Mittefunktsionaalsete nõuete juures on kirjeldatud nõudeid, mis on olulised kõikidele kasutajatele, hoolimata nende rollidest.

Mittefunktsionaalsed nõuded:

- Rakenduse kasutamiseks saab valida veebilehitseja järgnevate veebilehitsejate hulgast: Google Chrome, Mozilla Firefox, Microsoft Edge.
- Rakenduse eesmärk ei ole raha teenimine. Rakenduse lähtekood Tallinna Tehnikaülikooli IT kolledži töötajate kasutada ja jagada. (Lisa 7)
- Valmis rakendusele on loodud põhjalik dokumentatsioon, et uute inimeste kaasamine arendusprotsessi oleks võimalikult lihtne.
- Rakenduse kasutamine peab olema iga kasutaja jaoks võimalikult lihtne ja intuitiivne.
- Arendatav serverrakendus on universaalne, seda saavad kasutada erinevatel platvormidel töötavad klientrakendused. Näiteks on tulevikus võimalik luua valmisolevat serverrakendust kasutades mobiilirakendusi nii Androidi kui ka Apple'i loodud operatsioonisüsteemi kasutavatele nutitelefonidele.

3.3 Tehnoloogiate valik

Käesoleva töö raames kaalutakse ainult kaasaegseid suure kasutajabaasiga tehnoloogiaid. Tehnoloogia valiku juures on olulised järgmised aspektid. Tehnoloogia peab olema paindlik, e. Kasuks tuleb töö autori varasem kogemus tehnoloogia kasutamisel, et ei peaks kulutama aega uue tehnoloogia algusest peale õppimisele. Hoolimata sellest, et tehnoloogia valitakse autori varem tuntud tehnoloogiate seast, puututakse arendusprotsessi jooksul väga suure tõenäosusega kokku erinevate funktsionaalsustega, millega autor pole varem kokku puutunud. Seetõttu on oluline ka valitud tehnoloogia võimalikult madal õppimiskeerukus. Samuti arvestatakse tehnoloogia valikul ka kogukonna toega. Suurema kogukonnaga tehnoloogia kasutamisel tekkinud probleemidele on suurem võimalus leida kiiresti lahendus.

Rakenduse autorid otsustasid rakenduse teenuse ja kliendipoolse lahuse hoida. Tänu sellele on tulevikus lihtsam mõlemat poolt teineteisest sõltumatult ka muudele platvormidele edasi arendada ja vajadusel välja vahetada. Samuti muutis see esialgse rakenduse piiritlemise ja arenduse autoritele lihtsamaks. Arenduse käigus ei olnud autorid teineteisest sõltuvad.

Teenuse ja kliendi vahelisel suhtlusel kasutatakse REST arhitektuurstiili. REST arhitektuur lahendab kliendiga suhtlemise probleemi. Sellel on mitmeid eelised, mis on teinud sellest tänapäeval rakendusliideste standardi. RESTi eelised on paindlikkus, arhitektuurstiili kiirem õppimine, paindlik andmeformaate kasutamine ja kiirus. [8] Mõlema töö autoril on varasem kogemus REST arhitektuuri stiilil põhinevate rakenduste arendamisel.

3.3.1 Teenusepoolse programmeerimiskeele valik

Teenusepoolse keele valikul jäetakse välja programmeerimiskeeled, millega autoril puudub kogemus, kuna see pikendaks töö arendusprotsessi õppimisaja võrra. Sellegi poolest arvestatakse programmeerimiskeele valikul õppimiskeerukusega, sest nii on uute funktsionaalsuste juurdeõppimine kiirem. Kõige populaarsemad teenusele sobilikud programmeerimiskeeled, milles töö autoril on kogemus, on (populaarsuse kahanemise järjekorras TIOBE indeksi järgi) [9]:

- Python – lihtsasti kasutatav programmeerimiskeel, mis võimaldab erinevaid programmeerimisstiile. Pythonit saab kasutada mitmes eri valdkonnas näiteks tehisintellekt ja masinõpe, andmete analüüs, mängud jne. [10] Pythonil on erinevaid teke ka veebirakenduste arendamiseks, nendest üks populaarsemaid on Django. [11]
- Java – programmeerimiskeel, mis on suunatud objektorienteeritud programmeerimisstiilile. Javat kasutatakse lisaks veebirakenduste arendamisele ka näiteks mobiilirakendusi ja tööluarakenduste arendamiseks. Veebirakenduste arendamiseks on olemas palju Java raamistikke, mis lihtsustavad arendusprotsessi. [12]
- C# – Microsofti arendatud programmeerimiskeel, mis on suunatud objektorienteeritud programmeerimisstiilile. C# on Java ja programmeerimiskeele C++ edasiarendus. C#-i kasutatakse .Net platvormiga, millele on võimalik juurde lisada teke veebirakenduste arendamiseks. [13]

- JavaScript – programmeerimiskeel, mida kasutatakse eelkõige komplekssete funktsionaalsuste lisamiseks veebilehele. Käitussüsteemid, näiteks Node.js, võimaldavad JavaScriptiga luua ka brauseriväliseid veebiservereid. [14]
- PHP – programmeerimiskeel, mida kasutatakse eelkõige veebirakenduse serveri poolel. [15] PHP on loetletud keeltest kõige vähemlevinum ning selle populaarsus on langustrendis. [9]

Tabel 1. Programmeerimiskeelte võrdlus

Raamistik	Kogemus	Õppimiskeerukus	Kogukonna tugi
Python	Hea	Madal [10]	Suur [16]
Java	Hea	Kõrge [12]	Suur [17]
C#	Rahuldav	Kõrge [13]	Keskmine [18]
Javascript	Rahuldav	Madal [14]	Väga suur [19]
PHP	Vähene	Keskmine [15]	Keskmine [20]

Kuna lõputöö kirjutamiseks ja rakenduse arendamiseks on piiratud aeg, siis kõige paremini sobivad keeled Python ja Java, sest nendes on autoril kõige rohkem kogemust. Kusjuures töö autor on varasemalt arendanud veebirakendusi Java raamistikuga, kuid mitte Pythoni raamistikuga. Mõlemal programmeerimiskeelel on suur kogukonna tugi ja seetõttu on nende kohta võimalik internetist kiiresti kasulikku infot leida. Mõlemal keelel on sobivad lahendused antud rakenduse jaoks olemas, kuid Java näol on tegemist keelega, mis on laialdaselt kasutusel dünaamiliste veebirakenduste arenduses. [21] Pythonil seevastu on palju kasutusvaldkondi, millest ainult üks on veebirakenduste loomine, seega see ei ole nii laialdaselt kasutatud. Lisaks on Eesti tööturul palju rohkem nõudlust Java arendajatele. Näiteks cv.ee tööotsimisportaalil oli 22.veebruaril 2022 aastal 30 töökuulutust, milles nõutud Pythoni kasutusoskust, [22] ning 81 töökuulutusi, mis vajavad Java kasutusoskust. [23] Peale selle asub töö autor ülikooli lõpetamisel tööle Java tarkvaraarendajana. Kõiki neid aspekte arvesse võttes valiti antud töö teenusepoolse osa programmeerimiskeeleks Java.

3.3.2 Arvutitarkvara raamistiku valik

Kuna teenuspooleks programmeerimiskeeleks valiti Java, vaadatakse raamistiku valimisel ainult raamistikke, mis töötavad just selle programmeerimiskeele baasil. Ühed tuntumad Java raamistikud on:

- Spring Boot – mikroteenusebaasil töötav raamistik, mis seob erinevad Springi raamistikud kokku ja mille abil saab luua valmis rakendus väga lühikese aja jooksul. Kõik vajalik on automaatselt konfigureeritud. [24] Spring Booti kõrval saab kasutada Spring Security raamistikku, mille abil saab lisada turvalise sisselogimise funktsionaalsuse. [25]
- Dropwizard – raamistik, mille eesmärk on erinevate teekide ühendamine üheks lihtsasti kasutatavaks raamistikuks, mille abil on võimalik luua RESTi põhiseid veebirakendusi, mis on stabiilsed ja töökindlad. [26]
- Grails – programmeerimiskeele Groovy baasil töötav raamistik, mis võimaldab nii klient- kui ka serverrakenduse arendamist. On ehitatud Spring Boot raamistiku peale ja sisaldab selle funktsionaalsusi. [27]
- Hibernate – raamistik, mis on mõeldud andmebaasiga suhtluse lihtsustamiseks. Võimaldab kasutada korraka ka mitut erinevat andmebaasisüsteemi. Raamistikuga kaasnevad kasulikud tööriistad. Saab kasutada koos teiste raamistikega nagu Spring. [28]
- Struts – avatud lähtekoodiga raamistik, millega arendatakse MVC muustril töötavaid veebirakendusi. Raamistikku saab integreerida ka teistesse Java raamistikesse nagu Spring. [29]

Tabel 2. Java Raamistike võrdlus.

Raamistik	Kogemus	Õppimiskeerukus	Kogukonna tugi	Viimane väljalase
Spring raamistikud	Keskmine	Keskmine	Kõrge [30]	24.02.2022 [31]
Dropwizard	Puudub	Madal	Kõrge [32]	20.12.2021 [33]
Grails	Puudub	Madal	Keskmine [34]	10.03.2022 [35]
Hibernate	Puudub	Keskmine	Keskmine [36]	26.01.2022 [37]
Struts	Puudub	Madal	Keskmine [38]	19.11.2019 [39]

Kõik uuritud tehnoloogiad on enamjaolt sobilikud rakenduse arendamiseks nii raskusastme kui ka funktsionaalsuse poolest. Selle tõttu tehti valik autori oskuste põhjal. Autoril on varasem kogemus Spring raamistikuga nii töös kui ka ülikoolis. Spring raamistikul on hea dokumentatsioon ja ka suur kogukond, selle tõttu on probleemide tekkimisel lihtne abi leida. [32] Lisaks asub töö autor ülikooli lõpetamise järgselt tööle meeskonnas, mis kasutab Spring raamistikke oma töös igapäevaselt.

3.3.3 Arenduskeskkonna valik

Arenduskeskkonnana valitakse välja kõige paremini sobivad integreeritud arendus- ja versioonihalduskeskkonnad.

Versioonihalduseks kasutatakse tarkvara Git. Autoril on varasem kogemus kolme versioonihalduskeskkonnaga: GitLab, GitHub ja BitBucket.

GitHub – suletud lähtekoodiga versioonihalduskeskkond, mida omab samanimeline ettevõtte, mis on alates 2018.aastast Microsofti filiaal. [40] GitHubil on suur ja aktiivne kogukond. GitHubi kasutavad mitmed väga tuntud avatud lähtekoodiga projektid ning ka Google, Facebook ja paljud teised. [41] GitHubi baasfunktsionaalsus on tasuta kõikidele arendajatele, lisafunktsionaalsuste eest tuleb maksta. [42]

GitLab – avatud lähtekoodiga versioonihalduskeskkond, mis pakub tarkvaraarenduseks erinevaid lahendusi ja tööriistu. Nende kõige populaarsem pakett toetab *DevOps* lähenemist, selle abil on võimalik läbida arendusprotsess, sealhulgas rakendust planeerida, käibesse anda ja analüüsida. GitLabist on olemas tasulisi ja tasuta versioone. [41] Tallinna Tehnikaülikoolil on olemas oma GitLabi server, gitlab.cs.ttu.ee. GitLabil on ka olemas pistikprogramm populaarsele arenduskeskkonnale IntelliJ Idea. [43] Töö autoril on kõige suurem kogemus GitLab versioonihalduskeskkonnaga.

BitBucket – versioonihalduskeskkond, mis kuulub Atlassian Corporation Plc'le. Bitbucketi eelis on, et seda saab lihtsasti kasutada teiste Atlassiani toodetega. [41] Bitbucketi lähtekood on avatud litsentsi omanikele [44] BitBucketist on olemas tasulisi ja tasuta versioone. Tudengid ja õpetajad saavad BitBucketit tasuta kasutada. [45]

Rakendus arendatakse kasutades GitLab versioonihalduskeskkonna Tallinna Tehnikaülikooli serverit, sest töö autoril on selle kasutamises kõige suurem kogemus. Käesoleva töö esirakenduse autor valis versioonihalduskeskkonnaks samuti GitLabi. Ülikooli GitLab piirab koolivälistele inimestele ligipääsu lähtekoodile.

Spring rakenduste arenduseks kasutatakse tihti kahte järgmist integreeritud arenduskeskkonda: IBM Eclipse ja JetBrains IntelliJ Idea. IntelliJ Idea on populaarne integreeritud arenduskeskkond, mis on saadaval kahe väljaandena: Community ja Ultimate versioonid. Esimene on tasuta, kuid sisaldab endas vähem funktsionaalsust. Ultimate on tasuline versioon ning sisaldab kõige edasijõudnumaid funktsionaalsusi, mis tuleksid antud arendusprotsessi vältel kasuks. Näiteks Ultimate versioonil on tööriistad SQL-keele ja Spring raamistiku kasutamiseks. [46] Ultimate versioon on tudengitele kasutamiseks tasuta. IBM-i Eclipse on integreeritud arenduskeskkond, mis on avatud lähtekoodiga ja on seetõttu samuti populaarne. Eclipse sisaldab endas baastöölauda, kuhu on võimalik lisada vajalikku funktsionaalsust. IntelliJ kasutamisel on mitmed eelised Eclipse-i ees. IntelliJ IDEAs on efektiivsem refaktoreerimine, automaatne täitmine. Lisaks on IntelliJ IDEA kasutama õppimine kiirem ja lihtsam. [47] Samuti on rakenduse autoril varasem kogemus IntelliJ IDEA seadistamisega ja kasutamisega. Seega rakendus arendati kasutades IntelliJ IDEA integreeritud arenduskeskkonda. Integreeritud arenduskeskkonda saab edasisel arendusel lihtsa vaeva muuta kasutades versioonihaldustarkvara Git.

