

TARTU ÜLIKOOL
Arvutiteaduse instituut
Informaatika õppekava

Säde Mai Krusberg
Gümnaasiumi valikkursuse „Tarkvaraarendus“
PySimpleGUI õppemooduli loomine
Bakalaureusetöö (9 EAP)

Juhendaja: Tauno Palts, MA

Tartu 2021

Gümnaasiumi valikkursuse „Tarkvaraarendus“ PySimpleGUI õppe- mooduli loomine

Lühikokkuvõte:

Töö eesmärgiks on luua Tartu Ülikooli informaatika didaktika töörühma koosseisus eesti-keelsed õppematerjalid ning ülesannete kogu Eesti gümnaasiumi informaatika valikkursusele „Tarkvaraarendus“. Uute materjalide abil saab programmeerimise algkursused läbinu oma teadmised ja oskused proovile panna, õppides juurde graafilise kasutajaliidese loomist. Materjalid toetuvad programmeerimiskeelele Python ning selles tutvustatakse graafilise kasutajaliidese moodulit PySimpleGUI. Loodud tööle andsid tagasisidet nii didaktika töörühma liikmed, kahe gümnaasiumi õpilased ja õpetajad kui ka informaatikaõpetajate konverentsil osalenud õpetajad. Saadud tagasiside põhjal viidi sisse viimased muudatused ning lisati materjalid nii õpilastele kui ka õpetajatele avalikuks kasutamiseks veebiõpikusse.

Võtmesõnad: Graafiline kasutajaliides, PySimpleGUI, informaatika, gümnaasium, haridus

CERCS: P175 Informaatika, süsteemiteooria, S270 Pedagoogika ja didaktika

Creating a PySimpleGUI module for a programming course “Software development” taught in upper secondary schools

Abstract:

The aim of this study is to create a graphical interface study materials for a programming course “Software development” taught in Estonian upper secondary schools. Materials are based on a programming language Python and the module that is being taught is PySimpleGUI. The author of this study was also a member of the University of Tartu’s informatics didactics workgroup. The feedback for the new materials was received from the members of the workgroup, the students of two upper secondary schools and the teachers of informatics who participated in a conference. Based on the comments the last changes were conducted. Finally, the materials were uploaded to an e-book.

Keywords: Graphical user interface, PySimpleGUI, informatics, upper secondary school, education

CERCS: P175 Informatics, systems theory, S270 Pedagogy and didactics

Sisukord

Sissejuhatus	5
1. Programmeerimise õpetamine ja graafilise kasutajaliidese loomine koolis	7
1.1 Programmeerimise õpetamise olulisus	7
1.2 Programmeerimise õpetamine Eestis	8
1.3 Gümnaasiumi informaatika uued valikkursused	11
1.4 Graafiline kasutajaliides	13
1.4.1 Kasutajaliidese olulisus tarkvaraprojektis	13
1.4.2 Kasutajaliides Pythoni näitel	14
1.5 Õppematerjalide koostamise eeskirjad	15
2. Metoodika	18
2.1 Õppematerjali koostamise protsess	18
2.2 Tagasiside kogumine	20
3. Tulemused	21
3.1 Valminud õppematerjalid	21
3.1.1 PySimpleGUI ja esimene näide	21
3.1.2 PySimpleGUI nupuvajutus, hüpikaken ja uue graafikaakna avamine	22
3.1.3 PySimpleGUI graafikaakna vidinad	23
3.1.4 PySimpleGUI ja suurem näide	24
3.1.5 LISA: PySimpleGUI joonistamine	25
3.2 Valminud ülesannete kogu	26
3.3 Materjalide koostamine Merrilli mudeli põhjal	27
3.4 Tagasiside tulemused	32
3.5 Edasiarenduse võimalused	32
Kokkuvõte	34
Viidatud kirjandus	36

Lisad.....	39
I. Link loodud õppematerjalile „PySimpleGUI“	39
II. Link loodud ülesannete kogule.....	40
III. Litsents	41

Sissejuhatus

Kiiresti liikuv ja aina digitaliseeritumaks muutuv maailmas vajatakse üha rohkem inimesi, kes orienteeruksid hästi digimaailmas ning oleksid piisavalt pädevad selles vallas töötama [1]. Nõudlus infotehnoloogia (IT) haridusega ja sealhulgas ka programmeerimise oskusega inimeste järgi on jõudsalt kasvamas. Näiteks Eesti oli aastail 2009–2013 koos Taani, Rootsi, Soome ja Austriaga üks riikidest, kelle digipädevused olid kõige kiiremas arengus [2]. Et tööturule jõuaks piisavalt palju inimesi, kellel oleksid vajalikud oskused olemas ja kes saaksid aidata riigil veel edasi areneda, tuleks IT-ga tutvust teha juba varasest eest.

Hariduse ja Infotehnoloogia Sihtasutus (HITSA) eestvedamisel on loodud mitmeid IT valikkursuste materjale gümnaasiumile, nende seas näiteks ka programmeerimiskursus „Tehnoloogia tarbijast loojaks“, mis toetub digiõpikule „Programmeerimine“ [3]. Olemasolevaid õppematerjale tuleks aga uuendada, et toetaksid senisest enam õpilaste informaatikaõpinguid.

Soovitud eesmärkide täideviimiseks korraldas HITSA riigihanke nr 225368 „Gümnaasiumi informaatika valikkursuste arendus- ja koolitustegevus“, mille eesmärkideks on „valikkursuste edasiarendamine ning koolitus antud valikkursuste juhendajatele“ [4]. Tartu Ülikooli (TÜ) arvutiteaduste instituut tegi pakkumise hanke läbiviimiseks, et lisada gümnaasiumi informaatika kursuste juurde materjale nii, et see vastaks veelgi paremini õppurite vajadustele. 2020. aasta sügisel alustas informaatika didaktika töörihm oma tegevusega, millega ühines ka bakalaureusetöö autor.

Riigihanke läbiviimise käigus koostati käesolev lõputöö. Töö eesmärgiks on täiendada gümnaasiumi valikkursuse „Tarkvaraarendus“ uuendatud versiooni õppemooduliga, mis tutvustab graafilist kasutajaliidest PySimpleGUI. Lisaväärtust annab õpetajatele mõeldud ülesannete kogu, mida kasutades saab kontrollida ja kaardistada õpilaste teadmisi.

