

Tartu Ülikool  
Loodus- ja täppisteaduste valdkond  
Arvutiteaduse instituut  
Informaatika õppekava

Mark Muhhin

# Ülesannete komplekt ainele “Teeme ise arvutimänge - algus”

Bakalaureusetöö (9 EAP)

Juhendaja: Ljubov Jaanuska, MSc.

Tartu 2018

## **Ülesannete komplekt ainele „Teeme ise arvutimänge - algus”**

### **Lühikokkuvõte:**

„Teeme ise arvutimänge – algus“ on Tartu Ülikooli poolt pakutud veebipõhine programmeerimise kursus. See kursus on põhiliselt mõeldud gümnaasiumi tasemel õpilastele, kes tahavad õppida Pythoni programmeerimise keelt ning selle abil koostada mängu.

Kursuse viimane suur muudatus ülesannetes ja õppematerjalis oli aastal 2012, mille tõttu on kursuse suurimaks probleemiks vananemine. Käesoleva bakalaureusetöö kirjutamise käigus koostatakse uus ülesannete komplekt, mis oleks huvitav, kuid samas kontrolliks tõhusalt õpikus olevat materjali teadmist.

Töö kirjutamise käigus on ülesanded lisatud kursusele, et saaks nad testitud uue õppijate rühma peal. Kursuse jooksul palutakse täita tagasiside küsitlusi, mida antud töös analüüsitakse ning tehakse järeldusi uute ülesannete sobilikkusest ja kvaliteedist.

### **Võtmesõnad:**

E-kursus, Python, õppematerjalid, CERCS: S281 – Arvuti õpiprogrammide kasutamise meetodika ja pedagoogika

## **A set of exercises for the subject „Let’s make computer games – beginning”**

### **Abstract:**

„Let’s make computer games – beginning“ is an online programming course offered by University of Tartu. This course is mainly meant for high school students who want to learn the Python programming language and use it to create games.

The last time the courses’ exercises and the studying material had a major update was in the year 2012, which makes aging one of the biggest problems of the course. In the process of writing the following thesis, a new set of exercises was created that would be interesting but at the same time thoroughly test the knowledge of the studying material.

While writing the thesis, the exercises were added to the course with the intention of testing them on a new group of students. During this course, the students were asked to fill in a weekly questionnaire. The information gathered from the questionnaires are then used in this thesis to analyse and make deductions about the suitability and quality of the exercises.

### **Keywords:**

E-course, Python, teaching materials CERCS: S281 – Computer-assisted education

## Sisukord

1. Sissejuhatus.....	4
2. E-kursus “Teeme ise arvutimänge - algus” .....	5
2.1 E-kursuse sihtrühm .....	5
2.2 E-kursuse probleemid.....	6
2.3 Sarnased kursused.....	6
3. Kursuse muudatused.....	8
3.1 Kodused ülesanded .....	9
3.1.1 Esimene nädal .....	9
3.1.2 Teine nädal.....	15
3.1.3 Kolmas nädal.....	20
3.1.4 Neljas nädal .....	25
3.1.5 Viies nädal .....	31
4. Tulemused .....	34
4.1 Õpilaste tagasiside.....	34
4.2 Järeldused.....	43
5. Kokkuvõte .....	44

# 1. Sissejuhatus

Kiire tehnoloogiline areng kasvatab aastast aastasse aina rohkem huvi programmeerimise vastu. Informaatika on saanud üheks kõige populaarsematest kõrgkooli erialadest Eestis [1][2][3].

Aastast 2012 on Tartu Ülikool pakkunud kõigile programmeerimishuviliste gümnaasiumiõpilastele võtta osa veebipõhisest kursusest nimetusega „Teeme ise arvutimänge - algus“ [4], mille autor on Tiina Kull. See kursus keskendub Pythoni programmeerimiskeele põhiliste baasteadmiste õpetamisele ja soetatud teadmiste abil lihtsamate mängude tegemisele.

Kursuse põhiline probleem on see, et ülesanded on aastast 2012 muutmata ning aegunud. Seetõttu on soodustatud plagiaat, mis on tingitud eelnevate aastate õpilaste kodutööde Internetist kättesaadavusest. Lisaks sellele on kursuse õppematerjalid nüüdseks uuendatud [5] Tartu Ülikooli magistriõppe lõpetaja Jaan Janno poolt, mistõttu ka ülesannete komplekt vajab uuendust vastavalt muudetud õppematerjalile.

Käesoleva lõputöö eesmärgiks on luua uus ülesannete komplekt kodutööde jaoks, mis motiveeriks õpilasi ning ärataks suuremat huvi programmeerimise vastu, kuid sealjuures vastaks ka uuendatud õppematerjalidele.

Bakalaureusetöö jaguneb nelja peatükki, kus teises peatükis räägitakse lähemalt vaadeldavast kursusest, kolmandas peatükis analüüsitakse kõiki muutusi, mis olid tehtud antud kursusele bakalaureusetöö käigus ning neljandas peatükis antakse ülevaade õpilaste tagasisidest ning sellega kaasnevatest järeldustest.

## **2. E-kursus “Teeme ise arvutimänge - algus”**

„Teeme ise arvutimänge - algus“ on Tartu Ülikooli arvutiteaduse instituudi pakutav e-kursus, mis oli algselt koostatud Tiina Kulli poolt aastal 2012. Kursuse õppetöö toimub Moodle'i õpikeskkonnas, kus õpilased loevad iseseisvalt kursuse õpiku peatükke ning lahendavad vastavalt peatüki teemadele iseseisvaid ülesandeid kui ka teste, mis kontrollivad teoreetilisi teadmisi. Tegemist on programmeerimise kursusega, mis on mõeldud gümnaasiumiõpilastele. E-kursusel antakse ülevaade olulistest programmeerimisvõtetest, mida algaja peaks oskama selleks, et koostada konsooli või graafilisi mängu Pythoni keeles. Kuigi kogu õppetöö on veebipõhine, saavad õppijad tagasisidet iga tehtud ülesande kohta ning vajadusel saab küsida abi kursuse juhendajatelt probleemide tekkimise korral.

E-kursus kestab kaheksa nädalat ning selle maht on kolm EAP (1 EAP = 26 tundi). Iga nädal tuleb õppijatel lahendada viis kohustuslikku ülesannet (eranditeks on nädalad, kus õppijad kirjutavad lõputööd), mis kontrollivad vastava nädala teemade arusaamist. Esimesel kolmel nädalal tuleb lahendada ka teooria teste, mis kontrollivad Python'i keele põhiliste mõistete teadmist.

E-kursuse eesmärgiks on aidata õppijatel omandada algoritmilist mõtlemisviisi, tutvustada esmaseid programmeerimisvõtteid ja anda ülevaate programmeerimiskeelest Python. Viimane võib tulla eriti kasuks, sest Pythoni keelt õpitakse nii mõneski kõrgkoolis. See on esimeseks õpitavaks programmeerimiskeeleks ka näiteks Tartu Ülikooli bakalaureuse informaatika õppekavas [6].

### **2.1 E-kursuse sihtrühm**

E-kursus on mõeldud gümnaasiumiõpilastele Eestis ning on saadaval nii eesti kui ka vene keeles. On juhtumeid, kus suure soovi korral aktsepteeritakse ka neid, kes ei ole gümnaasiumiõpilased.

Kuigi e-kursuse tase on mõeldud pigem algajatele, siis võivad seda võtta ka need, kes on programmeerimisega varem kokku puutunud. Lisaks Pythoni baastadmisi kontrollivatele ülesannetele on ka graafiliste mängude koostamise ülesanded, mis võivad pakkuda huvi ja väljakutset edasijõudnud õpilastele.

## **2.2 E-kursuse probleemid**

E-kursuse põhiliseks probleemiks on vananemine. Nii materjal kui ka ülesanded on olnud ilma suuremaid muudatusi aastast 2012. Sellega võib kaasneda õpilaste motivatsiooni langus, teades, et kursuse materjal on aegunud. Lisaks sellele on soodustatud mahakirjutamise võimalus varasematelt õpilastelt, sest ülesanded jäävad aastast aastasse samasugusteks.

Teisest küljest, on vanadel harjutustel potentsiaalset kasvuruumi täiendusteks ja parandusteks. Kuna kursuse veebipõhine õpik on kohati uuendatud, siis avab see ka uusi võimalusi kodutööde osas. Saab koostada palju interaktiivsemaid ja huvitavamaid harjutusi, kui ka nõuda õpilastelt raskemate ja pikemate harjutuste koostamist, sest uuendatud õpik hõlmab endas rohkem õpetuslikke videosid ja konkreetsemaid näiteid.

Samuti vanade harjutuste hulgas esineb ülesandeid, millistel on vähe seost mängude ja lõputöö koostamisega. Vastupidiselt on ka teemasid, mis vajaksid tegelikkusest rohkem tähelepanu ning mitmekordset harjutamist. Kursusel esinebki ka selliste harjutuste puudus, mis seoksid omavahel kokku eelnevalt õpitud teadmised, seeläbi näidates kõikide õpitud teemade ja programmeerimistehnikate olulisust.

Lisaks kõigele eelnevale on e-kursusel ka harjutusi, kus on toodud vähe näiteid ja selgitusi ülesande eduka sooritamise kohta, mis võib õpilasi segadusse viia. Muuhulgas on ka õppenädalaid, kus on lihtsalt vähe kohustuslikke ülesandeid.

## **2.3 Sarnased kursused**

Tartu Ülikoolil on ka mitmeid teisi Pythoni programmeerimiskursuseid, mis on mõeldud algajatele. Selliseks e-kursuseks on näiteks „Programmeerimisest maalähedaselt“[7], mis

kestab neli nädalat. Sarnaselt „Teeme ise arvutimänge – algus“ kursusele, õpivad kursuse osalejad veebipõhise õpiku järgi ning teevad kontrollülesandeid. Erinevuseks on see, et „Programmeerimisest maalähedaselt“ kursusel on palju vähem kontrollülesandeid (seitse ülesannet kogu kursuse kohta) ning ülesanded on lihtsamad. Ülesannete lahendusi kontrollib automaatkontroller ning ei anna personaalseid soovitusi. Kuna kursuse „Teeme ise arvutimänge – algus“ eesmärk on õpetada programmeerima arvutimänge, siis antud kursusel käsitletakse kaudselt ka objektorienteeritud programmeerimise põhiaspekte. See eeldab osalejatelt rohkem tööd ja intensiivsemat harjutamist, võrreldes kursusega „Programmeerimisest maalähedaselt“, E-kursus „Programmeerimisest maalähedaselt“ andis bakalaureusetöö autorile hea idee kasutada kontrollülesannete selgitustes programmitöö väljundist pilte, et „Teeme ise arvutimänge – algus“ osalejatel oleks kergem kontrollülesannete lõpptulemustest aru saada. Teiseks sarnaseks e-kursuseks Tartu Ülikooli arvutiteaduse instituudi poolt on „Programmeerimise alused“[8]. See kursus, sarnaselt „Teeme ise arvutimänge – algus“ e-kursusega, kestab kaheksa nädalat. Kuigi õppematerjal ja teemad on sarnased „Teeme ise arvutimänge – algus“ e-kursusega, on sellel kursusel üks huvitav iseärasus. Iga nädal on kursuse osalejatel kohustus lahendada kolm kerget kontrollülesannet ning üks kolmest raskest kontrollülesandest. Sedaviisi on õpilastel võimalus valida endale sobilik ülesanne, kui üks tundub liiga raske või igav. Kuigi „Teeme ise arvutimänge – algus“ ei kasuta täpselt samasugust süsteemi omandatud teadmiste kontrollimiseks, tehti uuendatud e-kursusele *nuputusülesanded*, mida tohib asendada ükskõik millise teise kontrollülesandega.

### 3. Kursuse muudatused

Antud peatükk räägib muudatustest, mis selle bakalaureusetöö vältel olid tehtud e-kursusele “Teeme ise arvutimänge - algus”. Kõik muudatused, mis olid tehtud kodustele ülesannetele, on kajastatud nii eestikeelses, kui venekeelses kursuses “Teeme ise arvutimänge - algus”.

Suurem osa muudatusi oli tehtud koduste ülesannete valdkonnas. Kõige tähtsam eesmärk oli luua uued ülesanded, mis oleksid huvitavad ning väljakutset pakkuvad, säilitades samal ajal tiheda seose uue õpiku materjali ja videonäidetega. Selleks, et ülesanded oleksid huvitavad, pidid loodud ülesanded olema rohkem interaktiivsemad ning mängulisemad. Kursuse vanade ülesannete kasutamine oli vaieldamatult hea Pythoni põhitõdede kontrollimiseks, kuid harjutustes puudus sisu, mille olemasolu teeks ülesande lahendamise põnevamaks. Lisaks sellele, on paljud uued ülesanded seotud mängude tegemisega, mis juba kursuse algusest annab õppijatele kogemust ja ideid lõpuprojekti (enda mängu) koostamiseks.