3.3.4 Andmebaasi valik

Veebirakenduste juures kasutatakse tavaliselt relatsioonilisi andmebaase ehk andmebaase, mis koosnevad andmekogumistest ja nende vahelistest suhetest. Andmebaasis on andmed jaotatud tabelitesse. Peamine moodus andmebaaside ja sinna lisatud andmete lisamiseks ja muutmiseks on SQL. [48] SQLi kasutatakse antud rakenduses näiteks andmebaasi tabelite loomiseks.

Käesoleva lõputöö raames kaalutakse populaarseid andmebaasisüsteeme, mida autor on varem kasutanud. Valikul vaadeldakse kõige rohkem dokumenteeritust, nõutele vastavust ja õppekeerukust. Andmebaasi valiku nõuded antud rakenduse jaoks on suur skaleeritavus, avatud lähtekood, sobivus eelnevalt valitud tehnoloogiatega. Andmebaas peab sobima nii rakendusele, mis hoiab andmebaasis väikeses mahus andmeid mängude kirjeldustega, kui ka rakendusele, mis hoiab andmebaasis mitme reaajas käiva mängu infot. Andmebaasil ei tohi tekkida ühilduvus- ja litsentsiprobleeme, sest need võivad takistada rakenduse kasutamist.

Populaarsed variandid on:

- MySQL – Üks populaarseimad avatud lähtekoodiga andmebaasisüsteeme, mis on stabiilne valik veebiarenduseks. [49] Pärast seda kui Oracle omandas MySQL-li on selle arendusprotsess muutunud aeglaseks, selle tõttu kasutavad avatud lähtekoodiga kogukonna liikmed ka teisi andmebaase. [50]
- MariaDB – MySQL-i autorite poolt loodud parandatud versioon MySQL baasil. MariaDB autorid lubavad, et see jääb igaveseks tasuta kasutamiseks. Andmebaas sobib kasutamiseks nii Microsofti kui ka Unixi laadsete operatsioonisüsteemidega. [51]
- Microsoft SQL Server – Populaarne Microsofti loodud andmebaasisüsteem, mida on lihtne konfigurida ja mida saab kasutada lisaks Microsofti ja Linuxi operatsioonisüsteemiga. [52]
- SQLite – Failipõhine andmebaasisüsteem, mille kasutamisel pole vaja serverit konfigurida. Sobib pigem arendusprotsessi käigus kasutamiseks. [53]
- PostgreSQL – avatud lähtekoodiga lihtsasti kasutatav andmebaasisüsteem. PostgreSQL ei ole nii laialdaselt levinud kui MySQL. [54]

Tabel 3. Andmebaaside võrdlus.

Andmebaasisüsteem	Dokumenteeritus	Kogemus	Nõutele vastavus
MySQL	Väga hea [49]	Keskmine	Hea [49]
MariaDB	Väga hea [51]	Keskmine	Hea [51]
Microsoft SQL server	Hea [52]	Keskmine	Keskmine [52]
SQLite	Hea [53]	Vähene	Halb [53]
PostgreSQL	Hea [54]	Keskmine	Hea [54]

Valituks osutub MariaDB, mille eelisteks teiste ees on andmebaasi avatud lähtekoodist tulenev suur kasutajabaas ja hea dokumentatsioon, autori varasem kogemus vajaliku süntaksiga ja andmebaasi sobiv funktsionaalsus, sealjuures suur skaleeritavus.

Lisaks kasutatakse arendusprotsessi käigus HyperSQL Database-i. See andmebaas on Java baasil töötav mälu põhine andmebaas, mis ühildab paljude teiste andmebaasidega. Selle kasutamine muudab arendusprotsessi arendajale mugavamaks, sest andmebaasi abil saab vajadusel kiiresti andmebaasi disaini muuta ning see ühildub väga hästi Springi funktsionaalsusega. [55]

3.4 Klientrakendus

Klientrakenduse analüüsi käigus toodi välja funktsionaalsed nõuded, mittefunktsionaalsed nõuded ja Tulevikus loodavate funktsionaalsuste nõuded. Kliendipoolsete tehnoloogiate valiku tegi rakenduse kaasautor, kes valis kõige suurema sobivusega tehnoloogiad. Klientrakenduse arendusraamistikuks valiti React.js, kliendirakenduse disainraamistikuks MUI, klientrakenduse animeerimisraamistikuks Anime.js. Analüüsi käigus tehti veebirakenduse ja kasutajakogemuse disain. Vastavate joonistega saab tutvuda lisas 4.

3.5 Suhtlus klient- ja serverrakenduste vahel

Teenusepoolne rakendus suhtleb klientrakendusega läbi API päringute. Päringute defineerimisel võeti aluseks kasutajalood ja klientrakenduse kasutajakogemuse disaini joonised. Tabelis 3 on välja toodud 4 peamist päringut. Nende päringute abil edastatakse server- ja klientpoole vahel kogu info, mis on vajalik mängu mängimiseks ning selle kuvamiseks kasutajale. Tabelis on välja toodud päringu tüüp, päringu sisu, päringu vastus ja aadressid, millele päring tehakse. Vastavalt vajadusele saab edasiarenduste käigus päringuid juurde teha. Näiteks on võimalik lisada sisselogimisega seonduvad päringud.

Tabel 4. Kasutatavad API päringud.

Päringu kirjeldus	Päringu tüüp	Saadetud	Vastus	Aadress
Mängu kontrollimine	POST	[Järjed mängija poolt valitud kastidest]	{sõnum kasutajale mängu seisu kohta, viga kasutaja mängus, viga rakenduses järelejäänud mängukordade arv, taseme raskus numbrina, mängu id, tõeväärtus mängu seisu kohta, vajalikud pildid animatsiooni kuvamiseks, mängulaud, tegevused animatsiooni kuvamiseks}	/api/games/{gamePlayId}
Olemasoleva mängu info küsimine	GET			/api/gamePlay/{gamePlayId}
Uue mängu küsimine	GET		{mängukorra id, taseme info, järelejäänud mängukordade arv, rakenduse veateade, massiiv mängukastidest, massiiv lüngatäidetest, mängu nimi, mängu id andmebaasis, ülesandekirjeldus, ülesande lühikirjeldus autor}	/api/game/{id}
Kõikide mängude küsimine	GET	-	[{veateade, mängu nimi, mängu id, mängu kirjeldus, mängu autor}]	/api/games

Kõikide mängude küsimise päring tehakse, et kasutaja saaks valida kõikidest andmebaasis olevatest mängudest enda jaoks sobiva. Kõikide mängude küsimise päringule vastab server järgmise JSONiga, kus on ära toodud massiiv, mis sisaldab kõiki mängu ja nende infot järgnevalt:

```
[{
  "name": "mängu nimi",
  "id": mängu identifikaator andmebaasis,
  "description": "mängu kirjeldus",
  "author": "mängu autor"
}]
```

Joonis 1. Kõikide mängude edastamise JSON.

Uue mängu küsimise päringuga saab kasutaja alustada oma mängu. Selle päringu järel loob serverrakendus uue mängu, mille id andmebaasis on päringu aadressil ära märgitud. Päringu vastuses on välja toodud kogu vajalik info mängu kuvamiseks klientrakenduse abil järgnevalt:

```

{
  "gamePlayId": mängukorra id,
  "currentLevel": {
    "name": "Taseme nimi",
    "id": id,
    "rules": "Taseme reeglid",
    "attemptsLeft": järelejäänud mängukordade arv
  },
  "errorMessage": "Veateade",
  "boxes": list võimalikest mängukastidest[{
    "id": id,
    "name": "Kasti nimi",
    "randomNumber": suvaliselt genereeritud number,
    "gapFillers": tühi massiiv,
    "innerBoxes": tühi massiiv,
    "isParent": tõeväärtus
  }],
  "gapFillers": massiiv võimalikest kastitäidetest[{
    "id": id,
    "name": "Nimi",
    "isGameObject": tõeväärtus,
    "randomNumber": suvaliselt genereeritud number
  }],
  "name": "Mängu nimi",
  "id": id,
  "task": "Ülesande kirjeldus",
  "description": "Ülesande lühikirjeldus",
  "author": "autor"
}

```

Joonis 2. Mängu info edastamise JSON.

Mängu kontrollimise päringu korral saadab klientrakendus massiivi kastidest, mida mängija on kasutanud oma lahenduses. Info saadetakse järgnevalt:

Massiiv kasutatud kastidest

```

[ {
  "id": kasti id,
  "name": "kasti nimi",
  "gapFillers": massiiv lüngatäidetest
  "innerBoxes": massiiv alamkastidest
  "isParent": tõeväärtus, kas kast võib sisaldada alamkaste
} ]

```

Joonis 3. Kasutatud kastide edastamise JSON.

Päringu vastuseks saadab serverrakendus JSONi, mis sisaldab kasutajale infot mängimise edukuse kohta ja kõike vajalikku animatsiooni kuvamiseks.

```

{
  "message": "mängu õnnestumise/ebaõnnestumise sõnum kasutajale",
  "error": "mängu veakoha teade",
  "attemptsLeft": "järelejäänud mängukordade arv",
  "currentLevelNumber": "taseme raskus numbrina",
  "gamePlayId": "mängukorra id",
  "isCompleted": "tõeväärtus mängu õnnestumise/ebaõnnestumise
kohta,
  "characterPicture": "tegelase pildi URL",
  "gameObjectPicture": "mängu objekti URL",
  "obstaclePicture": "mängus oleva takistuse URL",
  "backgroundPicture": "animatsiooni taustapilt",
  "gameBoard": "massiiv mängulaua kujutamiseks",
  "activities": "massiiv tegevustest, mida animatsioonis kuvada
}

```

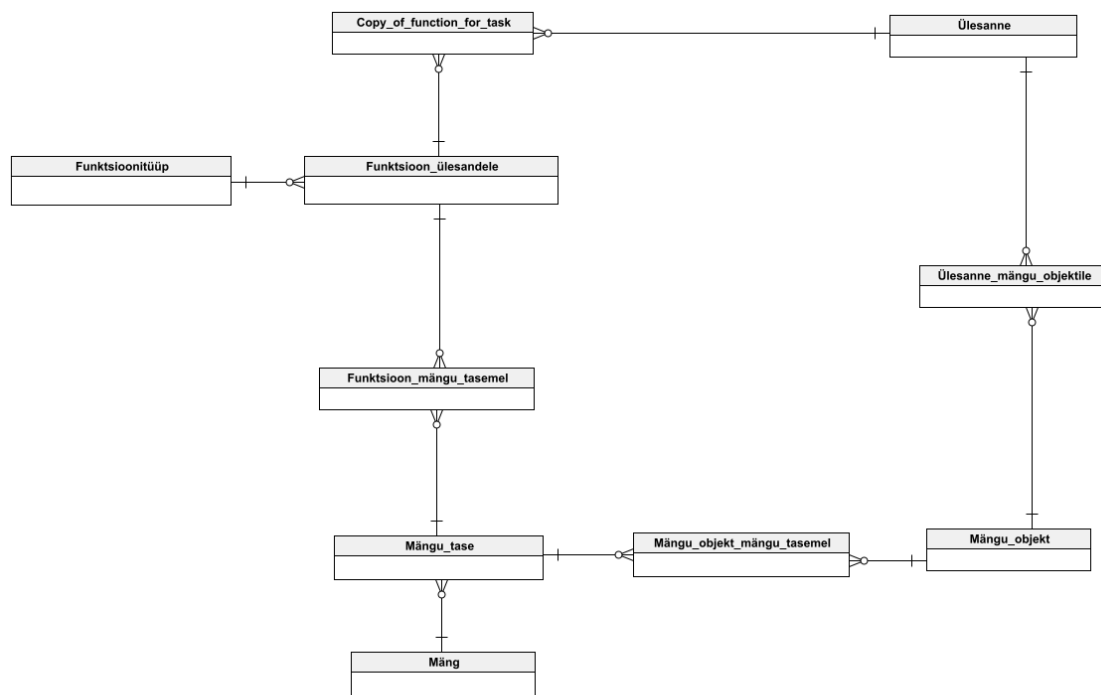
Joonis 4. Mängu vastuse edastamise JSON.

Olemasoleva mängu info küsimisel saab klientrakendus infot parasjagu käiva mängu kohta. Taseme läbimisel või mängukordade otsasaamisel muutub automaatselt mängu tase ja antud päring saadab automaatselt järgmise taseme info. Päringu vastus näeb välja samasugune nagu uue mängukorra küsimise päringu vastus.