Bakalaureusetöö koosneb kolmest osast. Esimeses peatükis arutletakse programmeerimisoskuste vajalikkuse üle, kirjeldatakse informaatikaõppe hetkeolukorda nii maailmas üldiselt kui ka Eestis ning tutvustatakse olemasolevaid eestikeelseid programmeerimise õppematerjale. Ühtlasi selgitatakse käesoleva HITSA projekti tagamaid ning kirjeldatakse, miks oleks uusi kasutajaliidese materjale vaja luua ning miks valituks osutus just PySimpleGUI. Teises osas tutvustatakse digimaterjalide loomise eeskirju ning uute õppematerjalide loo-

mise protsesse. Samuti selgitatakse, kuidas koguti tagasisidet loodud õppemoodulile. Kolmandas osas tutvustatakse valminud materjale, mis on mõeldud kasutamiseks nii õpilastele kui ka õpetajatele. Loodud materjalide sisu kirjeldamisel keskendutakse eelmainitud õppe-
materjalide loomise eeskirjadele. Lisaks analüüsitakse tagasiside tulemusi ning kirjeldatakse potentsiaalseid edasiarenduse võimalusi. Lisades on toodud veebilingid loodud materjalidele.

1. Programmeerimise õpetamine ja graafilise kasutajaliidese loomine koolis

Järgnevas peatükis arutletakse selle üle, miks tänapäeval on digipädevused ning sealhulgas ka programmeerimisoskused tähtsal kohal ning miks neid tuleks hakata õpetama juba alg-, põhi- või keskkoolis. Samuti tutvustatakse erinevaid eestikeelseid programmeerimise õppe-materjale, mis on hetkel kättesaadavad ning kasutusel Eesti koolides. Seejärel kirjeldatakse TÜ informaatika didaktika töörühma projekti, mille käigus uus õppematerjal loodi. Kesken-dutakse ka sellele, miks valituks osutus just PySimpleGUI kasutajaliidese mooduli tutvus-tamine.

1.1 Programmeerimise õpetamise olulisus

Digipädevuste ning sealhulgas ka programmeerimisoskuste alla võib liigitada sellised või-med nagu „probleemide lahendamine, loogiline mõtlemine, raalmõtlemine ning disainios-kused“ [5:420]. Nüüdisajal on nimetatud oskused aina aktuaalsemad, sest üha enam kasuta-takse erinevates eluvaldkondades nutikaid lahendusi, mis nõuavad inimestelt vastavat kom-petentsi.

Tänapäeval pole enam haruldane, kui programmeerimise õpetamist alustatakse juba las-teaiaealistele lastele. Arfé jt [6] on 2020. aastal läbiviidud uuringus leidnud, et 8-tunnine programmeerimise harjutamine Code.org keskkonnas avaldab positiivset mõju alles 6-aas-tatele lastele – näiteks arenesid nende planeerimise ning probleemide lahendamise oskused.

Sarnaseid oskuseid saab õpetada näiteks ka robotitega. Praha Ülikool korraldas 2014. aastal projekti [7], kus 20 juhendajat õpetasid 79-le 5. klassi õpilasele robootikakursust, mis koos-nes 6 tsüklist (iga tsükkel kestis 5 tundi). Kursuse käigus tutvuti esmalt legokomplektiga LEGO Mindstorms NXT/WeDo, seejärel ehitati nendega nii lihtsamaid kui ka keerulise-maid legodest koosnevaid konstruktsioone ning lõpuks pidid õpilased tegema ka ühe suu-rema projekti. Uuringu tulemusena leiti, et robootika kasutamine teoreetilistele õpetuste kin-nitamiseks arendab õpilaste teadmisi tehnoloogiast ning programmeerimisest.

Programmeerimisoskuste alla võib lisaks eelmainitule liigitada ka algoritmide arusaamise ning nende kasutamise oskused, sest just tänu algoritmilisele mõtlemisele saab probleemile läheneda järk-järgult ja seeläbi ka loogilist mõtlemist arendada. Jizzakh Pedagoogiline Ins-tituut (*Jizzakh State Pedagogical Institute*) teostas 2020. aastal uuringu, mille põhjal tehti

järeldus, et algoritmide õppimine plokk skeemi meetodil ja nende põhimõtete mõistmine parandab õpilaste üldist arusaama erinevate struktuuride toimimisest ning tuleb neile kasuks igas eluvaldkonnas. Ühtlasi võiks algoritmide tutvustamisega alustada juba keskkoolis, kasutades lihtsamaid elulisi näiteid [8].

Keskkoolis õpetatakse nii algoritmidega kui ka muude informaatikaalaste teemadega seotud kursusi kohustuslikus või vabatahtlikus korras. Täpsemalt kirjeldatakse erinevaid kursusi ja õppematerjale, mis on muuhulgas mõeldud ka gümnaasiumiastmele, järgmises alapeatükis. Selliste kursuste õpetamine on oluline, sest kui pärast gümnaasiumi lõpetamist peab noor tegema valiku, kuidas oma eluteed jätkata, võiks üheks variandiks olla ka informaatikaga seotud eriala. Mainitud valdkonnas on tänapäeval rohkelt võimalusi ning just Eesti näitel tuleb see hästi välja - üheks põhjuseks, miks Eestis on nõudlus programmeerimisoskusega inimeste järgi niivõrd suur, on näiteks asjaolu, et tegu on riigiga, kus *startup*-ettevõtete arv on suurim inimese kohta [9]. Suur ettevõtete arv tähendab omakorda ka rohkelt töökohti, mistõttu vajatakse aina enam inimesi, kes oleksid pädevad antud valdkonnas töötama.

Hoolimata asjaolust, et igal aastal astub pea mitusada inimest ülikooli vastavat teaduskraadi omandama, on siiski puudus oma ala ekspertidest. IT ettevõtte Tieto Estonia AS tegevjuht Anneli Heinsoo on aga märkinud, et ettevõtjad teevad kõik selleks, et kaasata noori IT tegijaid ning ühtlasi toob ta välja, et eelkõige oodatakse tööle just tarkvaraarendajaid [10].

Järelikult võiksid infotehnoloogiast huvitatud noored programmeerimisega tutvust teha juba vähemalt gümnaasiumis, sest vajadus programmeerimisoskusega inimeste järgi on suur ning võimalusi selles valdkonnas on rohkelt.

1.2 Programmeerimise õpetamine Eestis

Tartu Ülikooli informaatika õppekava vilistlane Eleriin Rein on oma bakalaureusetöös „Eesti gümnaasiumites õpetatavad programmeerimise kursused” väitnud, et „informaatika on olnud osa koolide õppekavadest gümnaasiumiastmes ligi 50 aastat“ [11:7]. Selle aja vältel on arenenud nii õpetamise kui ka õppimise kvaliteet, materjalide vorm, õppijate arv ning loomulikult ka õpetatava sisu ja arvutid.