Selleks, et pakkuda õppijatele väljakutset, olid uued ülesanded koostatud sedamoodi, et iga nädal alustatakse lühemate ülesannete lahendamisega ning lõpetatakse pikemate ülesannetega. Pikemad ülesanded pole alati raskemad, kuid vajavad kindlasti rohkem tähelepanu, kannatlikust ning erinevate õpitud teadmiste kombineerimist. Tihtipeale tähendab see, et igas järgnevas ülesandes tuleb kasutada eelnevates ülesannetes harjutatud teadmisi ja tehnikaid. Sealjuures, et ülesanded poleks aga liiga keerulised, on peaaegu kõikide ülesannete kirjelduses olemas ka õigesti töötava programmi väljundist pilt, mis aitaks õpilastel paremini aru saada programmi eesmärgist.

Uuendatud kursuse üheks uueks iseärasuseks on see, et mõningatel nädalatel on olemas nuputamisülesanded, mida õpilased saavad teha oma oskuste harjutamiseks või boonuspunktide kogumiseks. Kuigi need ülesanded on natukene raskemad teistest ülesannetest, sobivad nad nii edasijõudnutele õppijatele kui ka algajatele. Need ülesanded on pandud nädalatele, kus kohustuslikke ülesandeid on vähem (kursuse lõpu poole).



E-kursusel olid tehtud ka väiksemad administratiivsed muudatused. Nimelt vahetati Moodle õpikeskkonnas ülesannete esitamise keskkond VPL (*Visual Programming lab*) [9] ülesannete esitamise keskkonna vastu. Erinevalt Moodle'i tavalisest ülesannete esitamise keskkonnast, pakub VPL erinevaid lisavõimalusi. Näiteks saavad õppijad sealsamas veebilehitsejas, peale kodutöö esitamist, muuta ja täiendada oma koodi, ilma et nad peaksid uuesti seda üles laadima. Lisaks sellele saavad õpilased jookсутada oma programme otse veebilehitsejast. Seda sama saavad teha ka kursuse juhendajad, mis teeb kodutööde kontrollimise tunduvalt kergemaks. Samuti saavad juhendajad samas keskkonnas teha esitatud tööde plagiiaadi kontrolli, millega on kerge kursuse aususel silma peal hoida.

### **3.1 Kodused ülesanded**

Siin peatükis vaatame me läbi kõik muudatused, mis olid tehtud eestikeelsele kursusele ülesannete osas. Nagu sai varem mainitud, siis need muudatused olid tehtud ka selle e-kursuse venekeelsele versioonile, väiksemate muudatustega. Ülesannete koostamisel järgiti nii kursuse õpiku kui ka Pythoni kodulehe [10] õpetusi.

#### **3.1.1 Esimene nädal**

Esimese nädala õpiku peatükk on suuremalt jaolt sissejuhatav osa programmeerimiskeelde Python. Esimene peatükk annab hea ülevaate muutujate kasutamiseviisidest, erinevatest andmetüüpidest (sõned, täisarvud, ujukomaarvud), matemaatiliste tehete tegemisest, moodulite kasutamisest, andmete sisse lugemisest failist kui ka elementaarsetest programmi sisendi ja väljundi funktsioonidest.

#### **Ülesanne 1.1 – Vanuse väljastamine**

Iga nädal alustatakse kergemate ülesannetega. Esimeses ülesandes peavad õppijad näitama, et nad tunnevad programmeerimiskeele Pythoni põhiprintsiipe ning oskavad kasutada algelisi informatsiooni sisestamise, salvestamise (muutujas) ja väljastamise viise. Muuhulgas on ülesandele lisatud ka väike täiendus, mis sunnib õpilasi juba algusest peale vahet tegema ja

töötama erinevate andmetüüpidega. Nimelt peavad õpilased sisendina küsima programmi kasutajalt nende nime ja vanust (täisarv) ning seejärel liitma sellele ühe. Esmapilgul võib algajatele tunduda see väga kerge, kuid kuna sisendina võetud andmed on algselt alati sõned, siis tuleb vanusele ühe liitmiseks vahetada sisendi abil salvestatud andmete tüüpi sõnest arvuks. Juba varakult sellest konkreetsest protsessist aru saamine võib tunduvalt vähendada probleeme programmeerimisega tulevikus.

Võttes eelkirjeldatud arvesse, oli esimene kontrollülesanne tehtud nõnda, et see kontrolliks järgmist:

- muutujale väärtuse omistamine
- sisendi funktsiooni kasutamine
- andmetüüpide tundmine ning muutmine
- mitme erineva andmetüübiga lause väljastamine
- täisarvude liitmine

Uuendatud ülesande püstitus on toodud allpool.

*Kirjuta programm, mis küsib sisendiks kasutaja nime ja vanuse. Nende andmete põhjal peaks programm väljastama lause, mis ütleb kui vanaks (täisarv) kasutaja varsti saab.*

*\*\*Pange tähele, et kõik sisendid (input) salvestatakse algselt sõnedena (string)\*\**

Väljundi näidis:

```
>>> %Run Ylesanne-1-1.py
Sisestage oma nimi: Mari
Sisestage oma vanus: 16
Mari saab varsti 17 -aastaseks.
>>>
```

*Pilt 1: Esimese nädala esimese ülesande väljundi näidis.*

## Ülesanne 1.2 – Juku teekond koju

Teine ülesanne keskendub rohkem matemaatiliste tehete tegemisele, kuid see hõlmab ka juba eelmisest ülesandest kinnitatud teadmiste harjutamist. Õpilased peavad harjutuse sisu järgi aru saama, et neilt on alguses nõutud leida täisnurkse kolmnurga hüpotenuus, kasutades Pythagorase teoreemi. Selle käigus tuleb õpilastel näidata, et nad oskavad kasutada ka raskemaid matemaatilisi tehteid Pythoni keeles, nagu näiteks arvu ruudu ja ruutjuure võtmine. Ruutjuure võtmiseks tuleb õppijatel importida *math* moodul, kuid nutikamad võivad seda teha astendamise kaudu, mis ei nõua *math* mooduli importimist.

Teine kontrollülesanne oli tehtud nõnda, et see kontrolliks järgmisi teadmisi:

- abimoodulite importimine (valikuline)
- raskemad matemaatilised tehted
- eelmises ülesandes harjutatud teadmised

Uuendatud ülesande püstitus on toodud allpool.

*Juku kõnnib koolist koju. Ta ei soovi kõndida otse koju üle muru, vaid kõnnib mööda teepeenart. Selleks, et jõuda koju mööda teepeenart, peab ta kõndima 400 meetrit otse ning seejärel pöörama paremale (90 kraadi) ning kõndima veel 300 meetrit.*

*Kirjuta programm, mis väljastab, mitu minutit saaks Juku kokku hoida, kui ta kõnniks sirgelt koju üle muru, teades, et Juku kõnnib 50 meetrit minutis.*

***\*\*Ruutjuure võtmiseks võib vaja minna abimooduli *math* importimine\*\****

Väljundi näidis:

```
>>> %Run Ylesanne-1-2.py
      Juku hoiaks kokku 4 minutit
>>>
```

*Pilt 2: Esimese nädala teise ülesande väljundi näidis.*

### **Ülesanne 1.3 – Tekstifaili sisu lugemine**

Kolmas ülesanne tutvustab õpilastele algelisel tasemel failidega töötamise. Harjutuse käigus peavad õpilased näitama, et oskavad Pythoni abil avada faile ning väljastada sees olevaid andmeid ekraanile. Samuti saavad õppijad uuesti harjutada ka *input* funktsiooni kasutamist, kuna ülesanne nõuab programmil küsida sisendina kasutajalt avatava faili nime.

Kolmas kontrollülesanne oli tehtud nõnda, et see kontrolliks järgmisi teadmisi:

- Pythoniga faili lugemine ning sisu väljastamine
- eelmistes ülesannetes harjutatud teadmised

Uuendatud ülesande püstitus on toodud allpool.

*Kirjuta programm, mis küsib kasutajalt faili nimetust ning seejärel väljastab ekraanile faili sisu. Tehke endale tekstifail ning kontrollige, kas programm töötab. Ärge unustage, et igal tekstifailil on olemas ka laiendus .txt*

### **Ülesanne 1.4 – Kõrbe ületamine**

Neljandas ülesandes peavad õppijad taas näitama oma matemaatilisi oskusi. Ülesande põhimõte on kontrollida ühte spetsiifilist ala Pythoni programmeerimises, milleks on jagamine. Selle harjutuse käigus, on vaja õpilastel jagada antud teepikkuse kiirusega, mis omakorda annab neile ajalise väärtuse, mida tuleb väljastada ekraanile. Kuid ülesannet teeb tunduvalt raskemaks see, et ajalist väärtust tuleb väljastada korrektse lausena päevades ja tundides. Selle saavutamiseks tuleb kasutada täisosalist jagamist, jagatise jäägi leidmist ning ülesande püstitusest lähtuvat ümardamist.

Neljas kontrollülesanne oli tehtud nõnda, et see kontrolliks järgmisi teadmisi:

- täisosaline jagamine
- jäägi leidmine
- ümardamine
- eelmistes ülesannetes harjutatud teadmised

Uuendatud ülesande püstitus on toodud allpool.

*Otsustasid kaameli seljas läbida kõrbe. Eeldame, et koos sinuga ja sinu asjadega suudab kaamel kõndida keskmiselt kiirusega 3,5 km/h. Läbida tuleb 202 km. Harjutamise mõttes, võime eeldada, et kaamel ei pea vahepeal tegema puhkepause.*

*Kirjuta programm, mis väljastab mitu päeva ja tundi(ümardatult lähima täisarvuni) kuluks kõrbe ületamiseks.*

Väljundi näidis:

```
>>> %Run Ylesanne-1-4.py
      Kõrbe ületamiseks kulub 2 päeva ning 10 tundi
>>>
```

*Pilt 3: Esimese nädala kolmanda ülesande väljundi näidis.*

## Ülesanne 1.5 - Önnenumber

Kui esimese nädala tähtsaimad aspektid ja tehnikad on läbi harjutatud, võib asuda ka põnevamate harjutuste poole. Viiendas harjutuses puutuvad õppijad esmaselt kokku *randint* funktsiooniga, mille ülesandeks on juhusliku arvu genereerimine. See funktsioon muutub ajapikku väga tähtsaks, sest väga paljud mängud on üles ehitatud just juhuslikkuse peal. Esimesel nädalal ei puudutata õpikus veel juhuslikkuse teemat, kuid kasutades ülesande juhiseid ja näpunäiteid saavad õpilased juba varakult aimu *random* mooduli kasutamise kasulikkusest.

Viies kontrollülesanne oli tehtud nõnda, et see kontrolliks järgmisi õpilaste teadmisi:

- abimoodulite ja funktsioonide kasutamine
- eelmistes ülesannetes harjutatud teadmised
- abi otsimine Internetist (foorumitest ja Pythoni dokumentatsioonist)

Uuendatud ülesande püstitus on toodud allpool.

*Uuri abimoodulit random ja abifunktsiooni randint (from random import randint). Miks randint(0,5) annab suvalise täisarvu 0'st 5'ni?*

*Kirjuta programm, mis küsib kasutajalt tema nime ja kaks täisarvu (võib eeldada, et esimene arv on väiksem kui teine). Seejärel võtab programm need kaks arvu ning seab nad parameetriteks randinti abifunktsioonile.*

*Programm peaks väljastama lause, kus on öeldud kasutaja tänane õnnenumber.*

Väljundi näidis:

```
>>> %Run Ylesanne-1-5.py
Sisestage oma nimi: Mari
Sisestage esimene täisarv: 1
Sisestage teine täisarv: 100
Mari, sinu tänane õnnenumber on 47 !
>>>
```

*Pilt 4: Esimese nädala viienda ülesande väljundi näidis.*

### 3.1.2 Teine nädal

Veebipõhise õpiku teise nädala peatükk keskendub põhiliselt kahele teemale – tingimuslaused ja tsüklid. Tingimuslausete teemas tutvustatakse sellised võtmesõnad nagu *if*, *elif* ja *else*. Tsüklite osas õpitakse aru saama erinevatest tsüklite lausetest (*for*, *while*, *do ... while*) ning nende erinevustes. Kuna need teemad on üldse väga tähtsad programmeerimises, siis ka selle nädala kontrollülesanded keskenduvad nende tõhusale harjutamisele.