Rakenduse arenduse jooksul kasutati serverrakenduse poolse suhtluse testimiseks Postman tarkvara. Autor valis Postmani, sest ta on seda varasemalt kasutanud veebirakenduste arendamisel. Postmani abil saab teha vajalikke päringuid kogu vajaliku sisuga serverrakenduse vastu. Nii on võimalik APIde tööd testida ka siis kui klientrakendus ei ole veel valmisarendatud.

3.6 Andmebaasi disain

Andmebaasi mudel on koostatud pidades silmas kasutajalugusid ja kasutajakogemuse disaini. (Lisa 2 ja Lisa 3) Joonisel 5 on näha andmebaasi mudeli lihtsustatud versioon, mudeli täisversioon on lisa 5. Olemi-suhte diagrammi loomiseks kasutati tööriista Vertabelo. Vertabelo abil saab lihtsalt joonistada andmebaasi graafilisi kujutisi ja selle abil genereerida SQL laused andmebaasi loomiseks. Enne arendusprotsessi loodi esialgne andmebaasimudel, kuhu viidi arendusprotsessi jooksul sisse vajalikke muudatusi.



Joonis 5. Andmebaasi lihtsustatud olemi-suhte diagramm.

Andmebaasi eesmärk käesoleva lõputöö raames on ettemääratud sisuga mängude salvestamine. Nende mängude seast saab kasutaja valida endale sobiva mängu mängimiseks. Andmebaasi ülesehituse juures on oluline nõue, et rakendus töötaks erinevate mängudega, millel on erinevad nõuded. Näiteks võib andmebaasis olla kalastustemaatikaga mäng ning kosmosetemaatikaga mäng. Rakendus kasutab mõlemat mängu samamoodi, kogu muutuv info tuleb andmebaasist. Andmebaasi põhiinfo tuleb tabelist peakirjaga „Game“, kus on ära toodud mängu põhiinfo nagu mängu nimi ja ülesande kirjeldus kui ka hiljem mängu animeerimiseks vajalikud pildid. Piltide asemel salvestatakse andmebaasi pildi internetilink. Tabel „Game“ on ühendatud tabeliga „Game_level“, kus hoitakse mängu juurde käivaid raskusastmeid. „Game_level“ tabel on vahetabeli abil seotud tabeliga „Game_object“, kus hoitakse mängus kasutatavaid põhiobjekte. Nende objektide ümber toimub mängu kogu põhitegevus. Põhiobjekte võib mängus olla ka mitmeid. Näiteks kui põhiobjekt on puu, on objekti juures tehtavad võimalikud tegevused augu kaevamine, puu auku panemine, kastmine jne. Konkreetse mänguobjekti tegevused hoitakse tabelis „Task“, mis on tabeliga „Game_object“ seotud vahetabeli abil. Vajalikku infot mängu mängimise tegevusteks hoitakse tabelis „Function“. See tabel on vahetabeli abil seotud raskusastme tabeliga. Selle tõttu võib vastavalt raskusastmele mängus lubatud tegevuste hulk olla piiratud, muutes nii mängu

lihtsamaks või raskemaks. „Function“ tabeli kirjed on funktsioonid ja need jagunevad kaheks: kastid ja lüngatäiteid. Kastid on kõik tegevused, mida kasutaja saab kasutada mängu läbimiseks. Lüngatäited on abilised, mis käivad kastide juurde ja muudavad kasti funktsionaalsust. Kuna kasti ja lüngatäite salvestamiseks vajalikud andmed on sarnased, hoitakse neid andmebaasis ühes tabelis. Funktsioonide defineerimiseks rakenduses on ära toodud teatud hulk funktsiooni tüüpe, mille alla funktsioon peab kuuluma. Funktsionitüüpe hoitakse tabelis „Function_type“. Võimalikke kasti funktsionitüüpe ja nende tähendusi näeb tabelis NUMBER. Võimalikke lüngatäite funktsionitüüpe ja tähendusi näeb tabelis NUMBER. Üks võimalik kasti funktsionitüüp on „Special“. Sellise tüübiga kaste kasutatakse mänguobjekti mingi tegevuse täitmiseks. Tegevuse kokkuviiamiseks kastiga on tabel „Function“ ühendatud tabeliga „Task“ vahetabeli abil. Kõikides vahetabelites on välja toodud ka kogused.

Tabel 5. Kastide tüübid.

Nimi	Mida see teeb?	Muutujad
Move	Liikumine	Teepikkus, suund
For each	Tsükkel	mänguobjektid, mille alusel tsükkel tehakse; tegevused mida korratakse
While	Tsükkel	mänguobjektid, mille alusel tsükkel tehakse
Special	Spetsiaalne tegevus mänguobjekti mingi tegevuse täitmiseks	-
If	Kontrollib kas mingit tegevust saab teha	tegevus, mida kontrollitakse

Tabel 6. Lüngatäidete tüübid.

Nimi	Mida see teeb?
Forward	Näitab liikumise suunda edasi
Backward	Näitab liikumise suunda tagasi
Number	Ükskõik mis number
True	Tõeväärtus tõene
False	Tõeväärtus väär
Distance	Kaugus millestki, näiteks kaugus algusest
NoObstacle	Aitab kontrollida takistuse olemasolu

3.7 Analüüsi kokkuvõtte

Analüüsi käigus kirjeldati rakenduse funktsionaalsed ja mittefunktsionaalseid nõuded. Nõuete põhjal loodi infosüsteemi skeem (Lisa 3), mis aitab loodavat infosüsteemi piltlikustada. Teenusepoolse rakenduse tehnoloogiate valiku juures toodi välja olulisemad aspektid valikute juures, kirjeldati ja võrreldi sobivaid tehnoloogiaid ning valiti välja kõige paremad variandid. Rakenduse programmeerimiskeeleks valiti Java, arendusraamistikuks Spring Boot, versioonihalduseks GitLab, integreeritud arenduskeskkonnaks IntelliJ Idea, andmebaasiks MariaDB, andmebaasiks arendusperioodi jooksul HyperSQL Database. Analüüsi käigus kirjeldatud nõuete põhjal planeeriti server- ja klientrakenduse vaheline suhtlus, mis toimub läbi API päringute. Lisaks kirjeldati andmebaasi olemi-suhte diagramm.

4 Rakenduse arendus

Peatükis on kirjeldatud serverrakenduse arendusprotsessi ja tutvustatud rakenduse eri osasid. Välja on toodud näidismängu “Kosmoseroboti programmeerimine” kirjeldus. Lühidalt on tutvustatud ka klientrakenduse arendusprotsessi.

4.1 Projekti loomine

Teenusepoolne rakendus tegeleb äri loogika ja andmetele ligipääsuga. Rakendus on ehitatud Spring Boot raamistikuga ning kasutab MariaDB andmebaasi, mis on vajalik kogu info salvestamiseks.

Spring Bootiga on väga lihtne ja kiire luua veebirakendusi ja sellega kaasneb Tomcat või Jetty server. [56] Rakenduse saab luua nii integreeritud arenduskeskkonnas kui ka veebilehel <https://start.spring.io/>, [57] loomisel saab valida vajalikud teegid ja tööriistad. Vajadusel saab arenduse käigus teeke juurde lisada või eemaldada.

Joonisel 6 on välja toodud rakenduse esialgne konfiguratsioon: projekti ehitamise tööriist Gradle, programmeerimiskeel Java, Spring Boot versioon 2.6.4, projekti andmed ja soovitud teegid. Teekide valikul arvestati sellega, et võimalikult palju funktsionaalsust oleks projektis algusest peale kättesaadav.

Valitud teegid:

- Lombok – Java kommentaaride kogu, mis aitab vähendada ebavajalikku koodi. [57]
- Spring Web – teek, mille abil saab ehitada veebirakendusi. [57]
- HyperSQL Database – lihtsasti kasutatav Java baasil olev SQL andmebaasisüsteem. [57]
- Spring Data JPA – tehnoloogia, mille abil saab Java objekte andmebaasi objektideks vastendada. [57]

Sellise konfiguratsiooni järgi loodi baasprojekt, kus on olemas esmane baasklass, selle testimise klass, Gradle Wrapper failid, mis hoiavad infot Gradle versiooni kohta, ja konfiguratsioonifailid

The screenshot shows the Spring Initializr configuration interface. On the left, under 'Project', 'Maven Project' is selected. Under 'Language', 'Java' is selected. Under 'Spring Boot', '2.6.6' is selected. The 'Project Metadata' section includes: Group (com.example), Artifact (demo), Name (demo), Description (Demo project for Spring Boot), and Package name (com.example.demo). Under 'Packaging', 'Jar' is selected, and under 'Java', '17' is selected. On the right, the 'Dependencies' section lists 'Lombok' (Developer Tools), 'Spring Web' (Web), 'HyperSQL Database' (SQL), and 'Spring Data JPA' (SQL). A button 'ADD DEPENDENCIES... + B' is visible at the top right of the dependencies section.

Joonis 6. Spring initializr kasutajaliides.

Sellise konfiguratsiooni järgi loodi baasprojekt, kus on olemas esmane baasklass, selle testimise klass, Gradle Wrapper failid, mis hoiavad infot Gradle versiooni kohta, ja konfiguratsioonifailid.

Rakenduse arenduse jooksul lisati rakenduse konfiguratsiooni veel järgnevad teegid:

- MariaDB Java Client – selle abil ühendatakse rakendus andmebaasiga. [57]
- Flyway Core – versioonikontroll andmebaasile. [57]
- ModelMapper – teek objekti teisendamiseks ühest klassist teise. [57]

Rakenduse failid jaotati edaspidi selguse hoidmiseks järgnevatesse kaustadesse:

- BLL - rakenduse ärioloogika
- Controllers - REST API kontrollid
- DAL - andmeedastuskiht andmebaasi ja ärioloogika vahel
- GameBrain - mängu loogikaga seotud failid
- Models - andmete mudelobjektid

4.2 Andmebaasiga ühendamine

Arendusprotsessi vältel kasutati HyperSQL Database-i, sest see on sobilik ja mugav. Näiteks on selle abil võimalik kiiresti andmebaasis muudatusi sisse viia, kuna mälu põhine andmebaas luuakse igal rakenduse käivitamisel ja kustutatakse rakenduse sulgemisel. [58] Lisaks kasutati rakenduses JPA teeki, mis lisab projekti võimalusi erinevate andmebaasidega ühte moodi suhtlemiseks. [59] HyperSQL Database teek ja Springi JPA teek lisati projekti loomisel.

Arendusprotsessi lõpufaasis võeti kasutusele andmebaas MariaDB. Andmebaasi ülesseadmiseks võeti kasutusele konteineriseerimistehnoloogia. Loodi docker-compose fail, mis lõi Docker-i tarkvara abil konteineri, kus sees töötab andmebaas. Docker-compose faili sisu arendusprotsessi käigus on näha joonisel 7.

```
1 version: '3.8'
2
3 services:
4   db:
5     image: mariadb:10.6
6     container_name: game-mariadb-dev
7     environment:
8       MYSQL_DATABASE: game
9       MYSQL_ROOT_PASSWORD: root
10      MYSQL_USER: admin
11      MYSQL_PASSWORD: admin
12     ports:
13       - "13306:3306"
14     command: ['mysqld', '--character-set-server=utf8mb4', '--collation-server=utf8mb4_unicode_ci']
```

Joonis 7. Docker-compose andmebaasi loomiseks.

Pärast andmebaasi loomist ühendati see rakendusega. Selle jaoks lisati build.gradle faili MariaDB draiveri sõltuvus ning application.properties faili info kuidas konkreetse andmebaasiga ühenduda. Application.properties faili sisu on näha joonisel 8. Seejärel oli rakendus andmebaasiga ühendatud.

```
1 spring.datasource.url=jdbc:mariadb://localhost:13306/game?useMySQLMetadata=true
2 spring.datasource.username=admin
3 spring.datasource.password=admin
```

Joonis 8. Application.properties faili sisu.

Edasiseks andmebaasi haldamise lihtsustamiseks võeti kasutusele avatud lähtekoodiga andmebaasi versioonihalduse teek Flyway. Flyway võimaldab teha automaatseid versioonipõhiseid andmebaasi migratsioone. Teek kasutab enne rakenduse käivitamist migratsioonide tegemiseks migratsioonifaile, mille Flyway tunneb ära asukoha ja nimede järgi. Migratsioonifailides on antud rakenduses kasutatud SQL-i. Lisaks kasutab Flyway teek metatabelit andmebaasi versioonide dokumenteerimiseks. Selle tõttu on andmebaasi lisatabel nimega flyway_schema_history. [60] Kuna antud rakenduse andmebaasi mudel on arendusprotsessi jooksul HyperSQL andmebaasi kasutades täiustatud, on migratsioone ainult 1. Migratsiooni nimi on V1__create_initial_table.sql ning see asub resources/db/migration kaustas. Tulevased migratsioonid tuleb õigete nimedega samasse kausta lisada.

4.3 Rakenduse osad

Rakenduse osade kirjeldus on jaotatud kaheks osaks. Esimeses osas kirjeldatakse andmete edastamisega seotuid rakenduse osasid. Kirjeldamisel alustatakse andmebaasile kõige lähedamal olevast kihist ja lõpetatakse kõige teenuselähedasema kihiga. Teises osas kirjeldatakse ärioloogika osasid.