IT õpetamist ja selle ajalugu Eestis on kirjeldatud Wise'i tellitud raportis „IT oskuste arendamine Eesti koolides” [12]. Raportis on välja toodud, et esimest korda hakkas programmeerimise õpetamisega tegelema Ülo Kaasik praeguses Hugo Treffneri Gümnaasiumis (endine Tartu 1. Keskkool). Järgmised parkümmend aastat said vaid valitud koolid, kus olid

matemaatikaklassid, programmeerimise õpet, kuid kui 1980. aastatel muutusid arvutid üha populaarsemaks, hoogustus ka IT õpetamine. Alates sellest ajast on programmeerimist koolides suunanud riiklik õppekava, kuid seal ei ole kindlalt sätestatud, et seda teemat peaks kohustuslikus korras käsitlema [13].

Kori jt [12] andmeil saab aga väita, et Eestis eksisteerib rohkelt IT-teemalisi valikkursusi, mida teemast huvitatud õpilased saavad läbida, ning mida õpetatakse paljudes koolides. Huvi informaatikaalaste kursuste vastu on suur, sest näiteks Tartus asuvast üheksast gümnaasiumist ei õpetata mitte ühtegi informaatikaalast kursust vaid ühes, kus puudus piisav huvi teema vastu. Peamiselt õpitakse informaatikat või sellega seotud teemasid programmeerimiskeeles Python [11].

Tartu Ülikool on loonud ka mitmeid MOOCe ehk vaba juurdepääsuga e-kursusi, muu hulgas ka mitmeid informaatikaga seotud kursusi. Sellisel kujul pakutavate kursuste eesmärgiks on tutvustada tasuta kõigile huvilistele, sealhulgas gümnaasiumiõpilastele, erinevaid teemasid ja ülikoolis õpetatavat [14]. Kuna on ka leitud, et digitaalsete õppematerjalide kasutamine keskkoolis suurendab õppimise kvaliteeti ning tõhusust [8], siis veebipõhiste õpikute kasutamine programmeerimisega seotud oskuste õpetamiseks on igati õigustatud. Gümnaasiumiastme õpilastele on loodud järgmised veebipõhised kursused ja materjalid:

- MOOC Programmeerimisest maalähedaselt¹
- MOOC Programmeerimise alused²
- Tehnoloogia tarbijast loojaks³
- ProgeTiigri Pythoni materjalid⁴
- Teeme ise arvutimänge⁵
- Rakenduste loomise ja programmeerimise alused⁶
- Tartu Ülikooli programmeerimise õpik⁷

¹ <https://courses.cs.ut.ee/2019/progmaa/>

² <https://courses.cs.ut.ee/2018/eprogalused/>

³ <https://didaktika.cs.ut.ee/progttl/>

⁴ <https://courses.cs.ut.ee/t/pythonkoolis/>

⁵ <https://courses.cs.ut.ee/t/TIAM>

⁶ <https://e-koolikott.ee/oppematerjal/7364-Rakenduste-loomise-ja-programmeerimise-alused>

⁷ <http://progeopik.cs.ut.ee/>

Kuigi informaatikaga seotud kursusi ning teemasid, mida õpetatakse, on üsna mitu, siis 2017. aastal HITSA poolt tellitud info- ja kommunikatsioonitehnoloogia (IKT) uuringu [15] tulemusena selgus, et digioskuste õpetamist tuleks muuta. Uuringus kaardistati digioskuste õpetamise hetkeolukorda Eesti üldhariduskoolides ja lasteaedades ning kolm peamist järeldust, mis seal välja toodi, on järgmised:

1. „Digioskuste õpetamise korraldus Eesti üldhariduskoolides on ebahühtlane“ [15:5]. Näiteks on ligi 20% üldhariduskoolidest otsustanud juba I kooliastmes õpetada lastele digioskusi iseseisva aine kaudu, II ja III kooliastmes õpetatakse seda arvutiõpetuse/informaatika õppeaine näol juba rohkem. Samuti erineb õpetajate arusaam digivahendite kasutamise olulisusest, sest ligi veerand küsitletutest arvab, et selliste vahendite kasutamine ei peaks olema loomulik osa õppeainetest. Ühtlasi ei paku umbes kolmandik Eesti üldhariduskoolidest IKT huviringe, mis lisab samuti ebahühtlust digioskuste õpetamise korraldusse.
2. „Digivahendite (seadmed, keskkonnad ja tarkvara) ning digitaalsete õppematerjalide kättesaadavus ja kvaliteet on üks keskne takistus digioskuste õpetamisel nii õppetöös kui ka huvihariduse pakkumisel“ [15:7]. Seda illustreerivad näiteks faktid, et enam kui kolmandik õpetajaskonnast ei kasuta oma igapäevaelus üldse nutiseadmeid, mistõttu ei kasutata neid ka õpetamisel. Samuti tunnevad üle 30% õpetajatest, et puudu on kvaliteetsetest õppematerjalidest.
3. „Kuigi õpetajate ja õpilaste hoiakud digivahendite kasutamise suhtes õppetöös on valdavalt positiivsed, pole nende tegelik rakendumine digioskuste omandamisel ja nüüdisaegse õpikäsituse rakendumist soodustava vahendina veel realiseerunud“ [15:9]. Ligikaudu kolmandik õpetajatest tunneb, et nad pole piisavalt pädevad digioskuste edasiandmiseks, mistõttu pole neil võimalik või nad ei pea vajalikuks näiteks roboteid ehitada, veebilehti, mängu või rakendusi luua.

E. Rein on samuti oma uurimustöös leidnud, et informaatikaalaste kursuste läbiviimisel ilmnes probleem, et õpilased tunnevad, et hetkel kasutuses olevad materjalid ei toeta üksteist, mistõttu ei ole võimalik ühes suunas edasi areneda [11]. Tuginedes eelkõige IKT uuringu 2. järeldusele ning E. Reini bakalaureusetöö tulemusele võib väita, et informaatika ainekava uuendamine toetaks informaatika õpetamist koolides rohkem.

1.3 Gümnaasiumi informaatika uued valikkursused

HITSA eestvedamisel uuendati 2018.-2019. aastal gümnaasiumi informaatika ainekava, et see arendaks senisest enam nii raalmõtlemist (ingl *computational thinking*) kui ka disainmõtlemist (ingl *design thinking*). Uue ainekava loomisprotsessis osalesid Tallinna Ülikool, Tartu Ülikool ning Tallinna Tehnikaülikool [16].