Tahaks juhtida tähelepanu ka sellele, et teise nädala esimesel ja teisel ülesandel ei esine suuri muutusi võrreldes eelmise aastate ülesannetega. Selline otsus on tehtud seetõttu, et need kaks ülesannet on suuremal määral loomingulised ning sobivad väga hästi selle nädala sissejuhatuseks.

#### Ülesanne 2.1 – Põgenemismängu algoritm

Teise nädala esimese ülesande põhimõte seisneb selles, et kontrollida õppijate oskust kasutada õpikus olevaid instruksioone plokk skeemi koostamise osas. Plokk skeemi koostamise käigus peaksid õppijad täpsemini aru saama tingimuste kontrollimise protsessist, mis on järgmistes ülesannetes vajalik.

Esimene kontrollülesanne oli tehtud nõnda, et see kontrolliks järgmisi teadmisi:

- plokk skeemi koostamine
- *if/elif/else* funktsioonide tundmine

Uuendatud ülesande püstitus on toodud allpool.

*Mõttele välja üks seikluslugu, mis on seotud mingist kohast või situatsioonist põgenemisega. Loo vältel peab tegelane tegema õigeid otsuseid selleks, et valitud asukohast või situatsioonist välja saada. Seikluslugu tuleb ka hiljem realiseerida mänguks, kuid alguses teeme endale mugava skeemi, mille järgi hiljem koodi kirjutada.*

### **Ülesanne:**

- Joonista mängu täpne plokk skeemi algoritm
- Skeemil peab kindlasti olema näha:
  - mängu algus
  - kõik vahepealsed hargnevused/sammud
  - mängu lõpp
- Püüa skeemi joonistamisel vältida kattuvaid jooni

### **Ülesanne 2.2 – Põgenemismäng ise**

Teises ülesandes on vajalik eelnevalt koostatud plokk skeemi põhjal koostada Pythoni programm. Programm kujutab endast mängu, kus kasutajal tuleb teha otsuseid ning jõuda mängu lõppu. Kuna ülesanne on suuremal määral loominguline, siis sobib see nii algajatele kui ka neile, kellel juba on kogemust tingimuste kontrollimisega. Edasijõudnud saavad keskenduda rohkem loomingulisuse aspektile, algajad aga rohkem tehnilisele poolele. Erinevalt teistest ülesannetest, avalikustavad õpilased oma tehtud töid foorumis, kus ka teised õpilased saavad neid katsetada ja mängida.

Teine kontrollülesanne oli tehtud nõnda, et see kontrolliks järgmisi teadmisi:

- *if/elif/else* funktsioonide kasutamine

Uuendatud ülesande püstitus on toodud allpool.

*Realiseeri oma seiklusmäng Pythonis. Lisa programmi mõistlikud kommentaarid. Pane programmi lõppu `input()` käsk, sest siis saavad teised kohe sinu programmi ilma IDLE-sse laadimata katsetada. Hinda teiste programme, anna konstruktiivseid kommentaare.*

### **Ülesanne 2.3 – Täringu viskamise mäng**

Selle nädala kolmas kontrollülesanne keskendub mitmele erinevale asjale. Eduka ülesande sooritamiseks tuleb õpilastel kasutada oma oskusi tsüklite valdkonnas kui ka meelde tuletada



oma eelnevalt harjutatud programmeerimisteadmisi, milleks on juhusliku arvu genereerimine ning tingimuste kontroll. Ülesanne on üles ehitatud nõnda, et õppijad saaksid meelde tuletada ja harjutada eelnevalt õpitu. Ülesande tulemus on veel üks mäng, mida saab mängida konsooli abil.

Kolmas kontrollülesanne oli tehtud nõnda, et see kontrolliks järgmisi teadmisi:

- *if/elif/else* funktsioonide kasutamine
- tsüklite kasutamine
- juhusliku arvu genereerimise kordamine

Uuendatud ülesande püstitus on toodud allpool.

*Kirjuta programm, mis simuleerib täringu viskamise mängu. Kasutaja mängib arvuti vastu, kus mõlema korral visatakse täringut (randint) ning väljastatakse seejärel visatud tulemused ning tolle partii võitja. Kui üks partiidest on viik, siis ükski mängijatest punkti ei saa. Mäng kestab 5 partiid, kus lõpus väljastatakse mängu tulemus ning võitja.*

Väljundi näidis:

```
>>> %Run Ylesanne-2-3.py
Arvuti viskas... 3
Kasutaja viskas... 2
Arvuti võitis selle korra.

Arvuti viskas... 3
Kasutaja viskas... 1
Arvuti võitis selle korra.

Arvuti viskas... 1
Kasutaja viskas... 3
Kasutaja võitis selle korra.

Arvuti viskas... 3
Kasutaja viskas... 5
Kasutaja võitis selle korra.

Arvuti viskas... 5
Kasutaja viskas... 3
Arvuti võitis selle korra.

Kasutaja võitis kokku: 2 mängu.
Arvuti võitis kokku: 3 mängu.
Arvuti on võitja!

>>>
```

Pilt 5: Teise nädala kolmanda ülesande väljundi näidis.

## Ülesanne 2.4 – Arvamise mäng

Kuna tsüklite ja tingimuslausete teema on tähtis, siis need harjutatakse ka neljandas ülesandes. Õpilased peavad programmeerima mängu, kus mängijal tuleb hakata arvama arvuti poolt genereeritud juhusliku arvu. Väga sarnaselt eelmisele ülesandele, tuleb ka siin kasutada peamiselt kolme tähtsamat teadmist – juhusliku arvu genereerimine, tingimuste kontroll, tsüklid. Kuid erinevalt eelmisest, pole selles ülesandes enam kindlat tsükli korduste arvu, kuna mängu pikkus oleneb kasutajast. Samuti tuleb sellele mängule lisada ka raskusastme reguleerimise võimaluse, mis muudab mängu huvitavamaks.

Neljas kontrollülesanne oli tehtud nõnda, et see kontrolliks järgmisi teadmisi:

- *if/elif/else* funktsioonide kasutamine
- tsüklite kasutamine
- juhusliku arvu genereerimise kordamine

Uuendatud ülesande püstitus on toodud allpool.

*Programmeeri üks arvu arvamise mäng. Põhimõte on ära arvata, millisest arvust arvuti parajasti mõtleb. Programm peab andma võimaluse valida raskusastmete vahel, mis seab vastavalt arvamise piirkonna (näiteks: kerge 1-2, keskmine 1-4, raske 1-6). Mängijal on 3 elu ning iga vale arvamise korral kaotab ta ühe elu.*

*Programm peaks väljastama peale igit korda, kas mängija arvas õigesti või valesti. Vale arvamise korral tuleks välja tuua, millisest arvust siis tegelikult arvuti mõtles.*

*Mängu lõpus väljastage ka mängija tulemus.*

Väljundi näidis:

```
>>> %Run Ylesanne-2-4.py
Valige raskusaste - kerge/keskmine/raske: keskmine
Arvuti mõtleb arvu 1'st 4'ni

Kuidas arvad, millisest numbrist arvuti mõtleb? 3
Arvasid ära!

Kuidas arvad, millisest numbrist arvuti mõtleb? 2
Kahjuks arvasid valesti! Arvuti mõtles numbrist 1

Kuidas arvad, millisest numbrist arvuti mõtleb? 4
Kahjuks arvasid valesti! Arvuti mõtles numbrist 1

Kuidas arvad, millisest numbrist arvuti mõtleb? 1
Kahjuks arvasid valesti! Arvuti mõtles numbrist 3

Mäng on lõppenud
Sinu tulemus on: 1 p

>>>
```

Pilt 6: Teise nädala neljanda ülesande väljundi näidis.

### 3.1.3 Kolmas nädal

Kolmanda nädala õpiku peatükk käsitleb tsükleid ja liste. Erinevalt eelmisest nädalast, vaadatakse tsüklite keerukamaid variante nagu kahekordsed ja mitmekordsed tsüklid. Listide osas saavad õppijad selgeks, kuidas lisada, kustutada, otsida, muuta ja sorteerida elemendid listis. Lisaks sellele on kolmanda nädala materjalis ka üks lühike peatükk, mis hõlmab graafilist programmeerimist. Õppijad õpivad joonistama ekraanile erinevaid kujundeid kasutades *Turtle* moodulit.

#### Ülesanne 3.1 – Laevade pommitamine

Kolmanda nädala sissejuhatav ülesanne kontrollib õpilaste teadmisi kahekordsete tsüklite kohta. Ülesanne on püstitatud nõnda, et see oleks kerge ja mänguline. Õpilased peavad koostama kahekordse tsükli abil visuaalse esituse mänguplatsist.

Esimene kontrollülesanne oli tehtud nõnda, et see kontrolliks järgmisi teadmisi:

- kahekordsed tsüklid

Uuendatud ülesande püstitus on toodud allpool.

*Kuna tegeleme siin kursusel mängude temaatikaga, siis tihtipeale on mängudel olemas ka mänguplatsid. Selles ülesandes ongi meil vaja luua mänguplats ühe kuulsa mängu jaoks, milleks on laevade pommitamine.*

*Põhimõtte on küsida kasutajalt platsi suurus (Väike – 5x5 /Keskmine – 7x7 /Suur – 10x10 ühikut) ning vastavalt sisestatud suurusele väljastada väljak, mis koosneb merelainetest ehk „~“ sümbolitest.*

Väljundi näidis:

```
>>> %Run Ylesanne-3-1.py
Valige, mis suurusega väljakul soovite mängida (Väike/Keskmine/Suur)
Suurus: Väike

~ ~ ~ ~ ~
~ ~ ~ ~ ~
~ ~ ~ ~ ~
~ ~ ~ ~ ~
~ ~ ~ ~ ~
>>>
```

Pilt 7: Kolmanda nädala esimese ülesande väljundi näidis.

### Ülesanne 3.2 – Kilpkonna laevade pommitamine

Kolmanda nädala teise ülesande visuaalne lõpptulemus on peaaegu identne esimese ülesandega. Seetõttu on ülesande püstitusest arusaamine väga kerge. Õpilased peavad sarnaselt eelmise ülesandega koostama kahekordsete tsüklite abil visuaalse esituse mänguplatsist. Erinevus on selles, et õppijad peavad tegema seda *Turtle* mooduli abil.

Teine kontrollülesanne oli tehtud nõnda, et see kontrolliks järgmisi teadmisi:

- *Turtle* mooduli kasutamine
- kahekordsed tsüklid

Uuendatud ülesande püstitus on toodud allpool.

*Turtle* on nn kilpkonnagraafika moodul, kus kolmnurgaline kilpkonn saab teha ekraanile lihtsaid graafilisi joonistusi kilpkonna käskudega, mõned käsud on toodud kommentaarides. Kõikvõimalikud käsud, kui neid peaks vaja minema, leiad *Turtle* mooduli dokumentatsioonist veebis (<https://docs.Python.org/2/library/turtle.html>).

Analoogselt eelmise ülesandega, joonistame uuesti laevade pommitamise platsi. Kuid seekord juba *Turtle* abimooduli abil. Muidugi tuleb ka siin kasutada mitmekordseid tsükleid.

Tulemus võiks olla umbes selline (piisab 5x5 mänguplatsist, kuid soovi korral võite teha platsi suuremaks ning teha lained rohkem lainelisemaks ning värvilisteks):



Pilt 8: Kolmanda nädala teise ülesande Turtle akna näidis.

### Ülesanne 3.3 – Kullakoobaste sorteerimine

Kolmandas ülesandes kontrollime listidega töötamise teadmisi. Õppijad peavad olemasoleva listi muutma vastavalt instruksioonile ülesande kirjelduses. See harjutus on hea, sest kontrollib korraga mitmeid erinevaid listiga töötamise funktsioone.

Kolmas kontrollülesanne oli tehtud nõnda, et see kontrolliks järgmisi õpilaste teadmisi:

- listidega töötamine (*index/sort/reverse/insert*)

Uuendatud ülesande püstitus on toodud allpool.