4.3.1 Andmete edastamise osad

Igale andmebaasi tabelile vastab rakenduses Java klass, milles tabeli read on kujutatud klassi väljadena. Klasside loomisel kasutati Lombok teegi annotatsioone ja JPA teegi annotatsioone. Lombok lisab klassile selle kasutamiseks vajalikud meetodid, näiteks väljade saamine, väljade muutmine jne. Javas peab vajalikke meetodite tegemiseks kirjutama mitmeid ridu koodi, Lombok on selle probleemi lahendanud annotatsioonidega. [61] JPA teegi abil märgiti ära klassi vastavus andmebaasile. JPA on ORM standard. Selle kasutamisel ei pea koodi kirjutama andmebaasi põhiselt, vaid saab rakendust arendada loogiliselt ja see annotatsioonide abil andmebaasiga ühildada. [62] Klassi Function koodinäide on joonisel 9. Joonisel on näha Lombok teegi annotatsioone, nagu @Data, @NoArgsConstructor jne, ning JPA teegi annotatsioone @Entity, @OneToMany jne.

```

@Data
@NoArgsConstructor
@AllArgsConstructor
@Builder
@Entity
public class Function {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Column(name = "function_id")
    private Long FunctionId;

    @NonNull
    @Column(name = "is_function_filler")
    private Boolean IsFunctionFiller;

    @Column(name = "name", length = 200)
    private String Name;

    @Column(name = "picture", length = 500)
    private String Picture;

    @Column(name = "is_parent")
    private Boolean isParent;

    @Column(name = "animate_game_object")
    private Boolean AnimateGameObject;

    @OneToMany(mappedBy = "Function", cascade = CascadeType.ALL)
    private List<FunctionForGameLevel> FunctionForGameLevel;

    @ManyToOne
    @JoinColumn(name = "functionTypeId")
    private FunctionType FunctionType;
}

```

Joonis 9. Koodinäide Function klassi mudel.

Andmetele ligipääsuks andmebaasist loodi iga tabeli kohta repositoorium. Selle jaoks kasutati CrudRepository liidest, mis defineerib ära vajalikud funktsioonid andmebaasiga suhtlemiseks. Koodinäide klassi Function repositooriumist on joonisel 10. Repositooriume kasutab äriloogika kiht. Mängu äriloogika sisu kirjeldatakse täpsemalt järgmises peatükis. Mõned rakenduses kasutatud funktsioonid andmebaasiga suhtlemiseks:

- findAll - leiab kõik tabelis olevat kirjed,
- findById - võtab funktsiooni sisendiks mingi identifikaatori ja leiab selle järgi andmebaasist kirje,
- save - salvestab funktsiooni sisendiks antud kirje andmebaasi.

```

package game.DAL;

import game.Models.Function;
import org.springframework.data.repository.CrudRepository;

public interface FunctionRepository extends CrudRepository<Function, Long> {
}

```

Joonis 10. Koodinäide Function klassi repositoorium.

API päringute tegemiseks on klass GameController, kus on ära defineeritud kõik kliendiga suhtlemiseks kasutatavad päringud. Kõiki päringuid kirjeldati täpsemalt peatükis 3.5.2. Kontroller kasutab GameService klassi mängu äriloogika teostamiseks. GameService klassis on võimalik saada mängude nimekiri andmebaasist ning kasutada mängu äriloogika klasse mängu alustamiseks, muutmiseks ja mänguliigutuste tegemiseks. GameService klass viib andmed enne kontrolleri klassi saatmist kujule, mille järgi tehakse päringus saadetak JSON.

4.3.2 Mängu loogika

Mängu äriloogika on ära jaotatud GameBrain ja GameBoard klassi vahel. GameBoard klassis hoitakse mängu mängulauda, mille peal tehakse kasutaja saadetak tegevusi, GameBrain klassi kaudu alustatakse GameBoard klass uue mängu korral, suunatakse sinna tegevusi ja kontrollitakse mängu õnnestumist. Uue mängu alustamisel pärib GameBrain klass valitud mängu identifikaatori järgi mängu info ja alustab selle põhjal uue mänguga GameBoard klassis. Mängu alustatakse tasemel, mille raskusaste on 2. Vastavalt tasemenõuetele määratakse ära funktsioonid ja nõuded mängu mängimiseks. Funktsioonid jaotatakse kastideks ja kastide lüngatäideteks, nende abil saab kasutaja mängu käiku kirja panna. Nõuete all on mõeldud mängu kirjeldust, takistuste panekut mängulauale, mänguobjektide arvu ja muud vajalikku mängu läbimise kontrollimiseks.

Mängulaua liigutuste tegemiseks saadab kasutaja tegevuste järjekorra. GameBrain klass teeb saadetak tegevused GameBoardil läbi ja saadab kasutajale tagasi mängu tulemuse sõnumina. Mängu ebaõnnestumisel on kasutajal võimalik varem määratud arv kordi uuesti proovida. Ebaõnnestumise teates sisaldub veakoht, mille järgi antakse kasutajale vihje, kuidas uuel katsel tegevuste järjekorda parandada. Kui järelejäänud mängukordi on

0, on mängijal võimalik mängu kergemal tasemel mängida. Mängu õnnestumisel kuvatakse mängijale õnnestumise teade ja mängijal on võimalik mängu raskemal tasemel mängida. Uuel tasemel muudab GameBrain klass vastavalt tasemele mängijal kasutatavaid funktsioone ja mängu nõudeid.

4.4 Näidismäng “Kosmoseroboti programmeerimine”

Arenduse lõpus lisati andmebaasi mäng “Kosmoseroboti programmeerimine”, mille kontseptsiooni mõtles välja klientrakenduse autor ja täiendas käesoleva töö autor. Mängu eesmärk on olla mängijatele võimalikult huvitav, nii et mängija õpiks algoritmilist mõtlemist seda ise tähele panemata. Mängus on tegelaseks kosmonaudist õpilane, kes peab läbima eksami Kuu peal. Eksami sisuks on Kuule puuviljaia loomiseks ploomipuude istutamine. Mängu läbimiseks peab mängija kirjutama programmi, mille abil oskaks kosmonaudi robot ploomipuid istutada.

Mängu üldkirjeldus: „Aita kosmonaudiõpilasel eksamil kõrvale hiilida. Kosmonaut peab kooli lõpetamiseks läbima Kuu peal toimetamise eksami. Tal tuli aga idee saata kuurobot enda asemel ülesandeid tegema. Ta teab, et ta saab koolist vajaliku roboti laenutada, kuid puudu on programm, mis paneks roboti eksamiülesandeid sooritama. Pooleteise tunni pärast stardib eksamirakett Kuule, nii et tal on roboti programmeerimiseks jäänud napilt 45 minutit. Aita kosmonaudil „spikerdada”!”

Mängu põhiobjekt on ploomipuu, mida võib istutada iga 2m tagant. Ploomipuu istutamise erifunktsioonikastid on „augu kaevamine”, „puu auku panemine”, „augu kinniajamine” ja „kastmine”. Kõiki tegevusi tuleb ühe ploomipuu istutamiseks teha üks kord. Lisaks erikastidele on mängus tüüpilised kastid liikumiseks, tsükli kujutamiseks ja kontrolliks.

Mäng on jaotatud kolme tasemesse, mis vastavad mängija algoritmilise mõtlemise oskustele. Mängija alustab keskmiselt tasemelt ja saab vastavalt tulemustele liikuda kergemale või raskemale tasemele. Mängu tasemete spetsiifika on täpsemalt kirjeldatud tabelis 7.

Tabel 7. Kosmosroboti programmeerimise mängu tasemed.

	Kerge raskusaste	Keskmine raskusaste	Raske raskusaste
Kastide ja lüngatäidete arv	Ette on antud õige arv kaste ja lüngatäited	Pole teada mitut korda ühte kast või lüngatäidet kasutada tuleb	Pole teada mitut korda ühte kast või lüngatäidet kasutada tuleb
Istutavate ploomipuude arv	3	Arv pole teada, genereeritakse suvaline number	Arv pole teada, genereeritakse suvaline number
Tsüklite kasutus	Ei	Jah	Jah
Takistused puu istutamisel	Ei	Ei	Jah
Kastid	3x "Kaeva auk" 3x "Aseta puu auku" 3x "Aja auk kinni" 3x "Kasta" 3x "Liigu_meetrit_ "	"Kaeva auk" "Aseta puu auku" "Aja auk kinni" "Kasta" "Liigu_meetrit_ " "Kuni_on istutamata"	"Kaeva auk" "Aseta puu auku" "Aja auk kinni" "Kasta" "Liigu_meetrit_ " "Kuni_on istutamata" "Kui saab_ "
Lüngatäited	2x "2" 2x "Edasi" 1x "Kaugus alguspunktist" 1x "Tagasi"	"2" "Edasi" "Kaugus alguspunktist" "Tagasi" "Ploomipuud"	"2" "Edasi" "Kaugus alguspunktist" "Tagasi" "Ploomipuud" "Istutada"

Mängu „Kosmosroboti programmeerimine” mängimise vaateid rakenduse klientrakendusest on lisas 4.

4.5 Klientrakenduse arendus

Paralleelselt serverrakenduse arendusega toimus klientrakenduse arendus. Klientrakenduse ülesehituse tüüp on SPA ehk rakendus, mis laeb korraga ära lehe HTMLi ja JavaScripti koodi. Rakenduse projekt loodi Create React App lahenduse abil npm ehk node package manager kaudu käsureal. Veebileht ehitati üles eraldiseisvaid taaskasutatavaid komponente kombineerides. Klientrakenduse arenduse vältel kasutati fiktiiv-serverrakendust, mille abil sai klientrakenduse tööd testida. Klientrakenduse arenduse raames loodi ka animatsioon, mille abil antakse tagasisidet mängija mängu sooritamisele. Klientrakenduse arendusprotsessi kohta saab täpsemalt lugeda Triinu Malvi töös „TALTECHI Algoritmilise mõtlemise VEEBIMÄNGU serverrakenduse arendus”. Töös on kirjeldatud veel kasutajaliidese komponentidesse jagamist, rakenduse struktuuri, suhtlust veebiteenusega, lahendusalgoritmi koostamise võimaluse loomist ja raskusastmete eristamist kasutajaliidese.

5 Hinnang valminud rakendusele

Lõputöö tulemust saab hinnata mitmest aspektist: planeeritud nõuetele vastavus, tulevaste muudatuste võimalikkus ja rakenduse tehnoloogiate kohasus.

Veebirakenduse arendamise jooksul kasutatud tehnoloogiad on sobivad ja aktuaalsed. Java Spring Boot raamistik ja MariaDb andmebaas sobisid hästi käesoleva probleemi lahendamiseks, arendusprotsess nendega oli efektiivne ja mugav. Tehnoloogiate valiku juures arvestati ka kogukonna toega ja tehnoloogiate uuenduste relevantsusega, seega juba enne arendusprotsessi algust oli suur tõenäosus, et tulevikus on kasutatud tehnoloogiate uuendamine võimalikult lihtne.

Valminud rakenduse lähtekood on lisas 7, veebilehe URL on lisas 8.

5.1 Valminud veebirakenduse nõuetele vastavus

Valminud veebirakenduses on sobiva funktsionaalsusega, nii et seda on võimalik kasutada Tallinna Tehnikaülikooli IT kolledži erialasid tutvustavates töötubades. Funktsionaalsetest nõuetest said täidetud kõik kasutajalood, mis olid seotud epikuga 1 “Õpilasena tahan mängida mängu, et arendada oma algoritmilise mõtlemise oskust” ja epik 2 “Õpilasena tahan saada mängult interaktiivset tagasisidet, et mul oleks huvitav”. Täidetud said nii suure olulisusega kui ka väikese olulisega funktsionaalsed nõuded. Mittefunktsionaalsetest nõuetest said käesoleva lõputöö jooksul täidetud nõuded numbriga 2, 3, 5. Rakendus on üleval Tallinna Tehnikaülikooli IT-Kolledži veebihoidlas ning on ligipääsetav kooliga töötajatele ja tudengitele, koodihoidla link on lisas 5. Valminud rakenduse dokumentatsioon on käesolev lõputöö. Juhend rakenduse kasutamiseks on lisas 9. Selle põhjal on uutel arendajatel võimalik arendusprotsessi jätkata. Valminud serverrakendus töötab API päringute põhjal, selle tõttu on võimalik lisada mitmeid erinevaid klientrakendusi, mille tehnoloogiate valik ei sõltu serverrakendusest. Samuti said täidetud mittefunktsionaalsed nõuded 1 ja 4. Need said täidetud klientrakenduse arenduse käigus.

Funktsionaalsetest nõuetest jäid saavutamata kõik kasutaja ja administraatori rolliga seotud nõuded.

5.2 Edasiarendus võimalused

Esmased edasiarendusvõimalused on puuduvate funktsionaalsete nõuete funktsionaalsuse lisamine. Kasutajate ja administraatori sisselogimise saab kergesti lisada Spring Security teegiga, mängude lisamiseks tuleb juurde luua vastavaid APisid, kuid info andmebaasi lisamise ärioloogika ei ole eriti keeruline.