Haridus- ja Noorteameti koduleheküljel on kirjas ainekava keskmeks oleva arendusprojekti tähtsus: „Ainekava keskmeks olev digilahenduse arendusprojekt paneb meeskonnana tööle erineva ettevalmistuse saanud õpilased. Õpilased asetatakse päriselulisse olukorda, kus igapäev tuleb täita oma roll ning samas pingutada koos ühise eesmärgi nimel” [17]. Seega on uue ainekava eesmärgiks arendada gümnaasiumiõpilaste IT-alaseid teadmisi võimalikult laiaulatuslikult, samal ajal sidudes õpitavat projektide kaudu päriseluga. Uus ainekava koosneb järgmistest õppematerjalidega varustatud informaatika valikkursustest:

- Programmeerimine⁸
- Tarkvaraarendus⁹
- Kasutajakeskne disain ja prototüüpimine¹⁰
- Tarkvara analüüs ja testimine¹¹
- Digiteenused¹²
- DigiTaru ehk digilahenduse arendusprojekt¹³

Samuti võivad gümnaasiumiõpilased osaleda ka varem loodud valikkursustel, milleks on Robootika ja mehhatroonika¹⁴, 3D-modelleerimine¹⁵, Geoinformaatika¹⁶ ning Arvuti kasutamine uurimustöös¹⁷. Uue kursusena loodi ka veel Küberkaitse valikkursus, mida õppides

⁸ <https://web.htk.tlu.ee/digitalaru/programmeerimine/>

⁹ <https://web.htk.tlu.ee/digitalaru/tarkvara2/>

¹⁰ <https://web.htk.tlu.ee/digitalaru/disain/>

¹¹ <https://web.htk.tlu.ee/digitalaru/testimine/>

¹² <https://web.htk.tlu.ee/digitalaru/digiteenused/>

¹³ <https://projektid.edu.ee/display/progetiiger/Digilahenduse+arendusprojekt>

¹⁴ <https://www.robbootika.ee/wp/kasulikku/materjalid/mehhatroonika-ja-robbootika-kursus/>

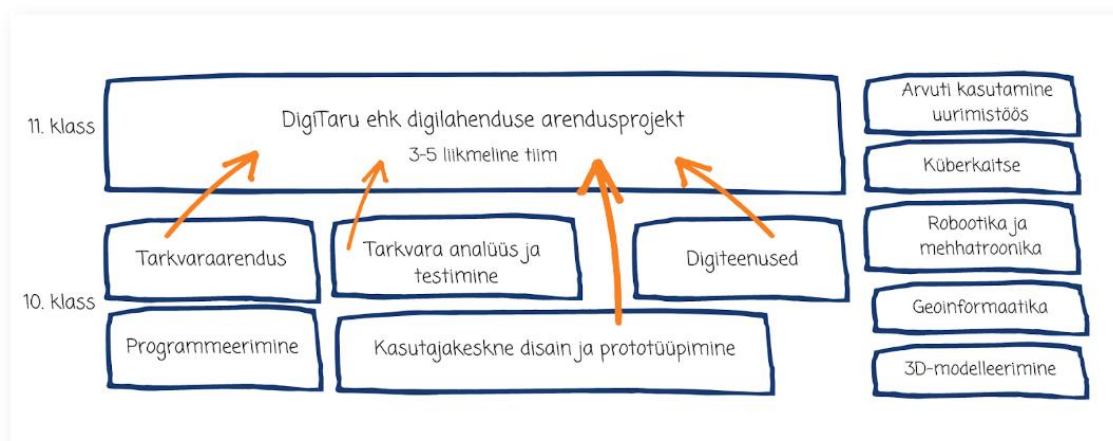
¹⁵ <http://vmg.v-maarja.ee/wp-content/uploads/2016/04/3D-modelleerimine-ja-printimine.pdf>

¹⁶ <https://e-koolikott.ee/oppematerjal/7365-Geoinformaatika>

¹⁷ <http://aku.opetaja.ee/>

on õpilastel võimalik saada teadmisi „küberkaitse olemusest ja distsipliinidest ning võimaldada neil omandada algteadmised antud valdkonnas” [18].

Võimalusi, kuidas õpetaja saaks kursuseid kombineerida, on mitmeid. Näiteks võib õpilastele pakkuda valikkursusi eraldiseisvatena ehk õpilane saab ise valida huvipakkuva teema. Võib järgida ka terviklikku kontseptsiooni, mille puhul saab õpilane valida 10. klassis 1-2 kursust ning seejärel osaleda Digilahenduse arendusprojekti valikkursusel 11. klassis. Kui õpilasel on vajalikud eelteadmised omandatud juba varem, võib ta osa võtta ainult DigiTaru projekti kursusest. Valikkursuste võimalikku paigutumist klassiti on kujutatud joonisel 1.



Joonis 1. Valikkursused gümnaasiumis [17].

HITSA korraldas 2020. aastal riigihanke „Gümnaasiumi informaatika valikkursuste arendus- ja koolitustegevus“, et olemasolevaid gümnaasiumi informaatika kursuste sisu veelgi täiendada ja uuendada. Kursuste kaasajastamist on vaja, sest eelmainitud info põhjal on selgunud, et õppematerjalide arendamine aitaks informaatikaga seotud teemade õpetamist tõhusamalt ellu viia ning toetada paremini õpilasi nende karjäärivalikutel. Tartu Ülikooli arvutiteaduse instituut tegi pakkumise valikkursuste „Tarkvaraarendus“ ja „Programmeerimine“ arendus- ning koolitustegevuste läbiviimiseks.

Projekti läbiviimiseks loodi Tartu Ülikooli informaatika didaktika töögrupp, millega liitus ka bakalaureusetöö autor. Vähesel määral hakati uuendama ka valikkursust „Programmeerimine“, kuid kuna enim kaasajastamist vajab gümnaasiumi valikkursus „Tarkvaraarendus“, siis peamiselt keskendus töörühm sellele. Eesmärgiks oli uuendada õppematerjale sellisel viisil, et need toetaksid paremini grupitööna tarkvaraprojekti läbiviimist. Töörühma ülesanne oli veebiõpikusse lisada järgmised moodulid [4]:

- Tarkvaraprojekt
- Veebirakenduse loomine. Flask.
- Andmete analüüs ja visualiseerimine - andmetöötlus mooduliga Numpy ning andmetöötlus mooduliga Pandas
- Graafilise kasutajaliidese loomine - kasutajaliides PySimpleGUI ning Tkinter

Bakalaureusetöö autori ülesandeks sai luua graafilist kasutajaliidest PySimpleGUI tutvustav moodul, sest kasutajaliidese teemat ei ole veebiõpikutes palju käsitletud. Samuti ei toeta olemasolevad eestikeelsed materjalid õpilasi piisavalt, et nad oleksid materjali läbinuna piisavalt pädevad oma lõpuprojekti ellu viima. Uus lisamoodul graafilisest kasutajaliidese loob ühtlasi lisaväärtust nende gümnaasiumiõpilaste jaoks, kes huvituvad näiteks veebidesainist ja kasutajaliidese arendamisest.