*Üks muinasmaa lohe otsustab oma finantsid korda teha ning tegeleda natukene oma kullakoobaste sorteerimisega. Järgmises listis on näha iga koopa kullamüntide arvu.*

*koopad = [1284,5801,8368,8074,6155,2287]*

*Kirjuta programm, mis*

- *näitab esialgu antud koobaste listi*
- *seejärel näitab, millises koopas on kõige rohkem kullamünti.*
- *seejärel liidab juurde kõige suurema müntide arvuga koopale esimese koopa mündid ehk 1284 (pange tähele, et esimesse koopasse jääb siis 0 münti).*
- *seejärel sorteeriks koopad müntide järgi kahanevas järjekorras*
- *seejärel lisaks listi esimesele kohale veel ühe uue koopa, milles on 5000 kullamünti*
- *väljastab tulemuse ekraanile*

Väljundi näidis:

```
>>> %Run Ylesanne-3-3.py
Vana koobaste list: [1284, 5801, 8368, 8074, 6155, 2287]
Vanas listis asub kõige rohkem kulda koopas nr.3
Uuendatud koobaste list: [5000, 9652, 8074, 6155, 5801, 2287, 0]
>>>
```

*Pilt 9: Kolmanda nädala kolmanda ülesande väljundi näidis.*

### Ülesanne 3.4 – Listi arvamine

Neljandas ülesandes peab kasutama nii tsüklite kui ka listide kohta teadmisi. Ülesanne on pikk, kuid see-eest sarnane eelmise nädala viimase ülesandega. Õpilased peavad koostama juhuslike arvudega listi, mis on mängija eest varjatud. Mängija peab seejärel püüdma ära arvata kõik numbrid tolles listis. Programmi koostamise käigus tuleb õpilastel kasutada ka eelnevalt õpitud teadmisi nagu tingimuste kontroll ja random abimoodul.

Neljas kontrollülesanne oli tehtud nõnda, et see kontrolliks järgmisi õpilaste teadmisi:

- tsüklid / topelt tsüklid
- listidega töötamine
- eelnevalt harjutatud teadmised

Uuendatud ülesande püstitus on toodud allpool.

*Koostage mäng, mis genereerib varjatud listi suvalise(randint) viie arvuga 0'st 9'ni (arvud listis võivad korduda).*

*Mängija peab ükshaaval arvama, millised numbrid seal sees on. Valesti arvamise arv on piiratud, mis omakorda tähendab, et tuleks anda mängijale valida mängu raskusastet. Näiteks kerge raskusastme korral võiks mängija maksimaalselt valesti arvata 5 korda, keskmise korral 4, raske korral 3.*

*Iga kord kui mängija arvab õigesti, siis peaks list selle võrra väiksemaks muutuma selleks, et samasugust arvu enam arvata ei saaks (pange tähele, et listis võivad arvud korduda – soovitatav kasutada while tsüklit kõikide nende eemaldamiseks).*

Programm peaks iga kord andma tagasiside selle kohta, kas mängija arvas õigesti või valesti ning milliseid numbreid pole veel arvamiseks kasutatud. Mäng lõpeb siis, kui valesti arvamiste limiit on täis või list on tühi. Mängu lõpus väljastage esialgne varjatud list. Kui mängija kaotas, siis tuleks väljastada koos mängu tulemusega ka list, kus näha, millised numbrid jäid varjatud listis arvamata.

Väljundi näidis:

```
>>> %Run Ylesanne-3-4.py
Valige raskusaste - kerge/keskmine/raske:raske

Valikvõimalused: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
Arvake, milline arv leidub listis: 5
Arvasid õigesti!

Valikvõimalused: [0, 1, 2, 3, 4, 6, 7, 8, 9]
Arvake, milline arv leidub listis: 3
Arvasid valesti!

Valikvõimalused: [0, 1, 2, 4, 6, 7, 8, 9]
Arvake, milline arv leidub listis: 2
Arvasid valesti!

Valikvõimalused: [0, 1, 4, 6, 7, 8, 9]
Arvake, milline arv leidub listis: 9
Arvasid valesti!

Esialgne varjatud list oli selline: [7, 1, 1, 4, 5]
Sa kaotasid mängu! Listis jäi veel järgi: [7, 1, 1, 4]

>>>
```

Pilt 10: Kolmanda nädala neljanda ülesande väljundi näidis.



### 3.1.4 Neljas nädal

Neljanda nädala materjal jaguneb kaheks – funktsioonid ja objektid. Funktsioonide osas õpitakse, kuidas koostada funktsioone, kuidas neid välja kutsuda, mis on funktsiooni argumendid, kuidas tagastada funktsiooni tulemust ning mis on lokaalsed ja globaalsed muutujad. Objektide osas õpitakse, kuidas luua ja kasutada objekte ning mis on objekti atribuudid, meetodid ja klassid.

Neljandal nädalal on kohustuslike kontrollülesandeid ainult kolm, sest õppijad peavad hakkama tegema juba esimesi samme lõpuprojekti koostamise suunas. See-eest on sellel nädalal kaks nuputamisülesannet, mis sobib nendele, kes tahavad lihvida oma programmeerimisoskusi või tahavad koguda boonuspunkte.

#### Ülesanne 4.1 – Kääbuste valuuta

Neljandat nädalat alustatakse ühe kergema funktsioonide teemat harjutava ülesandega. Õpilastel tuleb koostada programm, kus nad kasutavad funktsiooni, mis võtab argumendiks kasutaja sisendi ning tagastab ülesande instruktsioonile vastava väärtuse.

Esimene kontrollülesanne oli tehtud nõnda, et see kontrolliks järgmisi teadmisi:

- funktsioonide loomine/kasutamine
- funktsiooni väärtuse tagastamine

Uuendatud ülesande püstitus on toodud allpool.

*Kääbused otsustasid minna haldjate tänavatoidufestivalile ning osta endale maagilisi küpsiseid. Kääbused kasutasid valuutana hõbemünti, aga haldjad kasutasid valgeid kristalle. On teada, et üks haldjate valge kristall maksab 12 hõbemünti.*

*Koostage funktsioon, mis võtab argumendiks kääbuste hõbemünti ning **tagastab** ümardatult alla täisarvuni nende väärtust valgetes kristallides.*

*Programm peaks küsima hõbemüntide arvu ning väljastama funktsiooni abil nende väärtuse valgetes kristallides. Samuti väljastage kui palju küpsiseid saaksid kääbused osta, kui on teada, et üks küpsis maksab 3 valget kristalli.*

Väljundi näidis:

```
>>> %Run Ylesanne-4-1.py
Sisestage hõbemüntide arv: 235
Hõbemüntide väärtus valgetes kristallides: 19
Küpsiste arv, mida saab osta: 6
>>>
```

Pilt 11: Neljanda nädala esimese ülesande väljundi näidis.

## Ülesanne 4.2 – Tulnukate vahemaa

Selleks, et kinnitada teadmised funktsiooni koostamise osas, õppijad peavad veel ühe pikema ülesande lahendama, mis keskendub funktsioonide teemale. Selles ülesandes on tähtsal kohal ka varem õpitud teadmised, nagu näiteks tingimuste kontroll ja arvude ümardamine.

Teine kontrollülesanne oli tehtud nõnda, et see kontrolliks järgmisi teadmisi:

- funktsioonide loomine/kasutamine
- tingimuste kontroll
- arvude ümardamine

Uuendatud ülesande püstitus on toodud allpool.

*Teatavasti on tulnukatel kosmoselaev, mis suudab liikuda kiirusega 18 000 kilomeetrit tunnis. Kirjutage programm, mis küsiks kasutajalt vahemaa kilomeetrites (võime eeldada et vahemaa on täisarv), mis tuleb laevaga ületada. Samuti tuleks koostada funktsiooni nimega Aeg, mis võtab argumendiks vahemaa (kilomeetrites) ning **väljastab lause**, mis näitab vahemaa ületamiseks kuluva aja. Kuna tegemist on tulnukate poolt kasutusel oleva kõrgtehnoloogiaga, siis funktsioonil Aeg on ka paar iseärasust:*

*- kui aega kulub vähem kui päev, siis funktsioon väljastab selgitava lause tundides (ümardatud 2 kohta peale koma)*

*- kui kuluv aeg on võrdne ühe päevaga või ühe aastaga või on nende kahe vahel, (eeldame et aastat alati 365 päeva), siis funktsioon väljastab selgitava lause päevades (ümardatud 2 kohta peale koma)*

- kui kuluv aeg on rohkem kui üks aasta, siis väljastab programm selgitava lause aastates (ümardatud 2 kohta peale koma)

Programmi lõpus kutsuda välja funktsioon `Aeg`, mille argumendiks on kasutaja poolt sisestatud vahemaa.

Väljundi näidis:

```
>>> %Run Ylesanne-4-2.py
Sisestage vahemaa kilomeetrites: 1234567890
Aega sihtpunkti jõudmiseks kulub: 7.83 aastat
>>>
```

Pilt 12: Neljanda nädala teise ülesande väljundi näidis.

### Ülesanne 4.3 – Dinosaurused

Kolmas ülesanne keskendub objektide teemale. Kuna teema on iseenesest raske, siis kontrollülesanne on tehtud nõnda, et see sarnaneks õpiku uute videonäidetega. Õppijad peavad looma klassi, mis sisaldaks enda sees vastavalt ülesande kirjeldusele nimetatud atribuudid ja meetodid. Seejärel tuleb koostatud eksemplaride põhjal tegema objektid, mille info lõpptulemusena väljastatakse ekraanile.

Kolmas kontrollülesanne oli tehtud nõnda, et see kontrolliks järgmisi teadmisi:

- klasside/atribuutide/meetodite/objektide loomine ja kasutamine

Uuendatud ülesande püstitus on toodud allpool.

*Tehke programm, mis sisaldaks klassi nimega `Dinosaurus`. Selle sees peaks olema kirjelduse atribuut, mis kirjeldaks dinosauruse nimetust, kõrgust (meetrites) , värvi, ning kas ta on taimetoitlane või lihatoitlane*

*Lisaks tehke sinna ka info meetod, mis ütleks meile dinosauruse nimetuse, värvi ning toitumiseelistuse.*

*Seejärel tehke list nimetusega dinosaurused ning lisage sinna vähemalt viis eksemplari. Lõpptulemusena kasutage info meetodit, et väljastada ekraanile dinosauruste üldinfo.*

Väljundi näidis:

```
>>> %Run Ylesanne-4-3.py
    Stegosaurus on sinine ning ta on taimetoitlane.
    Türrannosaurus on roheline ning ta on lihatoiduline.
>>>
```

Pilt 13: Neljanda nädala kolmanda ülesande väljundi näidis.

#### Nuputamisülesanne 4.4 - Dinosaurused

Esimene nuputamisülesanne on eelmise (4.3) ülesande täiendus. Õppijad peavad lisama eelneva programmi peamisele klassile veel kaks meetodit. Ühe meetodi koostamisel tuleb jälgida ülesande kirjelduses olevaid instruktsioone ning teine on loominguline.

Neljas nuputamisülesanne oli tehtud nõnda, et see kontrolliks järgmisi teadmisi:

- meetodite loomine ja kasutamine

*Lisage klassi sisse meetod nimetusega suurus, mis vaataks dinosauruse kõrgust ning lähtudes sellest **tagastaks** lause, mis ütleks, kas ta on teiste dinosaurustega võrreldes:*

- väga suur ehk suurem kui 30 meetrit

- suur ehk suurem kui 10 meetrit

- keskmine ehk suurem kui 5 meetrit

- või väike

*Lisaks sellele meetodile, lisage veel üks meetod omal vabal valikul (võite Dinosaurus klassi ning kirjelduse atribuuti täiendada) ning väljastage tulemus.*

Väljundi näidis:

```
>>> %Run Ylesanne-4-4.py
Stegosaurus on sinine ning ta on taimetoitlane.
Ta on teiste dinosaurustega võrreldes keskmist suurust.

Türannosaurus on roheline ning ta on lihatoiduline.
Ta on teiste dinosaurustega võrreldes suur.

>>>
```

Pilt 14: Neljanda nädala neljanda nuputamisülesande väljundi näidis.

### Nuputamisülesanne 4.5 -Turtle tähed

Teine nuputamisülesanne nõuab mitme erineva teema teadmiste kombineerimist – funktsioonide koostamine, *Turtle* mooduli abil joonistamine, tsüklid ja juhuslikud arvud.

Õppijatele pakutakse ülesanne, mille käigus peavad nad koostama funktsiooni, mis võtaks argumendiks kasutaja poolt sisestatud arvu ning joonistaks vastava suurusega kujundi. Selleks, et lõpptulemus oleks huvitavam, peab programm joonistama kasutaja poolt sisestatud koguse kujundeid juhuslikesse kohtadesse ekraanile.