Kõige lihtsam edasiarenduse võimalus on andmebaasi uute mängude lisamine. Selle jaoks on vaja välja mõelda uue mängu kirjeldus, funktsioonid, mänguobjekt ja selle ülesanded. Seejärel tuleb need sisestada andmebaasi ning uut mängu on võimalik mängida.

Edasiarenduse käigus on tulevikus võimalik täpsemalt välja tuua mängija algoritmilise mõtlemise tase ning jagada soovitusi, mida teha selle arendamiseks. Antud lõputöö raames ei oleks selle lisamiseks piisavalt aega jäänud, kuid suurem viitamine mängu algoritmilise mõtlemise aspektile oleks kasulik edasiarendus nii IT-teaduskonnale kui ka õpilastele, kellele selline info aitaks tulevikus näiteks eriala valikul läbimõeldumaid otsuseid teha.

Tööd on võimalik edasi arendada ka klientrakenduse vaatepunktist. Tänu server- ning klientrakenduse eraldatusele on võimalik juurde arendada erinevatel platvormidel töötavaid klientrakendusi. Mängu sihtgrupile ehk gümnaasiumiõpilastele oleks meelepärane mängida mängu ka nutitelefonil. Suure infotehnoloogia huvi korral oleksid õpilased huvitatud mängu mängimisest ka väljaspool töötuba.

Hetkel toetab rakendus ainult eesti keelt. Kasulik oleks tulevikus juurde lisada ka inglise ja vene keele tugi. Pärast seda oleks rakendust võimalik kasutada ka võõrkeelsete koolide töötubades IT hariduse tutvustamisel.

6 Kokkuvõte

Käesoleva lõputöö eesmärk oli lahendada õpilase algoritmilise taseme hindamise probleem vastavastanduse mängu abil. Töö käigus tutvustati algoritmilist mõtlemist ja pakuti lahenduseks algoritmilise mõtlemise mängu rakendus. Rakenduse analüüsi käigus toodi välja rakenduse nõuded, valiti sobivad tehnoloogiad probleemi lahendamiseks, kirjeldati andmebaasi ja suhtlust klient- ja serverrakenduse vahel. Analüüsile järgnes rakenduse arendusprotsess. Lõputöös on kirjeldatud projekti loomist, andmebaasiga ühendamist, rakenduse erinevaid osasid, näidismängu „Kosmoseroboti programmeerimine” ja klientrakenduse arendusprotsessi.

Töö alguses püstitatud eesmärk täideti, töö käigus valmis koostöös rakenduse kaasautoriga algoritmilise mõtlemise mäng „tALGO tech”. Käesoleva töö raames arendati mängu serverrakendus. Rakenduse skooopi jäi mängude salvestamine, mängude ärioloogika ja suhtlus klientrakendusega läbi päringute. Klientrakendus arendati Triinu Malvi töös „Klientrakenduse loomine algoritmilist mõtlemist arendavale TalTechi veebimängule tALGOtech”. Mängu abil saavad Tallinna Tehnikaülikooli IT-kolledži töötajad läbi viia IT-erialasid tutvustavaid töötubasid. Arendusprotsessi käigus täideti kõige olulisemad nõuded. Rakenduse edasiarendamist saab kasutada ka õppetöö praktikumi osana. Mäng on kättesaadav läbi avaliku domeeni.

Kasutatud kirjandus

- [1] K. Parve, „Raalmõtlemine on kõigile,“ Õpetajate leht, 01 01 2022. [Võrgumaterjal]. Available: <https://opleht.ee/2022/01/raalmotlemine-on-koigile/>. [Kasutatud 15 05 2022].
- [2] S. Grover, „edsurge,“ 25 02 2018. [Võrgumaterjal]. Available: <https://www.edsurge.com/news/2018-02-25-the-5th-c-of-21st-century-skills-try-computational-thinking-not-coding>. [Kasutatud 15 03 2022].
- [3] „Why Use Games to Teach?,“ Serc, 07 05 2018. [Võrgumaterjal]. Available: <https://serc.carleton.edu/introgeo/games/whygames.html>. [Kasutatud 15 03 2022].
- [4] „Autothinking,“ AutoThinking-EESTI, 12 01 2020. [Võrgumaterjal]. Available: <https://sites.google.com/view/autothinking-estonian/avaleht?authuser=0>. [Kasutatud 02 05 2022].
- [5] „About Scratch,“ Scratch, [Võrgumaterjal]. Available: <https://scratch.mit.edu/about>. [Kasutatud 02 05 2022].
- [6] „What is Hopscotch?,“ Hopscotch Technologies, Inc, 06 01 2021. [Võrgumaterjal]. Available: <https://help.gethopscotch.com/article/5-what-is-hopscotch>. [Kasutatud 02 05 2022].
- [7] „The Pack,“ New York Hall of Science, 2022. [Võrgumaterjal]. Available: <https://nysci.org/the-pack>. [Kasutatud 2 05 2022].
- [8] „What is REST API?,“ akana, 20 11 2020. [Võrgumaterjal]. Available: <https://www.akana.com/blog/what-is-rest-api>. [Kasutatud 22 02 2022].
- [9] TIOBE, „TIOBE Index for April 2022,“ TIOBE, 04 2022. [Võrgumaterjal]. Available: <https://www.tiobe.com/tiobe-index/>. [Kasutatud 18 04 2022].
- [10] „General Python FAQ,“ Python Software Foundation, [Võrgumaterjal]. Available: <https://docs.python.org/3/faq/general.html>. [Kasutatud 22 04 2022].
- [11] „Django,“ Django Software Foundation, [Võrgumaterjal]. Available: <https://www.djangoproject.com>. [Kasutatud 18 04 2022].
- [12] „Java Introduction,“ W3Schools, [Võrgumaterjal]. Available: https://www.w3schools.com/java/java_intro.asp. [Kasutatud 18 04 2022].
- [13] „A tour of the C# language,“ Microsoft, 18 03 2022. [Võrgumaterjal]. Available: <https://docs.microsoft.com/en-us/dotnet/csharp/tour-of-csharp/>. [Kasutatud 19 04 2022].
- [14] „Introduction,“ Mozilla Foundation, 14 04 2022. [Võrgumaterjal]. Available: <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Introduction>. [Kasutatud 18 04 2022].

- [15] „What can PHP do?“, The PHP Group, [Võrgumaterjal]. Available: <https://www.php.net/manual/en/intro-whatcando.php>. [Kasutatud 18 04 2022].
- [16] „Python’s community is vast; diverse & aims to grow; Python is Open.“, Python Software Foundation, [Võrgumaterjal]. Available: <https://www.python.org/community/>. [Kasutatud 14 03 2022].
- [17] „Java SE Community at a Glance“, Oracle, [Võrgumaterjal]. Available: <https://www.oracle.com/java/technologies/community-glance.html>. [Kasutatud 14 03 2022].
- [18] D. Ramel, „Survey Reveals Bigger C# Community, Most and Least Popular Uses“, Visual Studio Magazine, 26 04 2021. [Võrgumaterjal]. Available: <https://visualstudiomagazine.com/articles/2021/04/26/slashdata-421.aspx>. [Kasutatud 14 03 2022].
- [19] „Javascript“, DEV community, [Võrgumaterjal]. Available: <https://dev.to/t/javascript>. [Kasutatud 19 04 2022].
- [20] „10 Programming languages with the biggest community support“, Techgig, 20 11 2020. [Võrgumaterjal]. Available: <https://content.techgig.com/technology/10-programming-languages-with-the-biggest-community-support/articleshow/78908491.cms>. [Kasutatud 14 03 2022].
- [21] „How to build a Web Application Using Java“, JavaTpoint, [Võrgumaterjal]. Available: <https://www.javatpoint.com/how-to-build-a-web-application-using-java>. [Kasutatud 18 04 2022].
- [22] „Tööpakkumised“, CV_Online, 22 02 2022. [Võrgumaterjal]. Available: https://cv.ee/et/search?limit=20&offset=0&categories%5B0%5D=INFORMATION_TECHNOLOGY&keywords%5B0%5D=Python&fuzzy=false&isHourlySalary=false&isRemoteWork=false. [Kasutatud 22 02 2022].
- [23] „Tööpakkumised“, CV-Online, 22 02 2022. [Võrgumaterjal]. Available: https://cv.ee/et/search?limit=20&offset=0&categories%5B0%5D=INFORMATION_TECHNOLOGY&keywords%5B0%5D=Java&fuzzy=false&isHourlySalary=false&isRemoteWork=false. [Kasutatud 22 02 2022].
- [24] „Difference between Spring MVC and Spring Boot“, GeeksForGeeks, 23 11 2020. [Võrgumaterjal]. Available: <https://www.geeksforgeeks.org/difference-between-spring-mvc-and-spring-boot/>. [Kasutatud 23 02 2022].
- [25] „Spring Security“, VMware, [Võrgumaterjal]. Available: <https://spring.io/projects/spring-security>. [Kasutatud 23 02 2022].
- [26] „Dropwizard is a Java framework for developing ops-friendly, high-performance, RESTful web services“, Dropwizard team, [Võrgumaterjal]. Available: <https://www.dropwizard.io/en/latest/>. [Kasutatud 19 04 2022].
- [27] „grails“, Grails Foundation, [Võrgumaterjal]. Available: <https://grails.org>. [Kasutatud 19 04 2022].
- [28] „Hibernate ORM Your relational data. Objectively.“, Hibernate, [Võrgumaterjal]. Available: <https://hibernate.org/orm/>. [Kasutatud 19 04 2022].

- [29] „From a birds eye,“ The Apache Software Foundation, [Võrgumaterjal]. Available: <https://struts.apache.org/birdseye.html>. [Kasutatud 19 04 2022].
- [30] „Pros and Cons of Using Spring Boot,“ Scand, 26 06 2020. [Võrgumaterjal]. Available: <https://scand.com/company/blog/pros-and-cons-of-using-spring-boot/>. [Kasutatud 14 03 2022].
- [31] „spring-projects / spring-boot,“ [Võrgumaterjal]. Available: <https://github.com/spring-projects/spring-boot/releases>. [Kasutatud 14 03 2022].
- [32] A. Zhitnitsky, „Java Bootstrap: Dropwizard vs. Spring Boot,“ OverOps, 02 03 2015. [Võrgumaterjal]. Available: <https://www.overops.com/blog/java-bootstrap-dropwizard-vs-spring-boot/>. [Kasutatud 14 03 2022].
- [33] „dropwizard / dropwizard,“ [Võrgumaterjal]. Available: <https://github.com/dropwizard/dropwizard/releases>. [Kasutatud 14 03 2022].
- [34] „Grails,“ G2, [Võrgumaterjal]. Available: <https://www.g2.com/products/grails/reviews>. [Kasutatud 14 03 2022].
- [35] „grails / grails-core,“ [Võrgumaterjal]. Available: <https://github.com/grails/grails-core/releases>. [Kasutatud 14 03 2022].
- [36] „Hibernate ORM Contribute,“ Hibernate, [Võrgumaterjal]. Available: <https://hibernate.org/orm/contribute/>. [Kasutatud 14 03 2022].
- [37] „Hibernate ORM Releases,“ Hibernate, [Võrgumaterjal]. Available: <https://hibernate.org/orm/releases/>. [Kasutatud 14 03 2022].
- [38] „You at Apache Struts,“ The Apache Software Foundation, [Võrgumaterjal]. Available: <https://struts.apache.org/youatstruts.html>. [Kasutatud 14 03 2022].
- [39] <https://struts.apache.org/releases.html>, „Apache Struts Releases,“ The Apache Software Foundation, [Võrgumaterjal]. Available: <https://struts.apache.org/releases.html>. [Kasutatud 14 03 2022].
- [40] F. Lardinois ja . I. Lunden, „Microsoft has acquired GitHub for \$7.5B in stock,“ Yahoo, 04 06 2018. [Võrgumaterjal]. Available: <https://techcrunch.com/2018/06/04/microsoft-has-acquired-github-for-7-5b-in-microsoft-stock/?guccounter=1>. [Kasutatud 14 03 2022].
- [41] „Versioonihalduskeskkonnad GitHubi, GitLabi ja Bitbucketi näitel,“ ICO wiki, 04 05 2020. [Võrgumaterjal]. Available: https://wiki.itcollege.ee/index.php/Versioonihalduskeskkonnad_GitHubi,_GitLabi_ja_Bitbucketi_näitel. [Kasutatud 14 03 2022].
- [42] N. Friedman, „GitHub is now free for teams,“ GitHub, 14 04 2020. [Võrgumaterjal]. Available: <https://github.blog/2020-04-14-github-is-now-free-for-teams/>. [Kasutatud 14 03 2022].
- [43] „GitLab Projects,“ JetBrains, [Võrgumaterjal]. Available: <https://plugins.jetbrains.com/plugin/7975-gitlab-projects>. [Kasutatud 14 03 2022].