1.4 Graafiline kasutajaliides

Graafiline kasutajaliides (ingl *Graphical User Interface* ehk GUI) loodi 1970. aastate lõpus Xerox Palo Alto laboratooriumi poolt ning seda kasutati esmalt Apple Macintoshi ning Microsoft Windows operatsioonisüsteemides [19]. Abdo A.M. jt [20] andmeil tekkis vajadus GUI loomise järele, sest tekstipõhine käskluste edastamine arvutile ei olnud piisavalt efektiivne ega kasutajasõbralik. GUI kaudu saab kasutaja arvutile käsklusi anda aga läbi erinevate vidinate (ingl *widgets*), mis on paigutatud graafikaaknasse. Seega teeb graafilise kasutajaliidese olemasolu programmi kasutamise kasutaja jaoks lihtsamaks, mistõttu võib seda pidada sillaks kasutaja ja programmi vahel.

1.4.1 Kasutajaliidese olulisus tarkvaraprojekti

Lisaks sellele, et (tehnoloogia)ettevõtetes töötaksid tarkvaraarendajad, kes suudaksid valitud toodet või teenust võimalikult professionaalselt arendada, on tänapäeval nii alustavate kui ka juba kogunud ettevõtete edu üheks võtmekohaks see, missugune on näiteks firma teenuse või kodulehekülje graafiline kasutajaliides [20]. Kasutajaliidese intuiitiivsus, kasutajasõbralikkus ja kasutamislihtsus on liidese põhilisteks omadusteks [19], mis võivad oluliselt mõjutada kleidi arvamust. Graafilise kasutajaliidese kaudu on ühtlasi suurem võimalus jõuda suurema publikuni, sest põhiline osa rahvastikust eelistab arvutile käskluste andmist visuaalsel ja praktilisel, mitte tekstipõhisel viisil.

GUI abil saab kasutaja anda arvutile käsklusi erinevate elementide (ingl *widgets*) kaudu – näiteks vajutades mingile nupule, suunates kursori teatud kohta või lohistades elementi [19].

Tänapäeval kasutatakse aina enam ka näiteks näo- või helituvastuse funktsioone. Vastavalt tehtud sündmusele reageerib arvuti ning viib läbi ettenähtud tegevuse (ingl *event handling*). Et kõik osad töötaksid eesmärgipäraselt ja korrektselt, nõuab graafilise kasutajaliidese loomine tegijalt nii heatasemelist programmeerimisoskust kui ka loovust ja empaatiavõimet. Liidese loomist on hea alustada näiteks programmeerimiskeeles Python, mis on peatüki 1.2 andmeil peamiseks keeleks, mida käsitletakse Eesti gümnaasiumites.

1.4.2 Kasutajaliides Pythoni näitel

Programmeerimiskeeles Python saab graafilise kasutajaliidese loomiseks kasutada mitmeid raamistikke (ingl *frameworks*) või mooduleid. Neid on erinevat tüüpi: on sellised, mis juba kaasnevad Pythoni standardteegiga, ning sellised, mida saab kasutada erinevate platvormide integratsioonil [21]. Näitena kirjeldatakse kolme peamist moodulit, sealhulgas ka bakalaureusetöö raames loodavate õppematerjalide teemaks olevat moodulit PySimpleGUI.

Üheks kasutajaliidese raamistikuks keeles Python on TkInter või tkinter (ingl *Tk Interface*). See kuulub keele Pythoni standardsete moodulite hulka, seega ei pea seda eraldi alla laadima. Seega on Tkinter üks enim kasutatavatest moodulitest, mistõttu leiab internetist arvukalt juhendeid ja õpetusi mainitud raamistiku kasutamiseks. Tkinteri peamiseks funktsioonideks on graafikaaknas erinevate elementide kuvamine ning sündmuste käsitlemine [22]. Moodulit kasutades saab elemente aknas kuvada näiteks maatriksi (ingl *grid*) loogikat kasutades, aga ka koordinaatidega.

Kasutajaliidese mooduliks on ka EasyGUI, mis on üks kõige algelisematest moodulitest. Selle abil lihtsa kasutajaliidese loomine ei nõua tegijalt suuremahulisi teadmisi näiteks elementide funktsioonidest ega ka tkinterist. Küll aga on hetkel kasutatav vaid 2014. aastal loodud versioon 0.97, seega moodulit EasyGUI pole rohkem uuendatud [23].

Graafilist kasutajaliidest saab programmeerimiskeeles Python luua ka vabavaralise mooduli PySimpleGUI abil. Nimetatud moodul teisendab raamistikud tkinter, Qt, WxPython ja Remi lihtsamaks mooduliks ning see töötab Pythoni versioonidel 2.7 ja 3.7. Sarnaselt tkinteriga töötab PySimpleGUI erinevatel operatsioonisüsteemidel ning selle teek põhineb samuti teegil Tk, mida kasutatakse laialdaselt ka teistes programmeerimiskeeltes. PySimpleGUI on ühtlasi aina enam populaarsust kogumas, mis oli ka üheks selle mooduli kajastamise põhjuseks veebiõpikus [24].

PySimpleGUI erinevus moodulist EasyGUI on see, et PySimpleGUI puhul toimub programmi interaktsioon kasutajaliidesega läbi sündmustele reageerimisele (ingl *event-driven*) *while*-tsükliks [24], kuid teisel juhul toimub see läbi funktsioonide käivitamise [23]. See võimaldab aga luua rohkem erinevaid sündmusi, mis omakorda soodustab võimekamate, aga ka keerukamate kasutajaliideste tegemist. PySimpleGUI erineb moodulist tkinter aga kõige enam selle poolest, et see vajab sama eesmärgi elluviimiseks lühemaid koodiridu [24].