Viies nuputusülesanne oli tehtud nõnda, et see kontrolliks järgmisi teadmisi:

- funktsioonide loomine ja kasutamine
- tsüklite teema kordamine
- turtle mooduli kordamine

*Tehke programm, mis joonistaks Turtle mooduli abil tähti .*

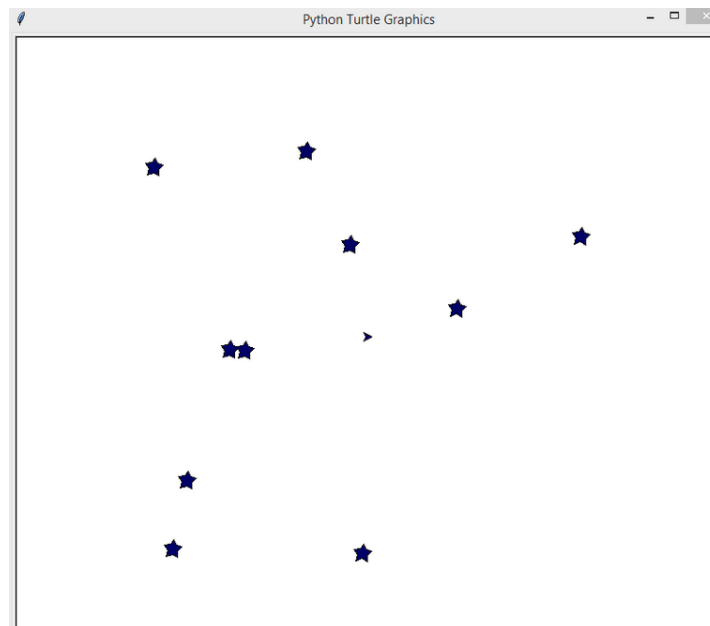
*Programmi koostamisel:*

- *tehke funktsioon, mis võtab argumendiks suuruse ning joonistab selle abil vastava suurusega tähe (täht peaks olema täidetud mingi värviga)*
- *programm küsib kasutajalt tähe suuruse ja tähtede arvu*
- *for tsükli ja randint'i abil joonistage juhuslikudesse kohtadesse nii palju tähti kui inimene algselt sisestas (tähed võivad ka juhuslikult kokku puutuda)*

Väljundi näidis:

```
>>> %Run Ylesanne-4-5.py
Sisestage tähtede suurus: 7
Sisestage tähtede arv: 10
>>>
```

Pilt 15: Neljanda nädala viienda nuputamisülesande väljundi näidis.



Pilt 16 Neljanda nädala viienda nuputamisülesande Turtle akna näidis.

### 3.1.5 Viies nädal

Viies nädal keskendub graafika loomisele lisamooduli Pygame abil. Käsitletakse järgmiseid aspekte: mängu akna ja tsükli loomine, joonistamine, piltide ja kujundite lisamine aknasse, kujundite ja piltide liigutamine, klaviatuurilt ja hiirelt informatsiooni vastuvõtmine ja kasutamine.

Sellel nädalal on minimaalselt kontrollülesandeid, sest lisaks ülesannetele peavad õpilased esitama ka oma lõputöö esimese osa.

#### Ülesanne 5.1 – Pygame taust

Esimeses ülesandes tahetakse kontrollida õpilase teadmisi graafilise akna loomises ning sinna joonistamises. Ülesanne on loominguline, kuid nõuab õpitu tähelepanelikku järgimist selleks, et kõik Pygame funktsioonid töötaksid õigesti.

Esimene kontrollülesanne oli tehtud nõnda, et see kontrolliks järgmisi teadmisi:

- Pygame akna loomine
- Pygame joonistamine

Uuendatud ülesande püstitus on toodud allpool.

*Kasutage Pygame ning joonistage selle abil üks pilt, mis võiks olla mõeldud mingisuguse mängu taustaks. See ei pea olema väga detailne. Põhimõtte on lihtsalt õppida ja harjutada kasutama kergemaid Pygame joonistamise funktsioone.*

*(pygame.draw.circle(), pygame.draw.rect(), pygame.draw.line())*

*Siis kui esitate oma tööd, siis palun kirjutage kommentaaridesse, millise mängu taustaga võiks olla tegu.*

*Nagu ka ennem mainitud, kodutöö koosneb ainult tausta joonistamist. Animatsiooni ja liikumist pole vaja sinna lisada.*

## Ülesanne 5.2 – Positsiooni muutmine

Teine kontrollülesanne nõuab, et õppijad oskaksid lisada pilte Pygame aknasse. Kui see on tehtud, siis programm peaks olema täiendatud sedaviisi, et iga kord kui kasutaja vajutab hiirega ekraanile, siis pilt peab muutma oma asukohta.

Teine kontrollülesanne oli tehtud nõnda, et see kontrolliks järgmisi teadmisi:

- Pygame piltide lisamine
- hiire nuppu vajutuselt informatsiooni vastuvõtmine ja kasutamine

Uuendatud ülesande püstitus on toodud allpool.

*Tehke Pygame'ga programm, kus programmi aknas on valge taust ning mingisugune väike pilt (näiteks mingi tegelane).*

*Tehke nii, et iga kord, kui klõpsate hiirega aknas suvalise koha peale, peab teie pilt muutma oma positsiooni (pilt ilmub kuskile teisse juhuslikult võetud kohta - randint).*

## Ülesanne 5.3 – Liikuv objekt

Kolmanda ülesande eesmärk on kontrollida, kas õppijad oskavad liigutada klaviatuuri nuppude abil Pygame aknas olevaid pilte või kujundeid. Selleks, et ülesanne oleks huvitavam, tuleb õppijatel teha programm sedaviisi, et kui pilt jõuab akna äärde, siis peab see vastaspool olevast äärest välja tulema (näiteks, kui jõuab paremasse äärde, siis peab vasakust äärest välja tulema). Antud ülesanne oli koostatud koostöös Jaan Jannoga.

Kolmas kontrollülesanne oli tehtud nõnda, et see kontrolliks järgmisi teadmisi:

- Pygame kujundi joonistamine / piltide lisamine
- klaviatuuril nuppude vajutusest informatsiooni vastuvõtmine ja kasutamine

Uuendatud ülesande püstitus on toodud allpool.

*Ülesandeks on koostada Pygame abil programm, mis teeb kõike järgnevat.*

- Ekraanil peab olema joonistatud kas pilt või kujund. Võib ise vabalt valida.



- Nooleklahvide abil on võimalik seda kujundit liigutada.
- Kui kujund läheb akna piirest välja, siis peab ta teisest otsast väljuma. Näiteks kui ta liigub üle vasaku akna ääre, siis peab ta paremalt äärelt välja ilmuma.

Aluseks võib kasutada järgmiste peatükkide videotest olevaid programme:

<https://moodle.ut.ee/mod/book/view.php?id=226365&chapterid=24624>

<https://moodle.ut.ee/mod/book/view.php?id=226365&chapterid=29611>

Need alusprogrammid juba katavad esimesed 2 nõuet, seega ise teha on kolmas nõue.

Vihje: kujundi asukoha aknas määravad näidisprogrammides muutujad "linnu\_x" ja "linnu\_y". Lahenduseni viib see, kui if'iga kontrollida nende koorinaatide väärtusi. Kui väärtusest paistab üle ääre minemine, siis muuta "linnu\_x" ja "linnu\_y" väärtus selliselt, et kujund on teises otsas.

Kuna näidise akna suurus on 640x480, on äärtel sellised väärtused:

- Kui "linnu\_x" on 0, siis on tegemist vasaku äärega.
- Kui "linnu\_x" on 640, siis on tegemist parema äärega.
- Kui "linnu\_y" on 0, siis on tegemist ülemise äärega.
- Kui "linnu\_y" on 480, siis on tegemist alumise äärega.

## 4. Tulemused

Selles peatükis vaadeldakse lähemalt e-kursuse õppijate tagasisidesid. Kõik tulemused ja tagasisidesid on võetud eestikeelsest e-kursusest „Teeme ise arvutimänge - algus“, mis oli läbiviidud uuendatud materjalidega 2017 aasta kevadel (27. märts kuni 21. mai 2017). E-kursusel osales 33 õpilast, millest 19 said edukalt lõpetatud.

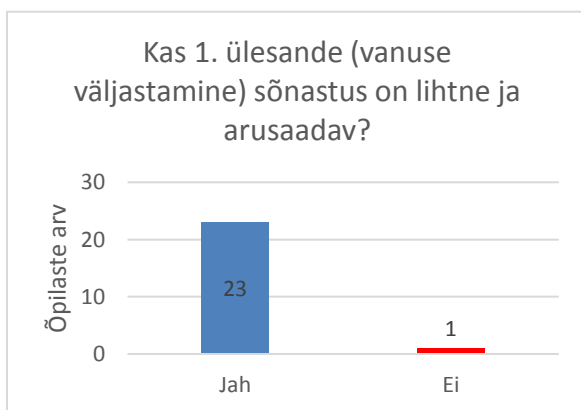
### 4.1 Õpilaste tagasiside

Kursuse esimesel viiel nädal oli õpilasi palutud täita tagasiside küsimustik. Iga kontrollülesande kohta esitati kolm küsimust:

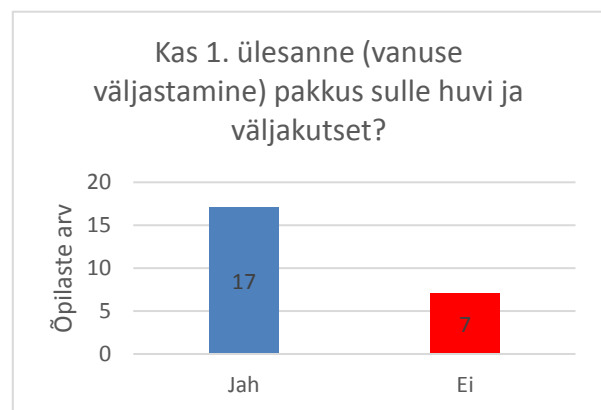
1. kas ülesande sõnastus on lihtne ja arusaadav
2. kas ülesanne pakkus huvi ja väljakutset
3. kui palju aega kulus ülesande lahendamise peale.

Lisaks sellele küsiti, milline oli vastava nädala lemmik ülesanne ning samuti esines võimalus anda vabatahtlikku täiendavat tagasisidet.

#### Esimese nädala tagasiside

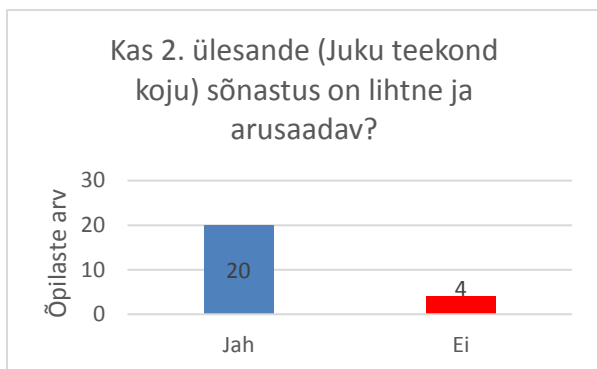


Pilt 18: Esimese nädala esimese ülesande lihtsuse ja arusaadavuse tagasiside.

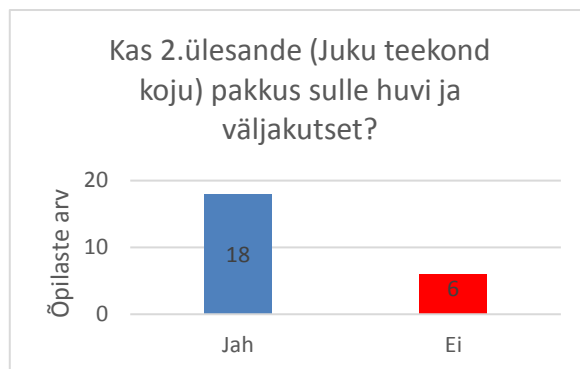


Pilt 17: Esimese nädala esimese ülesande huvitavuse ja väljakutse tagasiside.

Esimese ülesande tegemiseks kulus õpilastel keskmiselt 25 minutit.

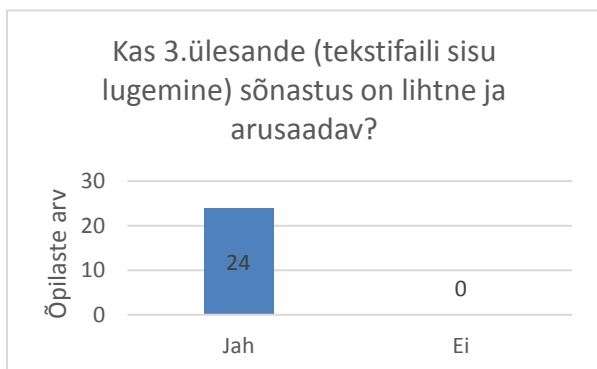


Pilt 19: Esimese nädala teise ülesande lihtsuse ja arusaadavuse tagasiside.