- [44] „Bitbucket Pricing & Licensing,“ Atlassian, 2022. [Võrgumaterjal]. Available: <https://www.atlassian.com/licensing/bitbucket#server-licensing>. [Kasutatud 13 04 2022].
- [45] „Free unlimited private repositories for student developers and teachers,“ Atlassian, 2022. [Võrgumaterjal]. Available: <https://bitbucket.org/product/education>. [Kasutatud 13 04 2022].
- [46] „IntelliJ IDEA Ultimate vs IntelliJ IDEA Community,“ JetBrains, 2022. [Võrgumaterjal]. Available: <https://www.jetbrains.com/products/compare/?product=idea&product=idea-ce>. [Kasutatud 18 04 2022].
- [47] „Difference between IntelliJ Idea and Eclipse,“ JavaTpoint, [Võrgumaterjal]. Available: <https://www.javatpoint.com/intellij-vs-eclipse>. [Kasutatud 14 03 2022].
- [48] „What is a Relational Database?,“ Amazon Web Services, 2022. [Võrgumaterjal]. Available: <https://aws.amazon.com/relational-database/>. [Kasutatud 14 03 2022].
- [49] „MySQL for Beginners,“ Oracle, 2022. [Võrgumaterjal]. Available: <https://www.mysql.com/news-and-events/web-seminars/mysql-for-beginners-202204/>. [Kasutatud 18 04 2022].
- [50] J. Hurley, „The Pros and Cons of MySQL,“ SmartFile, 03 06 2021. [Võrgumaterjal]. Available: <https://www.smartfile.com/blog/the-pros-and-cons-of-mysql/>. [Kasutatud 18 04 2022].
- [51] „MariaDB Server: The open source relational database,“ MariaDB Foundation, [Võrgumaterjal]. Available: <https://mariadb.org>. [Kasutatud 18 04 2022].
- [52] „Editions and supported features of SQL Server 2019 (15.x),“ Microsoft, 2022. [Võrgumaterjal]. Available: <https://docs.microsoft.com/en-us/sql/sql-server/editions-and-components-of-sql-server-2019?view=sql-server-ver15>. [Kasutatud 18 04 2022].
- [53] „SQLite vs MySQL - Comparing 2 Popular Databases,“ KeyCDN, 2022. [Võrgumaterjal]. Available: <https://www.keycdn.com/support/sqlite-vs-mysql>. [Kasutatud 14 03 2022].
- [54] R. Peterson, „What is PostgreSQL? Introduction, Advantages & Disadvantages,“ Guru99, 16 04 2022. [Võrgumaterjal]. Available: <https://www.guru99.com/introduction-postgresql.html>. [Kasutatud 18 04 2022].
- [55] „Version 2.6.1 20th Anniversary Release,“ The HSQL Development Group, 10 2021. [Võrgumaterjal]. Available: <http://hsqldb.org>. [Kasutatud 10 03 2022].
- [56] J. Abhisek, „An Introduction to Spring Boot,“ A Developer Diary , 27 03 2016. [Võrgumaterjal]. Available: <http://www.adeveloperdiary.com/java/spring-boot/an-introduction-to-spring-boot/>. [Kasutatud 14 03 2022].
- [57] „spring initializr,“ [Võrgumaterjal]. Available: <https://start.spring.io>. [Kasutatud 15 03 2022].
- [58] „Spring Boot – Working with Embedded Databases,“ TutorialCup, 2022. [Võrgumaterjal]. Available: <https://www.tutorialcup.com/spring-boot/spring-boot-working-with-embedded-databases.htm>. [Kasutatud 12 04 2022].

- [59] „Spring Data JPA,“ VMware, 2022. [Võrgumaterjal]. Available: <https://spring.io/projects/spring-data-jpa>. [Kasutatud 12 04 2022].
- [60] T. Janssen, „Getting Started with Flyway and Version-Based Database Migration,“ Thorben Janssen, [Võrgumaterjal]. Available: <https://thorben-janssen.com/flyway-getting-started/>. [Kasutatud 12 04 2022].
- [61] „Lombok features,“ The Project Lombok, [Võrgumaterjal]. Available: <https://projectlombok.org/features/all>. [Kasutatud 19 04 2022].
- [62] „The Java EE 6 Tutorial,“ Oracle, 2013. [Võrgumaterjal]. Available: <https://docs.oracle.com/javaee/6/tutorial/doc/bnacj.html#bnadb>. [Kasutatud 19 04 2022].
- [63] „The Java EE 6 Tutorial,“ Oracle, 2013. [Võrgumaterjal]. Available: <https://docs.oracle.com/javaee/6/tutorial/doc/bnacj.html#bnadb>. [Kasutatud 19 04 2022].

Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks¹

Mina, Maarja Helena Elisabeth Hoop

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose „TALTECHI Algoritmilise mõtlemise VEEBIMÄNGU serverrakenduse arendus“, mille juhendaja on Meelis Antoi ja kaasjuhendaja Kerman Saapar
 - 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.

25.04.2022

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

Lisa 2 – Funktsionaalsed nõuded

Eepiku ID	<i><Mingit tüüpi kasutajana></i>	tahan <i><sooritada mingit ülesannet></i>	et <i><saavutada mingi eesmärk></i>
1	Õpilasena	tahan mängida mängu,	et arendada oma algoritmilise mõtlemise oskust
2	Õpilasena	tahan saada mängult interaktiivset tagasisidet,	et mul oleks huvitav
3	Kasutajana	tahan ise mängu teha,	et oma algoritmiline mõtlemine veel rohkem proovile panna
4	Administraatorina	tahan hallata kõiki mängu,	et hoida keskkond töökorras
5	Administraatorina	tahan hallata kasutajaid,	et süsteemi kasutajate õigused vastaks vajadusele

Eepiku ID	Kasutajalo olulisus	Kasutajalo ID	<i><Mingit tüüpi kasutajana></i>	tahan <i><sooritada mingit ülesannet></i>	et <i><saavutada mingi eesmärk></i>
1	S	1	Õpilasena	tahan valida mängu,	et saaksin mängida enda jaoks huvitavat mängu
1	M	2	Õpilasena	tahan analüüsida ülesandepüstitust,	et ülesandest aru saada
1	L	3	Õpilasena	tahan lohistada kaste ja sõnu,	et luua oma programm
1	M	4	Õpilasena	tahan ebaõnnestumise l mängu uuesti proovida,	et parandada oma lahendust
1	M	5	Õpilasena	tahan korduval ebaõnnestumise l kergemale tasemele liikuda,	et saaksin lahendada oma tasemele vastavat ülesannet
1	S	6	Õpilasena	tahan õnnestumise korral	et saaksin lahendada oma

				raskemale tasemele liikuda,	tasemele vastavat ülesannet
1	L	7	Õpilasena	tahan mängida mängu samal ajal teiste mängijatega	kõik mängijad saaksid omale sobival ajal mängida
2	L	7	Õpilasena	tahan käivitada oma programmi,	et näha oma lahenduse toimimist
2	S	8	Õpilasena	tahan näha oma programmi põhjal loodud animatsiooni,	et saada interaktiivset tagasisidet lahenduse õigsusest
3		9	Kasutajana	tahan sisse logida,	et pääseda ligi kasutajavaatele
3		10	Kasutajana	tahan lisada mängu,	et arendada oma loovust algoritmilist mõtlemist
3		11	Kasutajana	tahan muuta enda loodud mängu,	et neid paremaks muuta
3		12	Kasutajana	tahan kustutada enda loodud mängu,	et eemaldada see süsteemist
3		13	Kasutajana	tahan valida oma loodud mängu privaatsust,	et vastavalt soovile jagada seda teistega või mängida ainult ise
3		14	Kasutajana	tahan välja logida,	et lahkuda kasutajavaatest
4		15	Administraatorina	tahan laiendatud õigustega sisse logida,	et pääseda ligi administraatorivaatele
4		16	Administraatorina	tahan muuta mängu,	et kindlustada süsteemis mängude korrektsus
4		17	Administraatorina	tahan kustutada mängu,	et eemaldada sobimatud mängud
4		18	Administraatorina	tahan kasutajate loodud mängu kinnitada,	et sobivaid mängu kõigiga jagada
4		19	Administraatorina	tahan välja logida laiendatud	et takistada teistele juurdepääsu administraatorivaatele

				õigustega kasutajast,	
5		19	Administraatori na	tahan anda kasutajale laiendatud õigusi,	et võimaldada valitud kasutajale administraatori funktsioonide kasutamist
5		20	Administraatori na	tahan kasutajalt laiendatud õigusi ära võtta,	et eemaldada valitud kasutajalt administraatori funktsioonide kasutamise võimalus