Seega võib väita, et PySimpleGUI on piisavalt lihtne algajale, aga ka piisavalt võimalusterohke edasijõudnule, et igäüks saaks graafilist kasutajaliidest luua just vastavalt enda võimetele ja eesmärkidele

1.5 Õppematerjalide koostamise eeskirjad

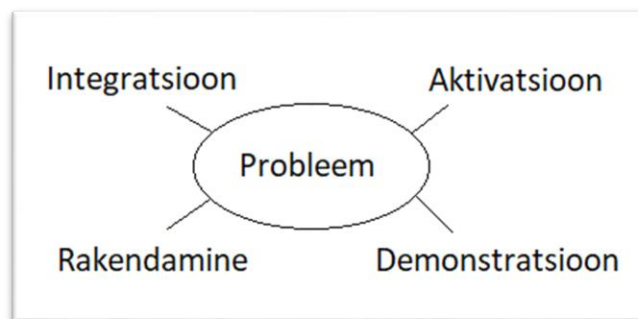
Õppematerjalide koostamisel tuleb silmas pidada erinevaid soovitusi ja eeskirju, et loodav sisu toetaks õppijate arengut ja uued teadmised kinnistuksid võimalikult efektiivselt.

Järgnev osa on refereeritud M. David Merrilli [25] uuringute tulemustest, kus ta arutleb selle üle, et kuidas läheneda õppimisele probleemipõhiselt ja miks sellisest viisist on kõige rohkem kasu. Õppida saab efektiivselt siis, kui õpetatava sisu koostamisel ja edasiandmisel võetakse arvesse järgmisi põhimõtteid:

- 1) Merrilli mudeli keskmeks on põhimõte, et õppija omandab kõige paremini uusi teadmisi olukorras, kus õpetatavale sisule lähenetakse läbi probleemide või ülesannete lahendamise (**probleem**, ingl *problem*). Üheks viisiks on see, et juba enne õpetamist tuleb tuua mõni suurem probleem näiteks, mida õppija suudab pärast materjali läbitöötamist lahendada või luua. Selline lähenemine tagab õpilaste parema orienteerituse õppematerjalides võrreldes olukorraga, kus õpieesmärgid on näiteks sõnastatud eraldi ja ühekaupa.
- 2) Teise põhimõttena toob Merrill välja selle, et õpitavat sisu on kergem omandada, kui see toetub juba olemasolevatele teadmistele (**aktivatsioon**, ingl *activation*). Seega on õpilasel juba olemas teatud alus ja teadmiste pagas, mis teeb uue materjali õppimise lihtsamaks. See omakorda tõstab motivatsiooni võõra sisu omandamiseks. Samas, kui õppija näiteks ei mäleta varemõpitud, tuleb õppematerjali koostajatel keskenduda ka sellele ja vajadusel olulist meelde tuletada või nendele kohtadele viidata.

- 3) Järgmisena tuleks õppematerjali koostades silmas pidada printsiipi, et õpitava demonstreerimiseks tuleks materjali lisada illustreerivaid näiteid (**demonstratsioon**, ingl *demonstration*). Visuaalselt atraktiivsed näited aitavad õppimisele kaasa, kuid näidete koostamisel tuleb aga arvestada, et need ei tohi olla liiga kirjud ega koosneks liiga erinevatest osadest (näiteks tekst ja graafika). Sellisel juhul võivad need õppurilt liigselt tähelepanu röövida, mistõttu ei täida need oma eesmärki.
- 4) Neljanda punktina mainib Merrill, et õpilased peaksid õpitud oskuste demonstreerimiseks oskama lahendada uusi probleeme või ülesandeid, et saadud teadmisi rakendada (**rakendamine**, ingl *application*). Siinkohal tuleb aga jälgida, et ülesande sisu toetuks õpitud materjalile. Samuti peab tehtud ülesandele järgnema asjalik ning informatiivne tagasiside, et õppija saaks aimdust võimalikest kitsaskohtadest.
- 5) Viimasena on välja toodud see, et õppimise motivatsioon on suurem, kui õpetamisel toetatakse omandatud oskuste või teadmiste demonstreerimist ning integreerimist tavaellu (**integratsioon**, ingl *integration*). Sellisel juhul ei omandata oskusi pelgalt mitte õppimise eesmärgil, vaid nende kasulikkus väljendub ka inimese tavaelus ning neid saab kasutada ka isiklikel eesmärkidel.

Seega võib nimetatud viite punkti kokkuvõtvalt illustreerida joonise 2 abil.



Joonis 2. Merrelli mudel. Koostatud allika [25] põhjal.

Eelpool toodud põhimõtteid saab edukalt kasutada nii tava- kui ka veebiõpikute koostamisel. Samas tuleb e-õppematerjalide loomisel silmas pidada ka just sellele valdkonnale omaseid printsiipe. Alljärgnev materjal toetub HITSA poolt koostatud digitaalsele õppematerjali loomise juhendile, kus kirjeldataksegi veebipõhiste materjalide loomise printsiipe. Siinkohal mõistetakse digitaalse õppematerjali all „digitaalsel kujul levitatavat õppeotstarbelist materjali“ [26]. See võib olla näiteks kas audio- või videoloeng, mõni ülesanne või test. Digitaalset õppematerjali iseloomustab ka selle interaktiivsus, mille

tähtsus oleneb nii õpetatava sisust kui ka eesmärkidest. E-õpikud või muud digiõppematerjalid võivad sisaldada erineval kujul tekste ning multimeedia elemente, mis peaksid õpetatavat illustreerima.

Kvaliteetset õppematerjali saab mõõta erinevate karakteristikute põhjal. HITSA juhendis on seda tehtud Leacock jt välja töötatud hindamismudeli LORI (*Learning Object Review Instrument*)¹⁸ põhjal. Nimetatud mudeli järgi on digitaalne õppematerjal kvaliteetne siis, kui see on:

- **„õppimist toetav**: õppematerjal vastab sihtrühma vajadustele, on loodud kindla eesmärgiga, sobivas mahus ja selles on sõnastatud õpitulemused, mida õppijaid suunatakse tõhusalt omandama;
- **sisult kvaliteetne**: õppematerjal moodustab sisulise terviku, on ainealaselt ja keeleliselt korrektne;
- **motiveeriv**: õppematerjal on õppija jaoks kaasav, nii raskusastmelt kui ka sisult eakohane, arvestab õppija eelteadmisi ja toetab õpioskuste arendamist;
- **kohandatav**: õppematerjal sobib kasutamiseks erinevates õpiolukordades ja erineva taustaga õppijatega;
- **interaktiivne**: õppematerjal võimaldab õppijal ise juhtida selle kasutamist ning saada õppimisele tagasisidet;
- **autoriõigusi järgiv**: õppematerjal järgib autoriõiguse seadust (AutÕS, 2015)¹⁹, sh sisaldab informatsiooni autori(te) kohta, teiste autorite materjalidele on korrektselt viidatud, soovitatavalt on lisatud kasutamistingimused (nt Creative Commons'i litsents);
- **kasutajasõbralik**: õppematerjal on liigendatud, visuaalselt köitev, intuitiivselt navigeeritav ja sobib ka erivajadustega õppijale;
- **tehniliselt korrektne ja ühilduv**: õppematerjal on tehniliselt universaalne, seda on võimalik kasutada levinumate operatsioonisüsteemide, tarkvarade ja seadmetega;
- **leitav**: õppematerjal on avalikustatud ja varustatud metaandmetega.“ [26].