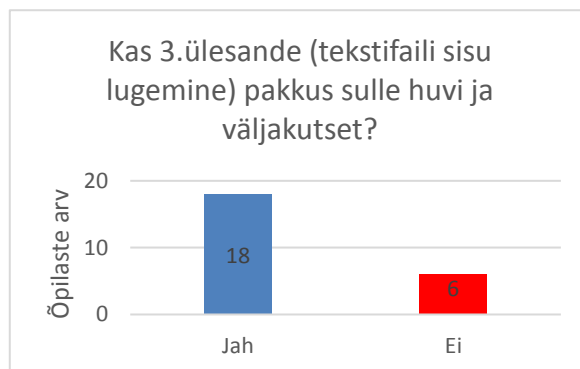


Pilt 20: Esimese nädala teise ülesande huvitavuse ja väljakutse tagasiside.

Teise ülesande tegemiseks kulus õpilastel keskmiselt 25 minutit.

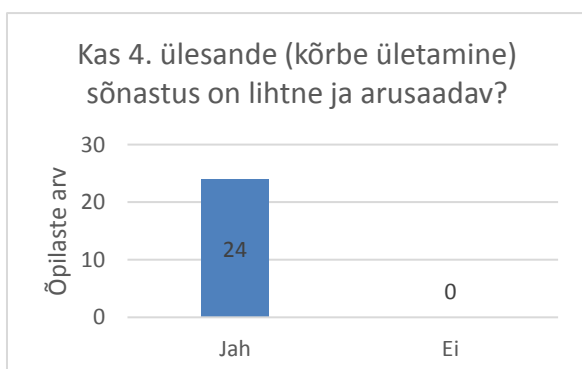


Pilt 21: Esimese nädala kolmanda ülesande lihtsuse ja arusaadavuse tagasiside.

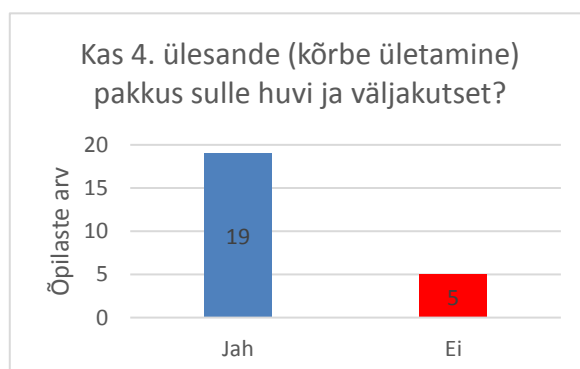


Pilt 22: Esimese nädala kolmanda ülesande huvitavuse ja väljakutse tagasiside.

Kolmanda ülesande tegemiseks kulus õpilastel keskmiselt 11 minutit.



Pilt 23: Esimese nädala neljanda ülesande lihtsuse ja arusaadavuse tagasiside.



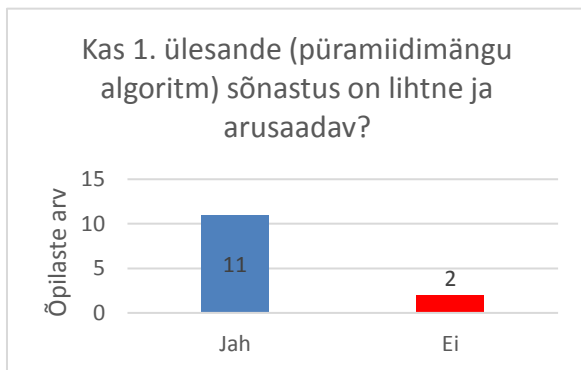
Pilt 24: Esimese nädala neljanda ülesande huvitavuse ja väljakutse tagasiside.

Neljanda ülesande tegemiseks kulus õppijatel keskmiselt 26 minutit.

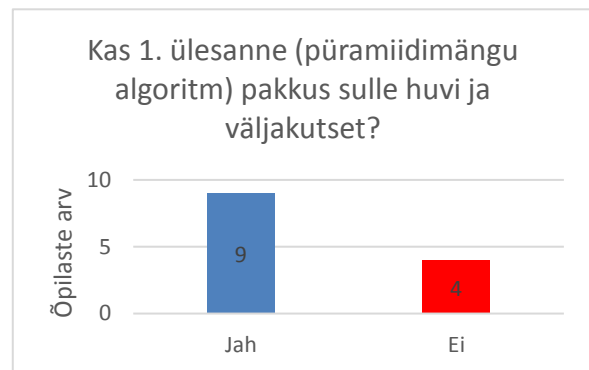
Esimese nädala ülesannete seast osutusid lemmikuteks esimene, teine ja neljas ülesanne (valiti võrdne arv õppijaid). Kolmas ülesanne osutus lemmikuks ainult ühel õpilasel. Enamasti kõik ülesanded olid kergesti arusaadavad ja huvitavad.

Antud kursusel ei võetud tagasisidet esimese nädala viienda ülesande kohta.

## Teise nädala tagasiside

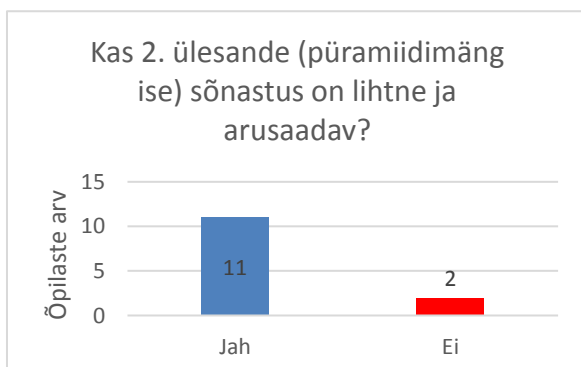


Pilt 25: Teise nädala esimese ülesande lihtsuse ja arusaadavuse tagasiside.

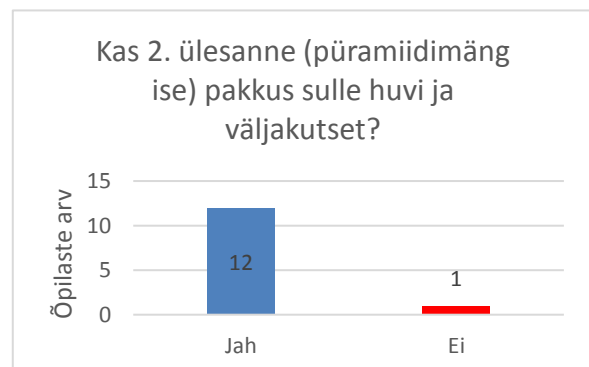


Pilt 26: Teise nädala esimese ülesande huvitavuse ja väljakutse tagasiside.

Esimese ülesande tegemiseks kulus õppijatel keskmiselt 62 minutit.

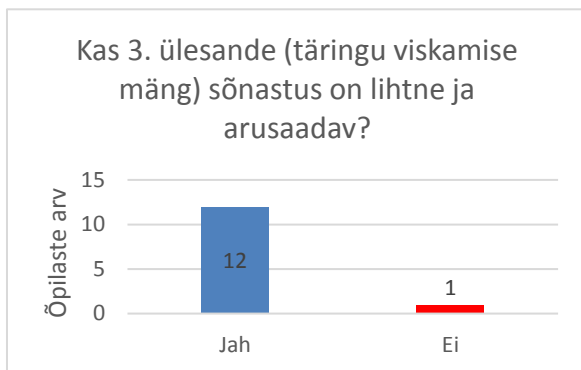


Pilt 27: Teise nädala teise ülesande lihtsuse ja arusaadavuse tagasiside.

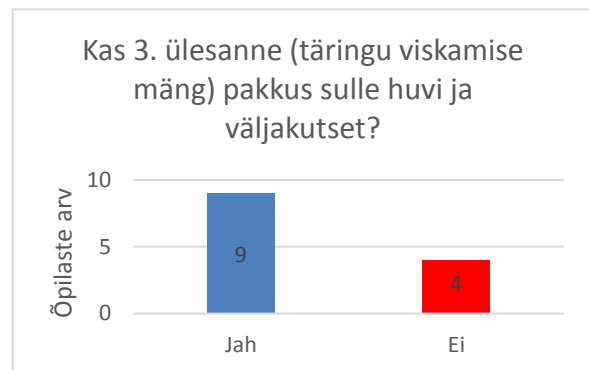


Pilt 28: Teise nädala teise ülesande huvitavuse ja väljakutse tagasiside.

Teise ülesande tegemiseks kulus õpilastel keskmiselt 64 minutit.

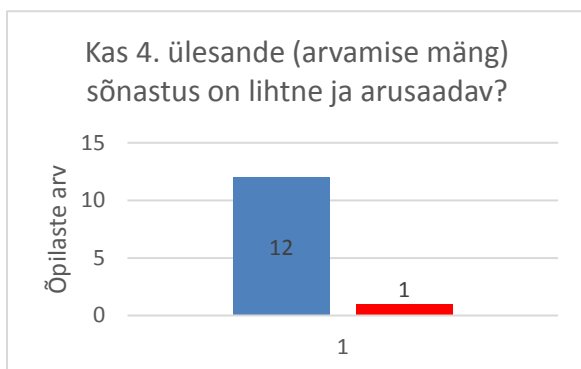


Pilt 19: Teise nädala kolmanda ülesande lihtsuse ja arusaadavuse tagasiside.

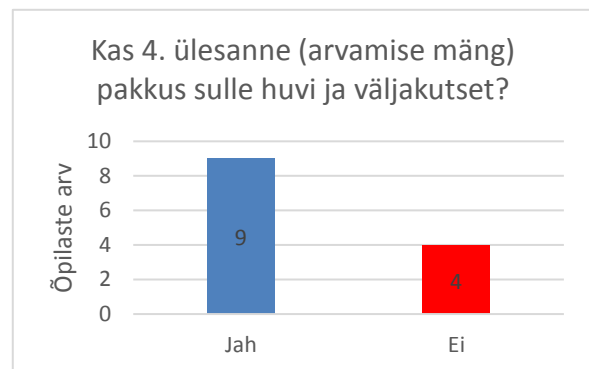


Pilt 20: Teise nädala kolmanda ülesande huvitavuse ja väljakutse tagasiside.

Kolmanda ülesande tegemiseks kulus õpilastel keskmiselt 48 minutit.



Pilt 22: Teise nädala neljanda ülesande lihtsuse ja arusaadavuse tagasiside.

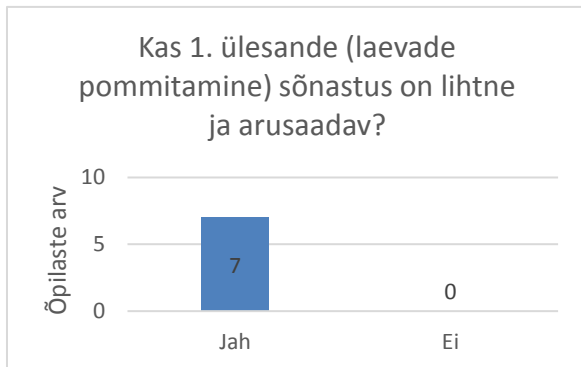


Pilt 21: Teise nädala neljanda ülesande huvitavuse ja väljakutse tagasiside.

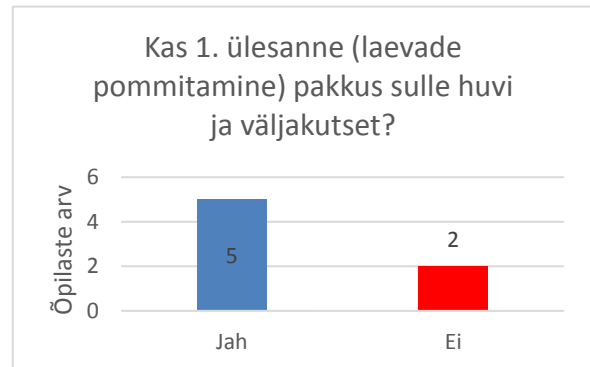
Neljanda ülesande tegemiseks kulus õpilastel keskmiselt 46 minutit.