S - vähem oluline

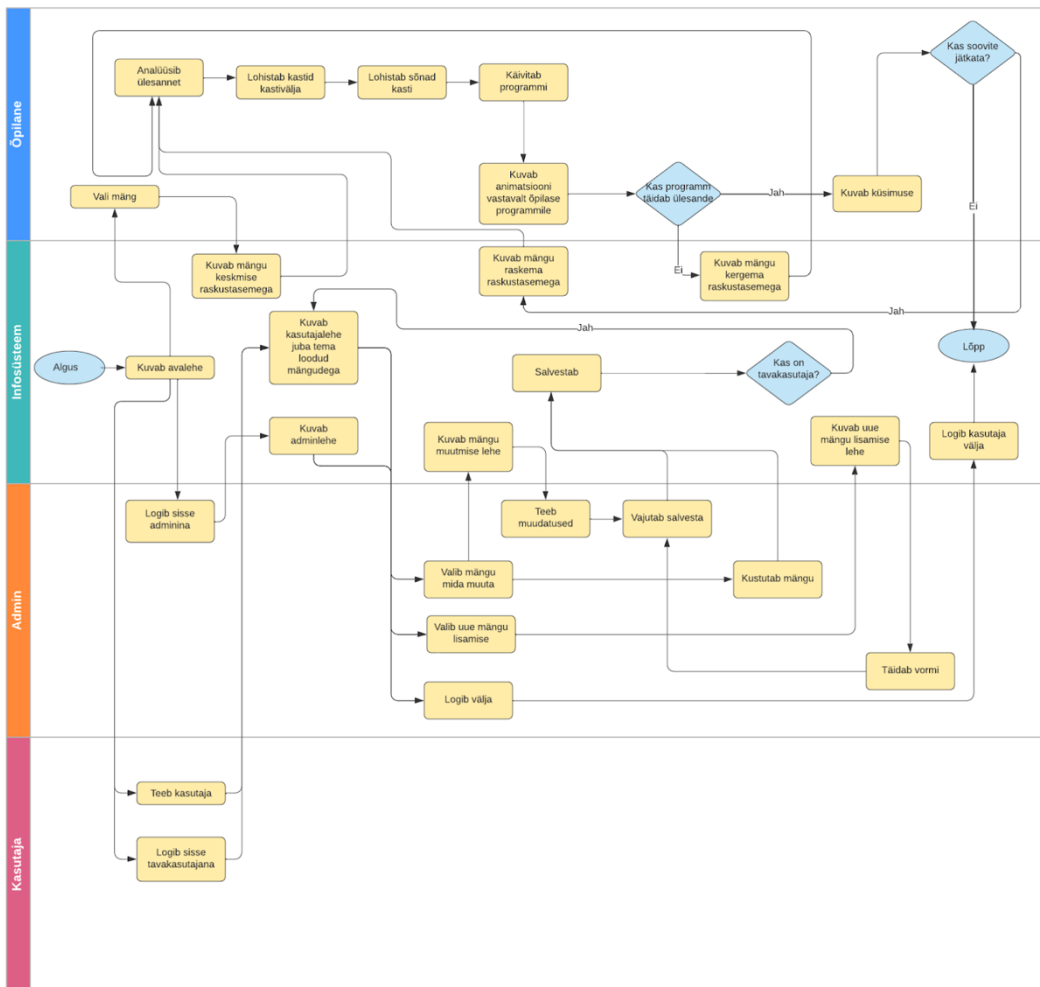
M - keskmiselt oluline

L - väga oluline

Lisa 3 – Infosüsteemi skeem

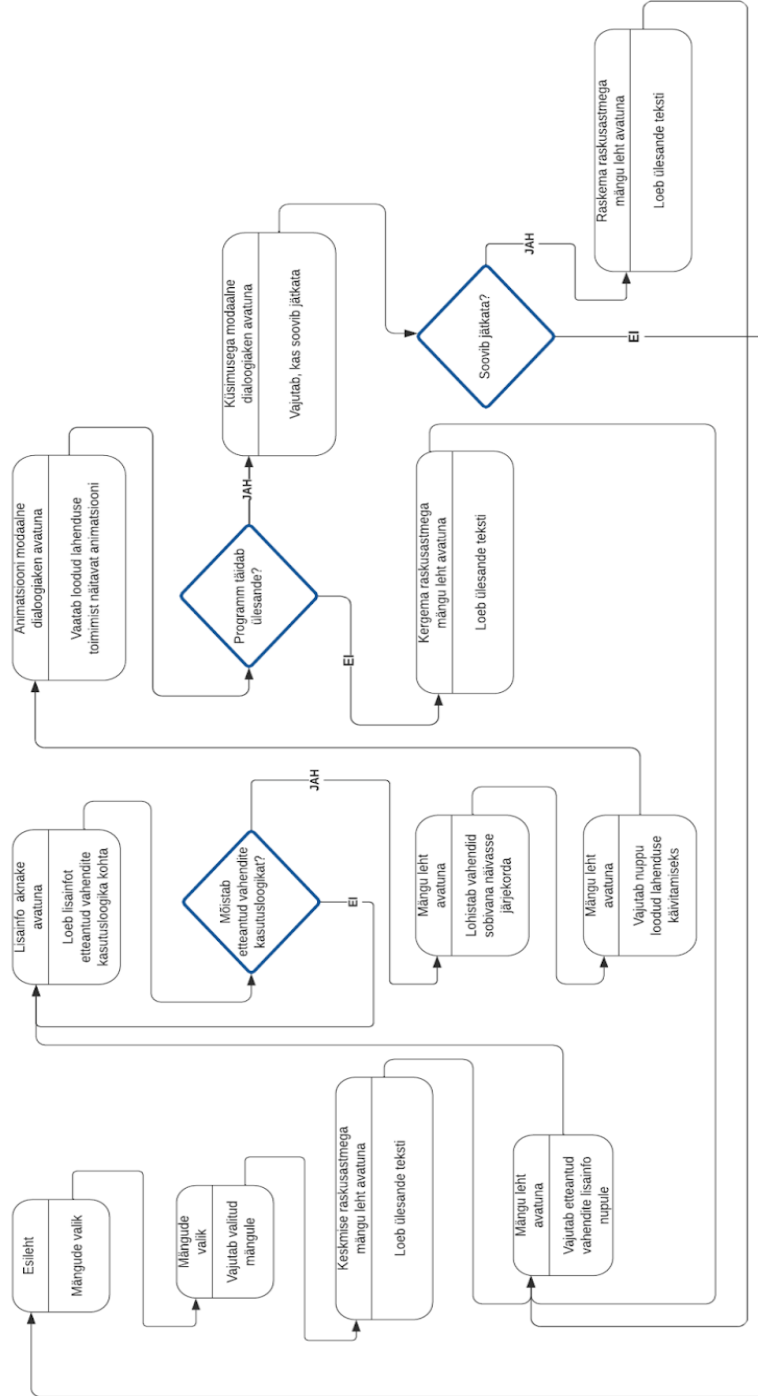
Esimene lõputöö skeem

Maaaja Hoop ja Tiinu Mälv

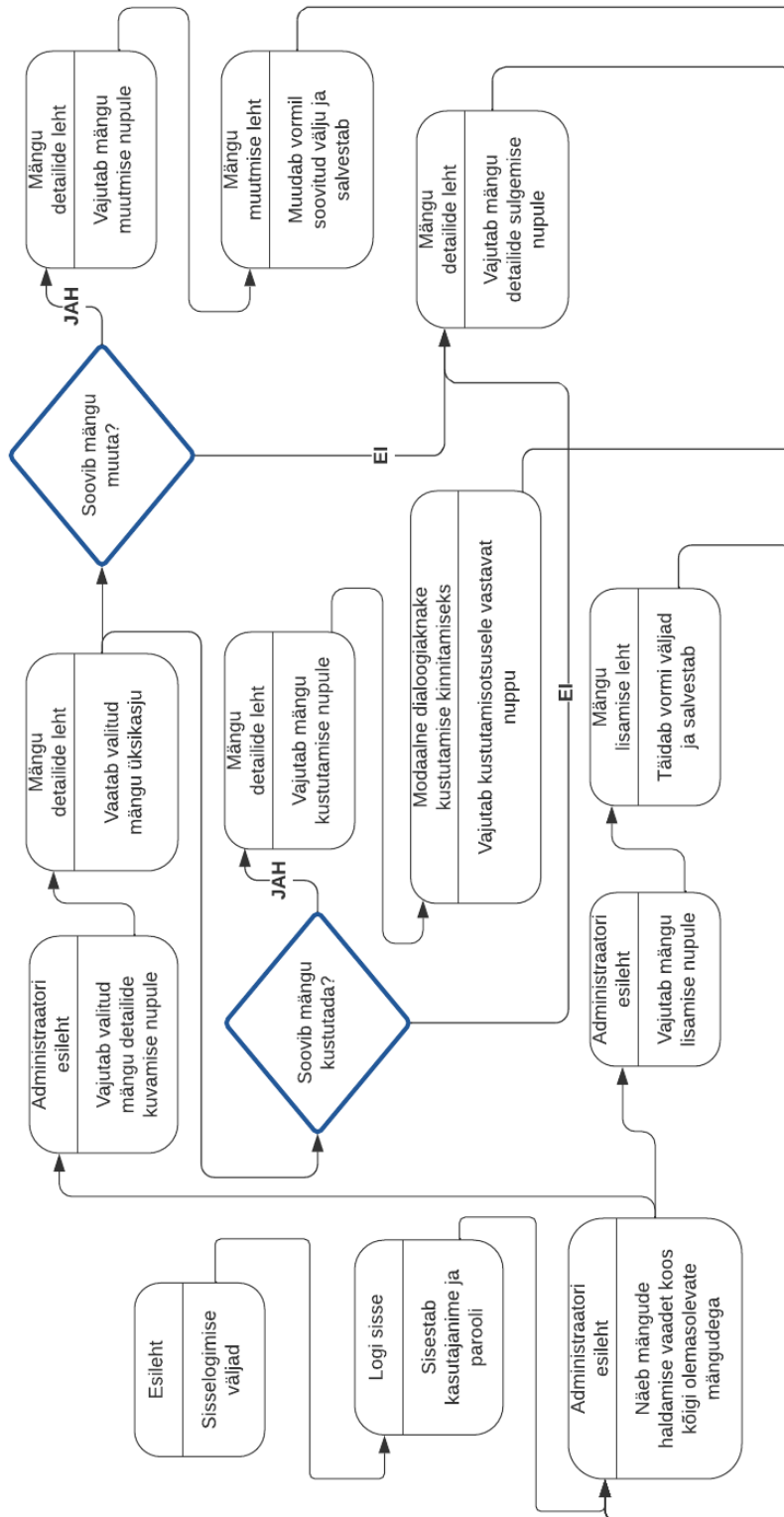


Lisa 4 – Kasutajakogemuse disaini joonised

Tavakasutaja ülesande lahendamise kasutajakogemuse diagramm:



Administraatori kasutajakogemuse diagramm:



Tavakasutaja ülesande lahendamise vaate prototüüp enne lahendamist:

Algoritmilise mõtlemise õpituba / Kosmosroboti programmeerimine

← Kosmosroboti programmeerimine

Ülesanne

Aita kosmonaudiõpilasel eksamilt kõrvale hiilida. Kosmonaut peab kooli lõpetamiseks läbima Kuu peal toimetamise eksami. Tal tuli aga idee saata kuurobot enda asemel ülesandeid tegema. Ta teab, et ta saab koolist vajaliku roboti laenutada, kuid puudu on programm, mis paneks roboti eksami ülesandeid sooritama. Pooletise tunni pärast stardib eksamirakett Kuule, nii et tal on roboti programmeerimiseks jäänud napilt 45minutit. Aita kosmonaudil "spikerdada"!

1 — 2

Vahendid lahendamiseks

KUNI	ON	puud
IGA	KOHTA	saab kaevata
LIIGU	MEETRIT	edasi
	+	istutamata
	=	tõsi
	=	2
	=	ei saa kaevata
	=	istutatud
	=	puu
	=	0
KAEVA AUK	ASETA PUU AUKU	kaugus laevast
KASTA	AJA AUK KINNI	tagasi
		vale

Vahendite kasutamine

Keskmise raskustasemega mängus läheb lahendamiseks vaja kõiki etteantud vahendeid. Küll aga ei ole teada, mitu korda igat kasti, numbrit või sõna tuleb kasutada.

Lahendusprogramm

Käivita >

Tavakasutaja ülesande lahendamise vaate prototüüp koos lahendusega:

Algoritmilise mõtlemise õpituba / Kosmosroboti programmeerimine

← Kosmosroboti programmeerimine

Ülesanne

Eksami sooritamiseks tuleb kosmonaudiõpilasel luua Kuu esimene puuviljaaed. Ta peab maha istutama kärutäie kooli poolt kaasa antud ploompuid, kusjuures pole teada, kui palju neid täpselt on. Iga puu jaoks tuleb kaevata auk, sinna panna puu, auk uuesti kinni ajada ja seejärel kasta. Puude istutusvahe peaks olema 2 meetrit. Kui kuskil pole liiga kõva pinnase tõttu võimalik kaevata, tuleb veel 2 meetrit edasi liikuda ja seal uuesti proovida. Kui kärü on istikutest tühjaks saanud, tuleb eksami sooritajal tagasi kosmoselaeva liikuda.

1 — 2

Vahendid lahendamiseks

KUNI	ON	puud
IGA	KOHTA	saab kaevata
LIIGU	MEETRIT	edasi
	=	istutamata
	=	tõsi
	=	2
	=	ei saa kaevata
	=	istutatud
	=	puu
	=	0
KAEVA AUK	ASETA PUU AUKU	kaugus laevast
KASTA	AJA AUK KINNI	tagasi
		vale