Järgmises peatükis tutvustatakse protsessi, kuidas valmisid nii graafilise kasutajaliidese õppemoodul kui ka ülesannete kogu.

¹⁸ <https://www.jstor.org/stable/pdf/jeductechsoci.10.2.44.pdf>

¹⁹ <https://www.riigiteataja.ee/akt/128122011005?leiaKehtiv>

2. Metoodika

Selles peatükis selgitatakse uue õppematerjali loomise protsessi, tuginedes eelpool kirjeldatud õppematerjali loomise eeskirjadele. Samuti tutvustatakse seda, kuidas ja kellelt koguti tagasisidet.

2.1 Õppematerjali koostamise protsess

Õppematerjali hakati koostama 2020. aasta septembri teises pooles. Tartu Ülikooli informaatika didaktika töörühmaga arutati läbi, milliseid teemasid tuleks olemasolevatesse digiõpikusse „Tarkvaraarendus“ lisada, et seda veelgi mitmekesisemaks ja tarkvaraarenduse projekte toetavamaks muuta. Et valituks osutus ka uus õppemoodul kasutajaliidesest PySimpleGUI, asus bakalaureusetöö autor teemaga tegelema. Paralleelselt õppematerjalide koostamisega hakati järk-järgult ülesannete kogusse ülesandeid lisama ja gümnaasiumis katsetama.

Esmalt tutvuti juba olemasolevate materjalide ja teemadega, et teada saada, milliseid oskusi on õpilane digiõpikute abil juba omandanud. Gümnaasiumiõpilane, kes on osalenud valik-kursustel „Programmeerimine“ ja „Tarkvaraarendus“ ning need õppematerjalid edukalt läbi töötanud, peaks valdama järgmisi programmeerimisega seotud teemasid:

- erinevad andmetüübid - sõned, täisarvud, ujukomaarvud, tõeväärtused;
- muutujate olemus ja defineerimine;
- kasutajalt sisendi küsimine - *input()*;
- tingimuslause, tingimuslause tingimuslause sees - *if, else, elif*;
- *while*-tsüklid;
- järjendid, kahemõõtmelised järjendid;
- funktsioonid.

Seejärel tehti kindlaks ja lepiti kokku, millises vormi ja struktuuriga õppematerjalid koostada. Eelmises peatükis kirjeldatud LORI mudeli kohaselt tuleb õppematerjali ja ülesandeid koostades järgida ühtset joont ja seda stiili püüti jätkata ka uues õppematerjalis. Sellega tagatakse materjalide ühtsus, mis toetab nii õpilaste kui ka õpetajate õppematerjali kasutamismugavust ning ka nõuetele vastavust. Näiteks õpilaste kaasamiseks on ülesannete kirjeldamisel on kasutatud meie-vormi (joonis 3).

Varem oleme kasutanud peamiselt järjendeid, mille elemendid on arvud või sõned. Tegelikult võime järjendis hoida igasugust tüüpi väärtuseid, kusjuures järjendi elemendid võivad Pythonis isegi eri tüüpi olla:

```
huvitav_jarjend = ['Taisi Telk', 29, True]
```

Joonis 3. Kuvatõmmis veebiõpiku „Tarkvaraarendus. 2. trükk“ 3. peatükist [27].

Teemade ja peatükkide planeerimisel tuli võtta arvesse LORI mudeli seisukohta, et loodav digitaalne õppematerjal peab olema õpilast toetav. Seetõttu pidi töö autor silmas pidama, et õpetatava sisu toetuks varemõpitule, muutuks järk-järgult keerulisemaks ning iga peatüki lõpuks oleks õpilane omandanud mõne uue väärtusliku teadmise või võtte. Ühtlasi tuli juba enne kirjutamist teadvustada seda, et uut infot tuleb õpilastele edastada väikeste osade kaupa, et see oleks samm-sammuline ja kergemini hoomatav. See tõstab õpilaste motivatsiooni õppida, mis on samuti LORI mudeli kohaselt oluline punkt õppematerjali loomise juures.

Pärast seda, kui töö autor oli ka ise mooduliga PySimpleGUI erinevaid katsetusi teinud ja internetis leiduvaid materjale uurinud, otsustati digiõpiku moodulisse luua viis peatükki. Õppematerjali sisu loodi algselt veebipõhisesse tekstiredaktorisse Google Docs [28], mis võimaldab kasutajal valminud teksti igal ajal igas seadmes muuta. Valitud viis peatükki, mille sisu hakati tekstiredaktorisse kirjutama, hõlmavad endas olulisemaid aspekte, mida omandades on gümnaasiumiõpilasel võimalik luua efektiivne ning sisukas kasutajaliides oma programmile. Uue õppemooduli osad on järgmised:

- 1) PySimpleGUI ja esimene näide.
- 2) PySimpleGUI nupuvajutus, hüpikaken ja uue graafikaakna avamine.
- 3) PySimpleGUI graafikaakna vidinad.
- 4) PySimpleGUI suurem näide.
- 5) LISA: PySimpleGUI ja joonistamine.

Õppematerjali sisu, nende loomise protsessi ning ka ülesannete kogus olevaid ülesandeid kirjeldatakse üksikasjalikumalt töö 3. osas.

Tööprotsessi käigus toimusid töörühmaga iganädalased veebikoosolekud, kus arutleti teemakohaste probleemide üle, ühtlustati õpiku stiili ning räägiti, kuidas kellelgi töö on kulgenud. Graafilise kasutajaliidese õppematerjali esimene täispikk versioon valmis 2020. aasta

novembri keskpaigas. Kuna selleks ajaks olid kõik töörühma liikmed oma osaga valmis saanud, katsetati ka üksteise materjale. Seejärel viidi vastavalt tagasisidele sisse viimased muudatused ning 2020. aasta detsembris pandi kogu õppematerjal üles „Tarkvaraarendus. 2. trükk.“ veebiõpikusse, mis asub Pressbooks²⁰ keskkonnas.