Suuremale osale õpilastest pakkusid teise nädala ülesanded väljakutset ja huvi ning jäid sõnastuse poolest lihtsaks ja arusaadavaks. Nädala lemmikülesandeks sai püramiidimängu programmeerimine.

## Kolmanda nädala tagasiside

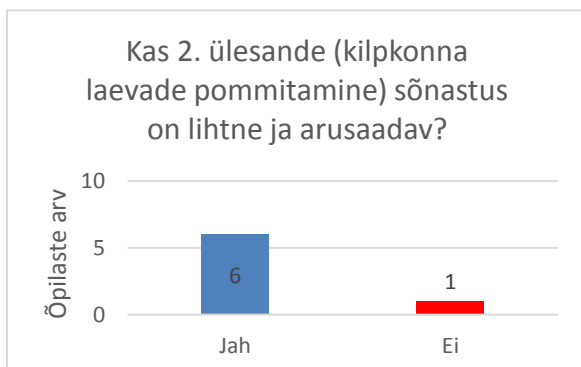


Pilt 23: Kolmanda nädala esimese ülesande lihtsuse ja arusaadavuse tagasiside.

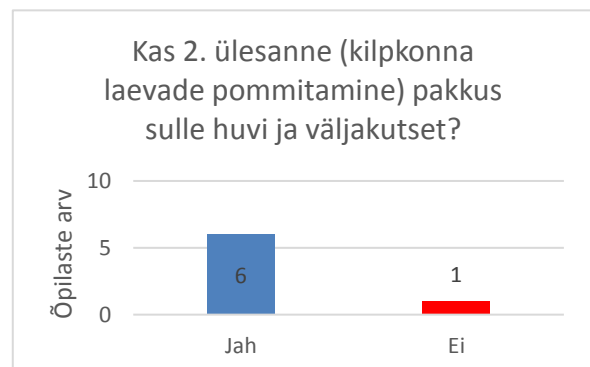


Pilt 24: Kolmanda nädala esimese ülesande huvitavuse ja väljakutse tagasiside.

Esimese ülesande tegemiseks kulus õpilastel keskmiselt 21 minutit.

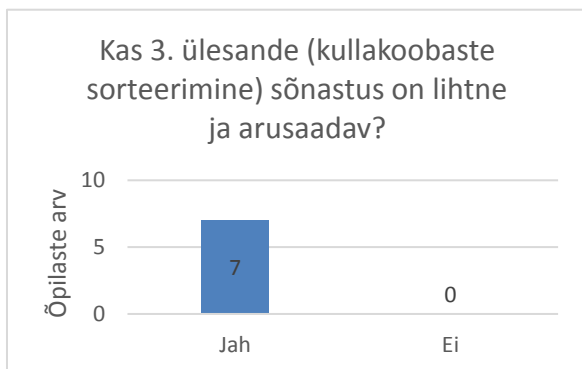


Pilt 26: Kolmanda nädala teise ülesande lihtsuse ja arusaadavuse tagasiside.

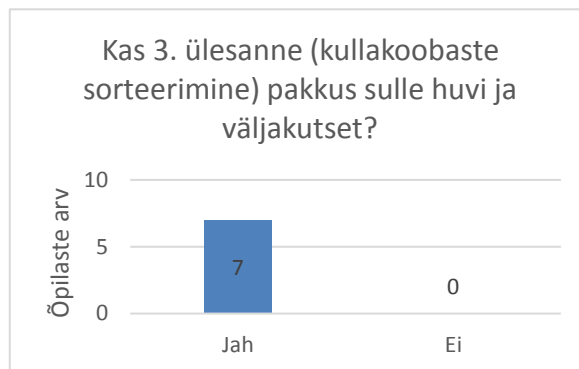


Pilt 25: Kolmanda nädala teise ülesande huvitavuse ja väljakutse tagasiside.

Teise ülesande tegemiseks kulus õpilastel keskmiselt 41 minutit.

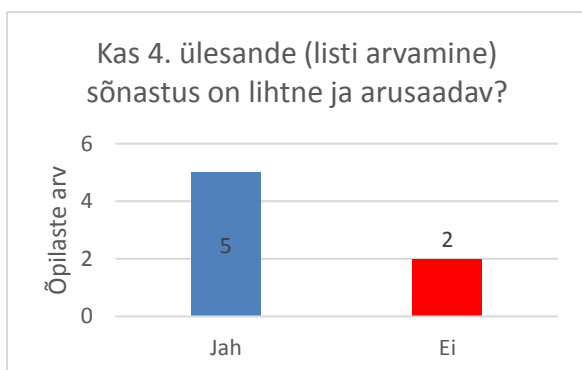


Pilt 27: Kolmanda nädala kolmanda ülesande lihtsuse ja arusaadavuse tagasiside.

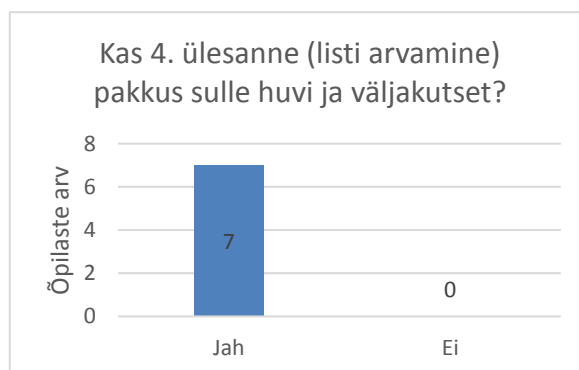


Pilt 28: Kolmanda nädala kolmanda ülesande huvitavuse ja väljakutse tagasiside.

Kolmanda ülesande tegemiseks kulus õpilastel keskmiselt 47 minutit.



Pilt 29: Kolmanda nädala neljanda ülesande lihtsuse ja arusaadavuse tagasiside.

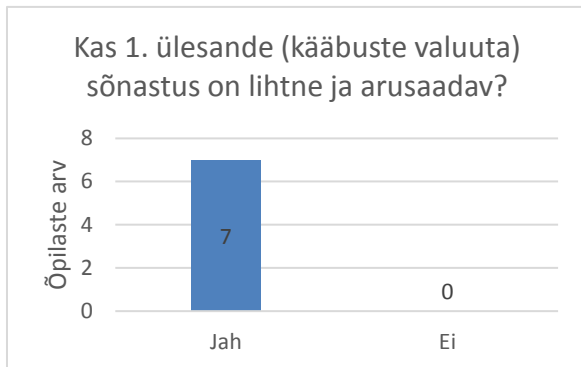


Pilt 30: Kolmanda nädala neljanda ülesande huvitavuse ja väljakutse tagasiside.

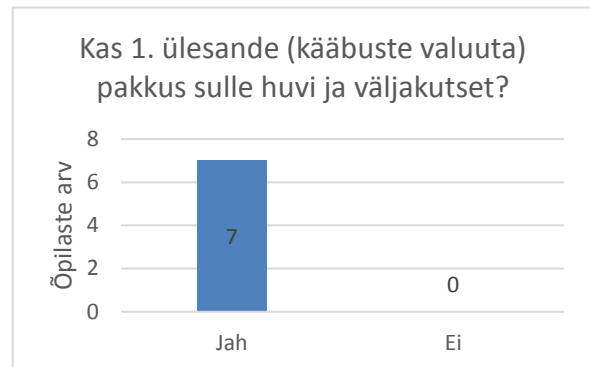
Neljanda ülesande tegemiseks kulus õpilastel keskmiselt 100 minutit.

Enamus selle nädala ülesannete sõnastus oli õpilastele arusaadav ning ülesanded pakkusid huvi ja väljakutset. Nädala lemmikülesandeks sai neljas ülesanne – listi arvamine.

## Neljanda nädala tagasiside

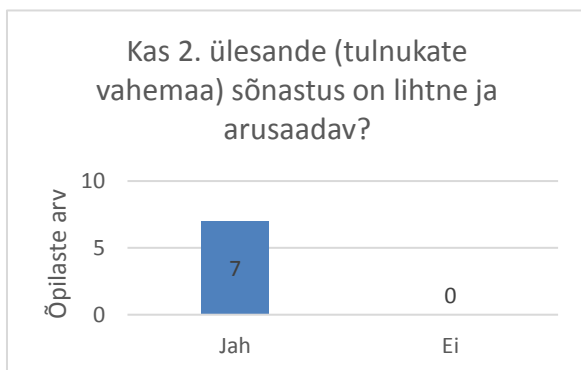


Pilt 32: Neljanda nädala esimese ülesande lihtsuse ja arusaadavuse tagasiside.

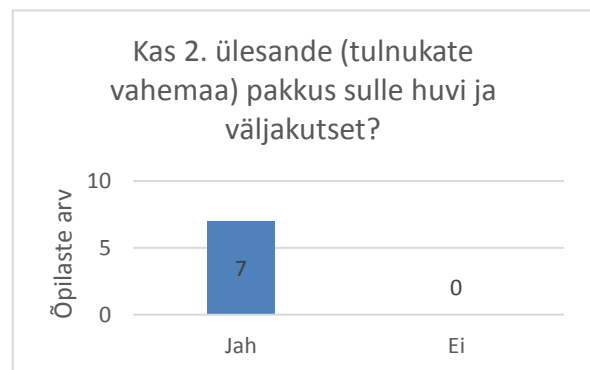


Pilt 31: Neljanda nädala esimese ülesande huvitavuse ja väljakutse tagasiside.

Esimese ülesande tegemiseks kulus õpilastel keskmiselt 21 minutit.

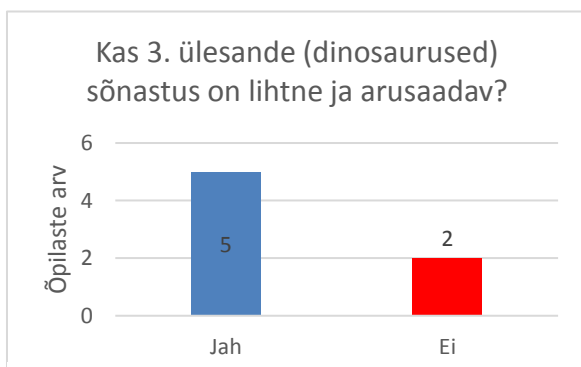


Pilt 43: Neljanda nädala teise ülesande lihtsuse ja arusaadavuse tagasiside.

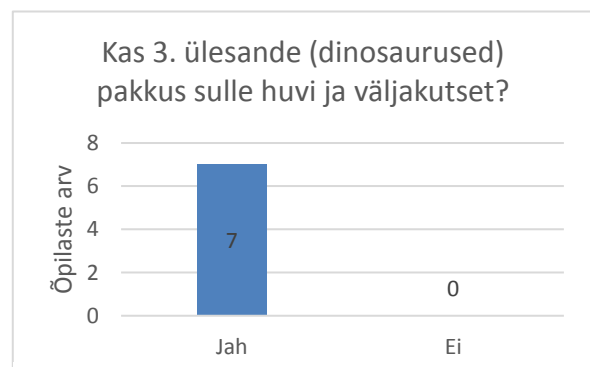


Pilt 44: Neljanda nädala teise ülesande huvitavuse ja väljakutse tagasiside.

Teise ülesande tegemiseks kulus õpilastel keskmiselt 30 minutit.



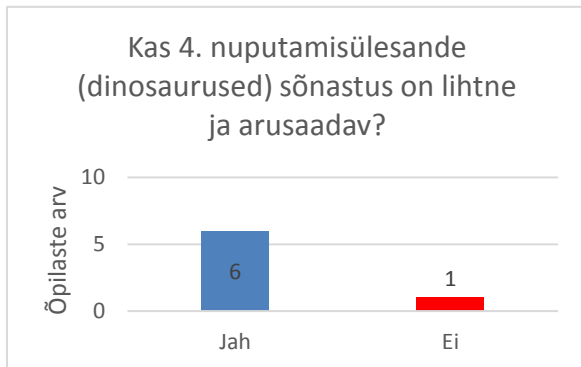
Pilt 34: Neljanda nädala kolmanda ülesande lihtsuse ja arusaadavuse tagasiside.



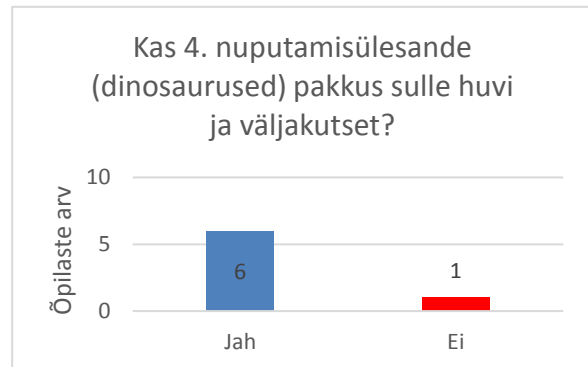
Pilt 33: Neljanda nädala kolmanda ülesande huvitavuse ja väljakutse tagasiside.