Lahendusprogramm

kaugus laevast = 0 MEETRIT

KUNI puud ON istutamata

IGA puu KOHTA

LIIGU 2 MEETRIT edasi

kaugus laevast = kaugus laevast + 2

KUNI saab kaevata ON vale

LIIGU 2 MEETRIT edasi

kaugus laevast = kaugus laevast + 2

KAEVA AUK

ASETA PUU AUKU

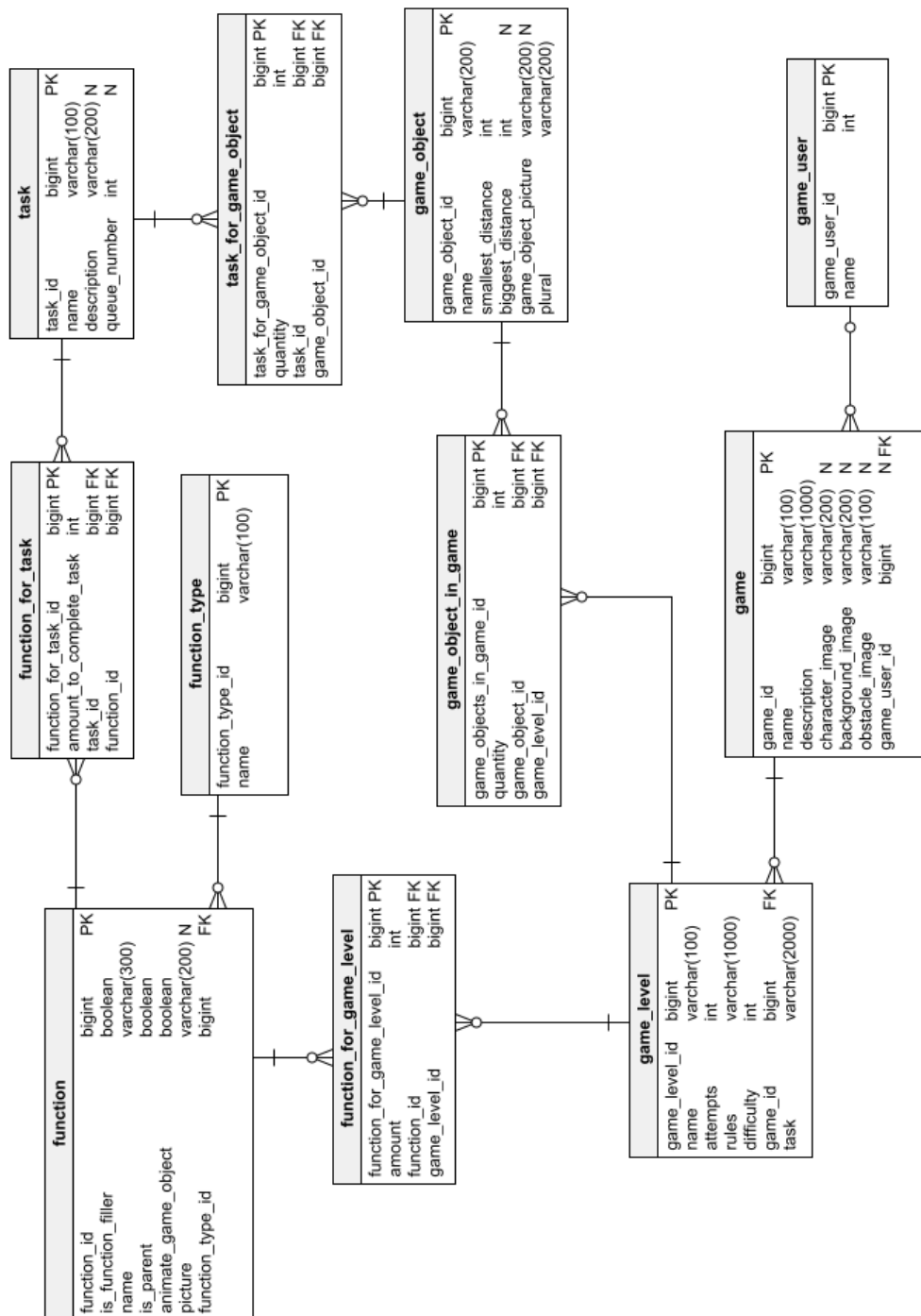
AJA AUK KINNI

KASTA

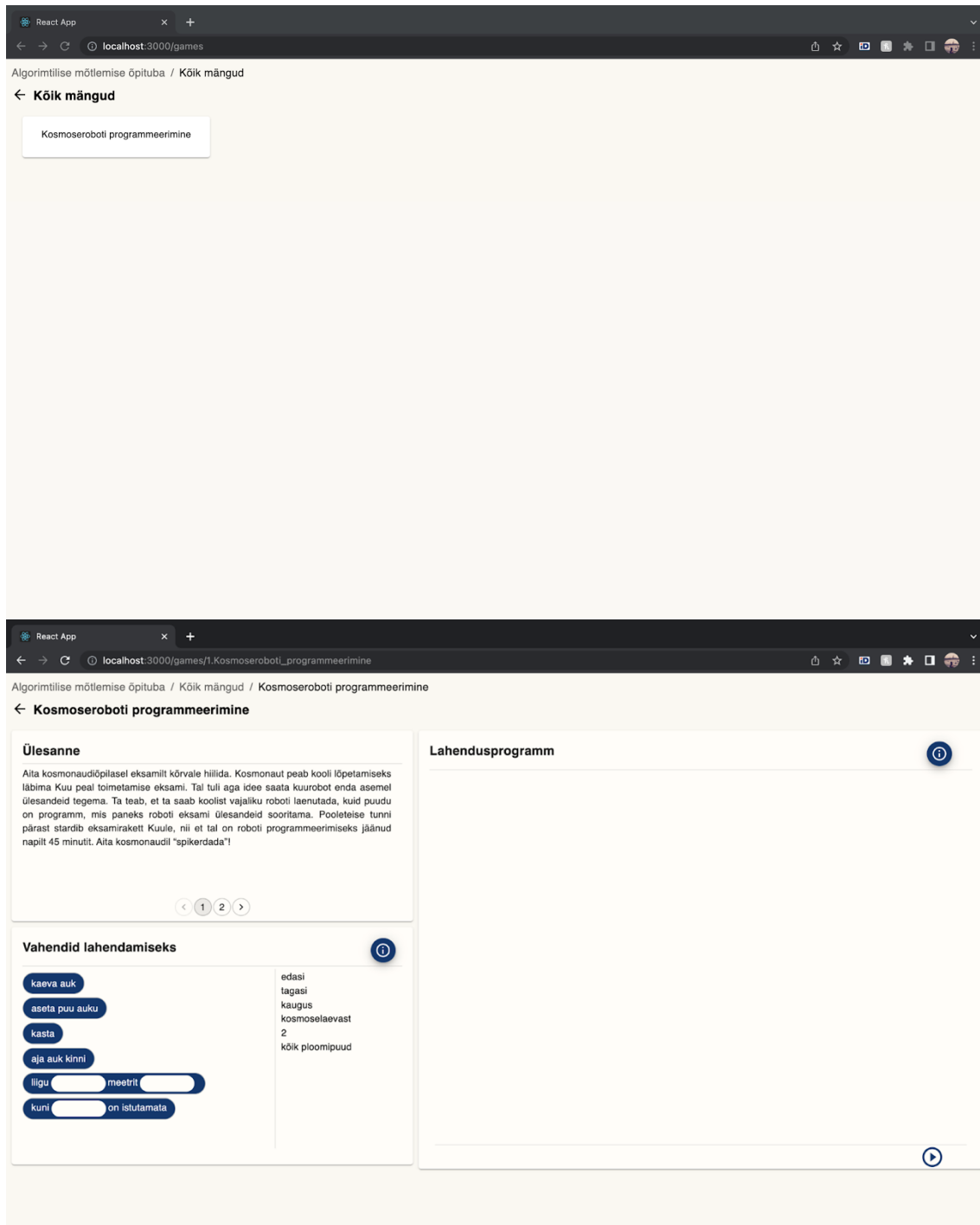
LIIGU kaugus laevast MEETRIT tagasi

Käivita >

Lisa 5 – Andmebaasi täielik olemi-suhte diagramm



Lisa 6 – Mängu kosmoseroboti programmeerimine vaated



React App | localhost:3000/games/1.Kosmoseroboti_programmeerimine

Algoritmilise mõtlemise õpituba / Kõik mängud / Kosmoseroboti programmeerimine

Kosmoseroboti programmeerimine

Ülesanne

Aita kosmonautidõpilasel eksamilt kõrvale hõllida. Kosmonaut peab kooli lõpetamiseks läbima Kuu peal toimetamise eksami. Tal tuli aga idee saata kuurobot enda asemel ülesandele tegema. Ta teab, et ta saab koolist vajaliku roboti laenutada, kuid puudu on programm, mis paneks roboti eksami ülesandeid sooritama. Pooleteise tunni pärast startib eksamirakett Kuule, nii et tal on roboti programmeerimiseks jäänud napilt 45 minutit. Aita kosmonaudil "spikerdada"!

Lahendusprogramm

```

kuni kõik ploomipuud on istutamata
  kaeva auk
  aseta puu auku
  aja auk kinni
  kasta
  liigu 2 meetrit edasi
liigu kaugus kosmoselaevast meetrit tagasi

```

Vahendid lahendamiseks

kaeva auk	edasi
aseta puu auku	tagasi
kasta	kaugus
aja auk kinni	kosmoselaevast
liigu _____ meetrit _____	2
kuni _____ on istutamata	kõik ploomipuud

React App | localhost:3000/games/1.Kosmoseroboti_programmeerimine

Algoritmilise mõtlemise õpituba / Kõik mängud / Kosmoseroboti programmeerimine

Lahenduse animatsioon

Hästi tehtud! Said ülesandega hakkama.

Jätka mängu raskemal ta...

Välju mängust

Lisa 7 – Link rakenduse versioonihaldusele

Teenus - <https://gitlab.cs.ttu.ee/mahoop/algorithmicgameback>

Klientrakendus - https://gitlab.cs.ttu.ee/trmalv/algorithmic_thinking_workshop_client

Lisa 8 – Link rakenduse veebilehele

URL – talgotech.itcollege.ee

Lisa 9 – tALGOtech kasutusjuhend töötoa läbiviijale

1. Lahenduse koostamine

tALGOtech rakenduse mängudes esitatud ülesannete lahendamiseks tuleb kasutajal etteantud vahenditest moodustada sobiv algoritm. Vahendid jagunevad kaheks: kastid ja lüngatäited. Algoritmi koostamiseks tuleb kastid õigesti järjestada ning osades kastides olevad lüngad vajalike sõnade või numbritega täita.

Rakenduse kasutajad leiavad lühiversiooni järgnevatest juhistest, vajutades ülesande lahendamise vaates olevatele infonuppudele (üks on etteantud vahendite ning teine lahendamisala juures).

Kastide järjestamine

- Kastides kirjeldatud tegevuste järjestamiseks tuleb need **vahendite hulgast lahendusala**le nihutada.
- Mõned kastid seisnevad teatud tingimuse seadmises (näiteks "Kui saab istutada") või korduste tekitamises (näiteks "Kuni puud on istutamata"). Nende alla tekivad pärast lahendusala nihutamist **helesinised lisaalad**, kuhu on võimalik paigutada **nende kastide mõjualasse kuuluvad tegevused**.
- Lahenduses olevaid kaste on võimalik omavahel **ümber järjestada**, kuid seda saab teha vaid n-ö sama kihi siseselt. See tähendab, et omavahel on võimalik ümber järjestada kas lahenduse põhialas olevaid kaste või samas lisaalas olevaid kaste.
- Kaste on võimalik lahendusest **eemaldada**, vajutades kasti kõrval olevat ristikest.
- Selleks, et kasti **ühest kihist teise liigutada**, tuleb kast lahendusest kustutada ning vahendite hulgast uuesti soovitud kohale nihutada.

Lüngatäidete kasutamine

- Lüngatäite **soovitud lünka lisamiseks** tuleb see vahendite hulgast juba lahenduses oleva kasti lünga kohale nihutada.
- Lahenduses olevaid lüngatäiteid on võimalik **ühest lüngast teise** nihutada.
- Lahenduses olevate lüngatäidete **eemaldamiseks** tuleb vajutada lüngatäite kõrval olevat ristikest.

Tasemete erisused

- **Kergel tasemel** on etteantud **täpselt vajaminev hulk vahendeid** ehk iga kasti ja lüngatäidet läheb vaja ainult üks kord. Selle tagamiseks muutuvad juba lahendusse nihutatud vahendid deaktiveerituks. Kui kasutaja mõne vahendi algoritmist kustutab, muutub see taas aktiivseks.
- **Keskmisel ja raskel tasemel pole teada**, kui **mitu korda iga vahendit** õige lahenduse koostamiseks **vaja läheb**. Seega jäävad vahendid pärast kasutamist aktiivseks, kuna sama vahendit võib algoritmi loomiseks vaja minna mitmel korral.

2. Algoritmi käivitamine

Kasutaja saab koostatud algoritmi toimimist näha, **vajutades** lahendusala alumises osas olevale **play-nupule**. Seeläbi **vähendatakse kasutaja allesjäänud katsete arvu** ning avatakse animatsiooni näitav lisaaknake. **Animatsioon** näitab kõiki tegevusi, mida kasutaja algoritm on juhendanud mängu tegelast tegema. Selle lõppedes näidatakse kasutajale, **kas algoritm lahendas ülesande või mitte** ning annab ebaõnnestumise korral ka vihje, mis valesti läks. **Kui algoritm täitis ülesande tingimusi**, saab kasutaja soovi korral **liikuda raskemale tasemele**. Vastasel juhul tuleb kasutajal kas **uuesti proovida** või kui tal praegusel tasemel katsed otsa on saanud, peab ta **kergemale tasemele liikuma**.

3. Mäng “Kosmoseroboti arendamine”

Mängus on tegelaseks kosmonaudist õpilane, kes peab Kuu peal eksami läbima. Eksami sisuks on ploompuid istutamine, et luua Kuule puuviljaaeda. Mängu läbimiseks peab mängija kirjutama programmi, mille abil oskaks kosmonaudi robot vastavalt ülesande nõuetele ploompuid istutada.

Mängu üldkirjeldus: “Aita kosmonaudiõpilasel eksamilt kõrvale hiilida. Kosmonaut peab kooli lõpetamiseks läbima Kuu peal toimetamise eksami. Tal tuli aga idee saata kuurobot enda asemel ülesandeid tegema. Ta teab, et ta saab koolist vajaliku roboti laenutada, kuid puudu on programm, mis paneks roboti eksamiülesandeid sooritama. Pooleteise tunni pärast stardib eksamirakett Kuule, nii et tal on roboti programmeerimiseks jäänud napilt 45 minutit. Aita kosmonaudil “spikerdada!”

Mängu põhiobjekt on ploompuu, mida võib istutada minimaalselt iga 2m tagant. Mängu algoritmi koostamiseks on õpilasel võimalik kasutada kaste ja lüngatäiteid. Ploompuid istutamise erifunktsioonikastid on “augu kaevamine”, “puu auku panemine”, “augu kinniajamine” ja “kastmine”. Kõiki tegevusi tuleb ühe ploompuid istutamiseks teha üks kord. Lisaks erikastidele on mängus tüüpilised kastid liikumiseks, tsükli loomiseks ja vajalike tingimuste kontrolliks.

Võimalikud kastid:

Nimi	Mida see teeb?	Muutujad
Liigu _ meetrit _	Liikumine	Teepikkus, suund
Kuni _ on istutamata	Tsükkel. Teeb tsükli sisse pandud tegevusi nii mitu korda kui palju on muutujana antud mänguobjekte	Mänguobjektid, mille alusel tsükkel tehakse
Erifunktsioonid: kaeva auku, aseta puu auku, aja auk kinni, kasta	Spetsiaalne juhised ploompuid istutamise tegevuse täitmiseks	-
Kui saab _	Kontrollib kas mingit tegevust saab teha	Tegevus, mida kontrollitakse

Võimalikud lüngatäited:

Nimi	Mida see teeb?
Edasi	Näitab liikumise suunda edasi
Tagasi	Näitab liikumise suunda tagasi
Number (nt 2)	Vahemaa, mille võrra liigutakse
Kaugus kosmoselaevast	Kaugus millestki, näiteks kaugus algusest
Kõik ploompuid	Mänguobjektid, mille alusel tsükkel tehakse
Istutada	“Kui saab _ “ lüngatäide. Kontrollib, kas selles kohas mänguala saab teha vajalikke toimingud

4. Mäng "Kosmoseroboti arendamine" lahendused

Lihtsa taseme lahendus:

The image shows a sequence of Scratch code blocks for a simple level solution. The blocks are arranged in a vertical stack and are all marked with a red 'x' icon, indicating they are not yet executed. The blocks are:

- kaeva auk
- asetapuuka
- ajaukukinni
- kasta
- liigu 2 meetrit edasi
- kaeva auk
- asetapuuka
- ajaukukinni
- kasta
- liigu 2 meetrit edasi
- kaeva auk
- asetapuuka
- ajaukukinni
- kasta
- liigu kauguskosmoselaevast meetrit tagasi

Keskmise taseme lahendus:

kuni kõik ploomipuud on istutamata

kaeva auk

asete puu auku

aja auk kinni

kasta

liigu 2 meetrit edasi

liigu kaugus kosmoselaevast meetrit tagasi

Raske taseme lahendus:

kuni kõik ploomipuud on istutamata

kui saab istutada

kaeva auk

asete puu auku

aja auk kinni

kasta

liigu 2 meetrit edasi

liigu kaugus kosmoselaevast meetrit tagasi