Kolmanda ülesande tegemiseks kulus õpilastel keskmiselt 32 minutit.

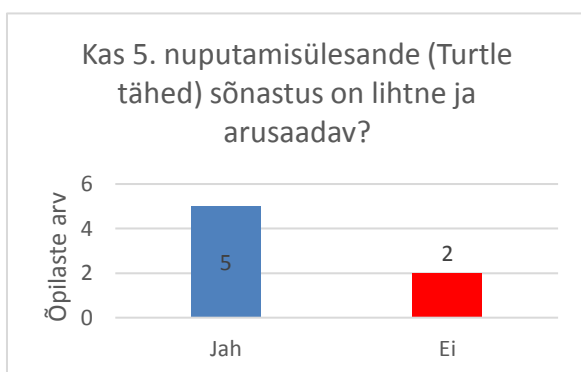


Pilt 36: Neljanda nädala neljanda nuputamisülesande lihtsuse ja arusaadavuse tagasiside.

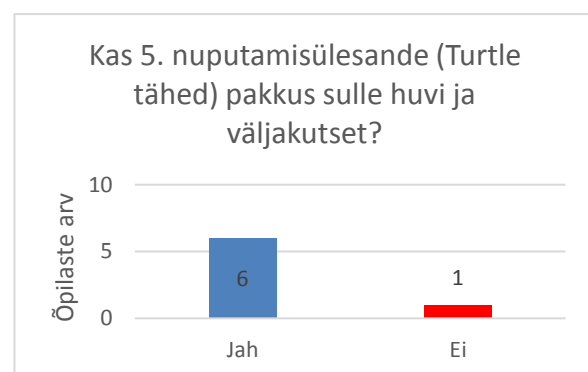


Pilt 35: Neljanda nädala neljanda nuputamisülesande huvitavuse ja väljakutse tagasiside.

Neljanda ülesande tegemiseks kulus õpilastel keskmiselt 24 minutit.



Pilt 49: Neljanda nädala viienda nuputamisülesande lihtsuse ja arusaadavuse tagasiside.

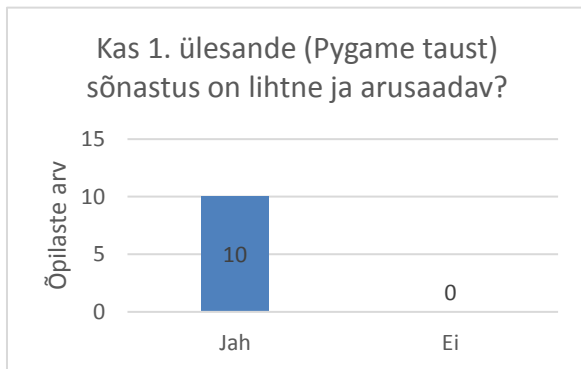


Pilt 50: Neljanda nädala viienda nuputamisülesande huvitavuse ja väljakutse tagasiside.

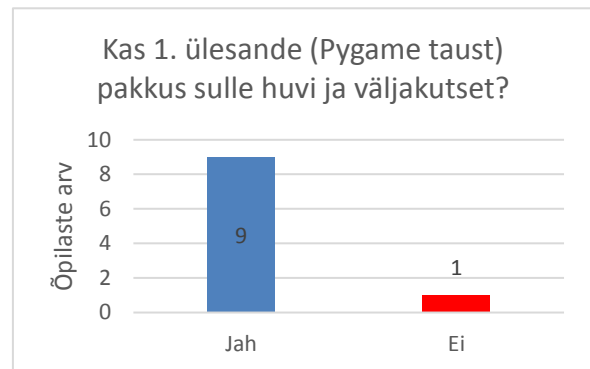
Viienda ülesande tegemiseks kulus õppijatel keskmiselt 37 minutit.

Neljanda nädala ülesanded olid enamasti hästi arusaadavad ning meeldisid õppijatele. Lemmikülesandeks saab see nädal nimetada koguni kolm ülesannet, mis said võrdse arvu hääli – tulnukate vahemaa, dinosaurused, Turtle tähed.

## Viienda nädala tagasiside

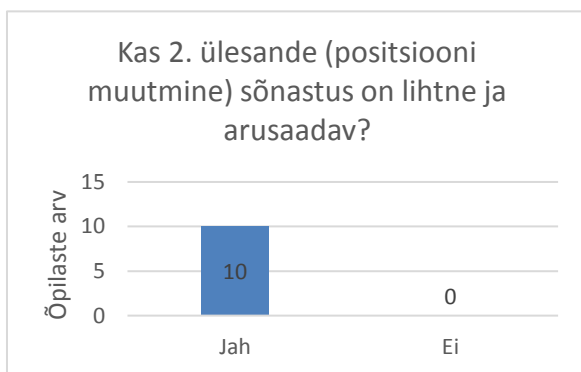


Pilt 38: Viienda nädala esimese ülesande lihtsuse ja arusaadavuse tagasiside.

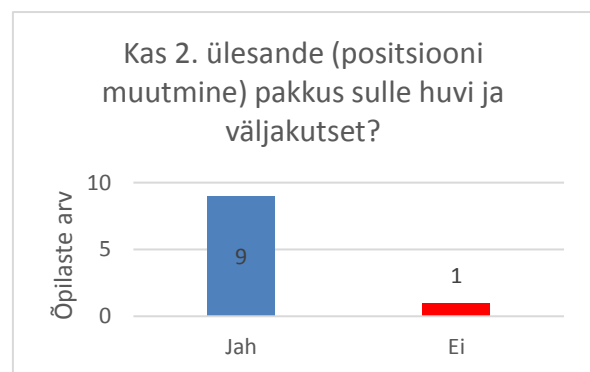


Pilt 37: Viienda nädala esimese ülesande huvitavuse ja väljakutse tagasiside.

Esimese ülesande tegemiseks kulus õpilastel keskmiselt 23 minutit.



Pilt 40: Viienda nädala teise ülesande lihtsuse ja arusaadavuse tagasiside.



Pilt 39: Viienda nädala teise ülesande huvitavuse ja väljakutse tagasiside.

Teise ülesande tegemiseks kulus õpilastel keskmiselt 35 minutit.

Viienda nädala ülesanded oli sõnastuse poolest arusaadavad ning pakkusid õpilastele huvi ja väljakutset. Lemmikülesandeks sai kolmas ülesanne - Pygame taust.

Antud kursusel ei võetud tagasisidet viienda nädala kolmanda ülesande kohta.

## 4.2 Järeldused

Üldiselt olid ülesannete tagasisided positiivsed. Kõige rohkem probleeme ülesande arusaamisega oli esimese nädala teises ülesandes - Juku teekond koju (neljal õppijal). Arusaamatus oli tingitud sellest, et need õppijad ei omanud nõutud matemaatilisi teadmisi ülesande täitmiseks (õppijad polnud veel koolis kokku puutunud Pythagorase teoreemi teemaga). Kuna kursus on eelkõige mõeldud gümnasistidele, siis ülesande koostajal on õigus eeldada, et õpilased peaksid seda teemat juba oskama.

Kursuse vältel olid kõvasti ülekaalus hääled selle poolt, et harjutused olid huvitavad ja väljakutset pakkuvad. Oli ka ülesandeid, mis tundusid mõnedele õpilastele igavavõitu, kuid iga nädalaga selliste ülesannete arv vähenes. Selline suhtumine ülesannetesse võis olla tingitud sellest, et kursusel olid õpilased, kes juba algselt oskasid mingil määral programmeerida. Iga nädalaga muutusid teemad raskemaks ja huvitavamaks, mistõttu ka edasijõudnud õpilased hakkasid oma meelt muutma ülesannete suhtes. Alates kolmandast nädalast, kus õpiku teemad kui ka harjutused muutusid keerulisemaks, vähenesid drastiliselt arvamused selle kohta, et mõni ülesanne pole väljakutset pakkuv või huvitav.

## 5. Kokkuvõte

Veebipõhisel e-kursusel „Teeme ise arvutimänge – algus“ esines läbi aastate mitmeid probleeme, mille tõttu kursuse kvaliteet ajapikku langes. Bakalaureusetöö käigus olid mõned probleemid lahendatud ning kursus sai suure uuenduse. Selleks, et olla kindel, et tehtud muudatused on head, viidi õpilaste seas läbi tagasiside küsitlused, mille tulemused osutusid positiivseteks. Lisaks sellele muutus ka kursuse juhendamine mugavamaks, mis on tingitud VPL ülesannete esitamise keskkonna kasutamisele võtmisele. Tänu muudatustele on koduseid ülesandeid kergem kontrollida ja võrrelda teiste õpilaste töödega.

Selleks, et kursus ka edaspidi säilitaks hea kvaliteedi, tuleb teatud aja tagant kontrollülesandeid muuta, et hoida ära õpilastevahelisi ülesannete jagamisi. Muidugi tuleb kontrollülesandeid muuta ka juhul, kui uuendatakse uuesti kursuse õpikut. Lisaks sellele võiks leida rohkem mooduseid, et edasijõudnud õpilastel oleks kursuse esimestel nädalatel huvitavam. See polnud küll suureks probleemiks, kuid kindlasti tõstaks mõningate õpilaste rahulolu kursusega.

## Viited

- [1] Nele-Mai Olup, Pärlin Luhila. Vaata, mis on kõrgkoolide populaarseimad erialad (2015). Külastatud 05.08.2018. <https://www.postimees.ee/3253047/vaata-mis-on-korgkoolide-populaarseimad-erialad>
- [2] Jaan-Juhan Oidermaa, Grete-Liina Roosve. Graafiline ülevaade: Eesti ülikoolide kõige populaarsemad erialad (2018). Külastatud 05.08.2018. <https://novaator.err.ee/844467/graafiline-ulevaade-eesti-ulikoolide-koige-populaarsemad-erialad>
- [3] Elisabeth Kungla. Vaata, millised on tänavused kõige populaarsemad erialad (2017). Külastatud 05.08.2018. <https://www.postimees.ee/4170137/vaata-millised-on-tanavused-koige-populaarsemad-erialad>
- [4] Tiina Kull. „Teeme ise arvutimänge – algus“ e-kursuse materjal (2012). Külastatud 05.08.2018. <http://www.e-ope.ee/repositoorium?@=7knk>
- [5] Jaan Janno. Sissejuhatav mänguarenduse ja programmeerimise kursuse õppematerjal (2017) . Külastatud 05.08.2018. [https://comserv.cs.ut.ee/home/files/Janno\\_softwareengineering\\_2017.pdf?study=ATILoputoo&reference=942502F1A166E21D3D1F15587BF9F760EB2C5EF3](https://comserv.cs.ut.ee/home/files/Janno_softwareengineering_2017.pdf?study=ATILoputoo&reference=942502F1A166E21D3D1F15587BF9F760EB2C5EF3)
- [6] Aivar Annamaa. Programmeerimise õpik. Külastatud 05.08.2018. <https://progeopik.cs.ut.ee/>
- [7] Tartu Ülikool. Programmeerimisest maalähedaselt (2018). Külastatud 05.08.2018. <https://courses.cs.ut.ee/2018/progmaa/spring/Main/HomePage>
- [8] Tartu Ülikool. Programmeerimise alused (2017). Külastatud 05.08.2018. <https://courses.cs.ut.ee/2017/eprogalused/fall/Main/HomePage>
- [9] Visual Programming lab. Külastatud 05.08.2018. <http://vpl.dis.ulpgc.es/>
- [10] The Python Tutorial. Külastatud 05.08.2018. <https://docs.python.org/3/tutorial/index.html>

## **Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks**

Mina, Mark Muhhin

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) enda loodud teose  
Ülesannete komplekt ainele "Teeme ise arvutimänge - algus",

mille juhendaja on Ljubov Jaanuska

- 1.1. reprodutseerimiseks säilitamise ja üldsusele kättesaadavaks tegemise eesmärgil, sealhulgas digitaalarhiivi DSpace-is lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;
  - 1.2. üldsusele kättesaadavaks tegemiseks Tartu Ülikooli veebikeskkonna kaudu, sealhulgas digitaalarhiivi DSpace'i kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.
2. olen teadlik, et punktis 1 nimetatud õigused jäävad alles ka autorile.
3. kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest tulenevaid õigusi.

Tartus, **09.08.2018**