2.2 Tagasiside kogumine

Bakalaureusetöö autor sai loodud materjalidele tagasisidet juba töö käigus. Informaatika didaktika töörühma liikmed kommenteerisid tehtud tööd veebikoosolekul või tekstiredaktoris Google Docs. Põhjalikum tagasiside ühelt töörühma liikmelt, kes on ka tegev informaatikaõpetaja, saadi pärast seda, kui oli valminud esimene versioon õppematerjalidest.

Õpilastelt sai töö autor tagasisidet juba nii õppematerjalide koostamise käigus kui ka pärast nende valmimist. Kommentaare koguti kahe gümnaasiumi õpilastelt. Ühes gümnaasiumis läbis 36 õpilast 2020. aasta septembrist oktoobrini „Tarkvaraprojekti” kursust, mille raames said huvilised katsetada ka uusi graafilise kasutajaliidese õppematerjale. Teine gümnaasium alustas uute materjalide katsetamisega 2020. aasta novembris, mil õpilased ja õpetajad said kasutamiseks juba parandatud materjalid. Katsetamise jaoks loodi eraldi Google Docs fail õppematerjalidega, kuhu said nii õpetajad kui ka õpilased enda kommentaare lisada.

Uut õppemoodulit graafilise kasutajaliidese moodulist PySimpleGUI tutvustati ka informaatikaõpetajate konverentsil 14.–16. jaanuaril 2021. aastal. Konverentsil toimusid erinevad ettekanded uute informaatika õppematerjalide ning e-õppe teemadel. Ürituse eesmärgiks oli vahetada mõtteid ja informatsiooni teemast huvitatud inimeste vahel, et soodustada informaatika õpetamist [29].

²⁰ <https://pressbooks.com/>

3. Tulemused

Selles peatükis antakse ülevaade töö autori poolt koostatud õppemoodulist „PySimpleGUI“ (lisa 1), mis kuulub nüüd uuendatud informaatika valikkursuse „Tarkvaraarendus. 2. trükk.“ digiõpiku [30] lisamoodulite hulka. Samuti tutvustatakse loodud ülesannete kogu (lisa 2). Seejärel keskendutakse nii õppemooduli kui ka ülesannete kogu sisule konkreetsete näidete varal, toetudes Merrilli mudelile. Viimasena analüüsitakse saadud tagasisidet ning arutletakse potentsiaalsete edasiarenduse võimaluste üle.

3.1 Valminud õppematerjalid

Uuendatud veebiõpiku uus õppemoodul „PySimpleGUI“ on loodud *Creative Commons Attribution 4.0 International Licence*²¹ alusel. Õppemoodul koosneb viiest alapeatükist, millest üks on lisapeatükk. Eelkõige on õpetatava sisuks erinevate komponentide, kujunduste ja funktsioonidega graafikaakende loomine eesmärgiga anda õpilastele uusi teadmisi rühmatööna valmiva projekti elluviimiseks.

Õppemooduli ja ülesannete läbimiseks on arvestatud 10 tundi ning see on läbitav iseseisvalt või koolitunnis õpetaja juhendamisel. Koostatud õppematerjali sisu inspiratsiooniks oli põhiliselt ingliskeelne Pythoni dokumentatsioon moodulist PySimpleGUI, mis tutvustab üksikasjalikult mooduli võimalusi [24]. Modifitseeritud õppematerjal ja ülesannete kogus olevad ülesanded on bakalaureusetöö autori enda looming.

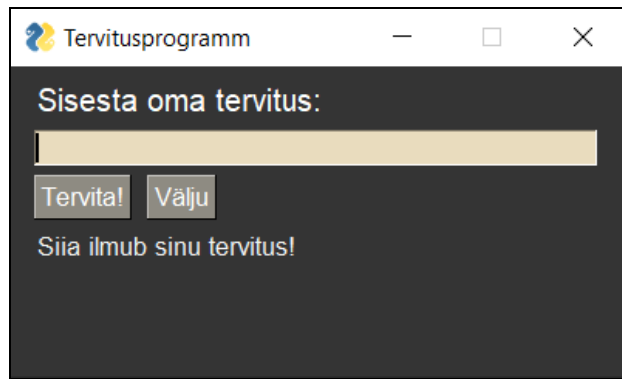
3.1.1 PySimpleGUI ja esimene näide

Õppemooduli esimene peatükk „PySimpleGUI ja esimene näide“²² algab graafilise kasutajaliidese loomise vajaduse selgitamisest. Lühidalt kirjeldatakse ka seda, kuidas moodulit PySimpleGUI paigaldada.

Esimene peatükk toetub ühele näitele (joonis 4), mille põhjal saab õpilane omandada esimesed oskused, mida graafikaakent luues peaks teadma. Lisaks akna loomisele tutvustatakse ka teksti, tekstisisestuskasti ning lihtsamate nuppude loomist, et akent interaktiivsemaks muuta.

²¹ <https://creativecommons.org/licenses/by/4.0/>

²² <https://web.htk.tlu.ee/digitaru/tarkvara2/chapter/pysimplegui-sissejuhatus/>



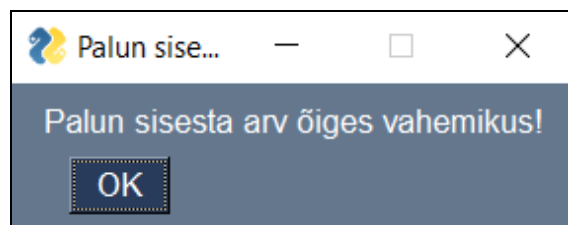
Joonis 4. Kuvatõmmis loodud õppematerjali 1. peatükist (lisa 1).

Et õpilane saab ise aknale ka värviteema valida, on digiõpikusse toodud vastav viide erinevate teemade sirvimiseks. Oluliseks osaks graafikaakna loomise juures on ka programmi toimimine vastavalt sellele, missuguseid tegevusi kasutaja aknas läbi viis – esimese näite puhul saab kasutaja sisestada tekstisisestuskasti tervituse ning nupule vajutades kuvatakse see nuppude all tekstisildil. Õpilane tutvub samm-sammult sellega, kuidas programm toimib vastavalt kasutaja tegevusele. Peatüki lõppu on kuvatud terve programmikood koos abistavate kommentaaridega.

3.1.2 PySimpleGUI nupuvajutus, hüpinkaken ja uue graafikaakna avamine

Teises peatükis²³ näidatakse võtteid, mis võimaldavad programme veelgi interaktiivsemaks muuta. Näiteks seletatakse, kuidas integreerida funktsioon programmi nii, et nupule vajutades arvutatakse selle funktsiooni abil etteantud arvu ruut.

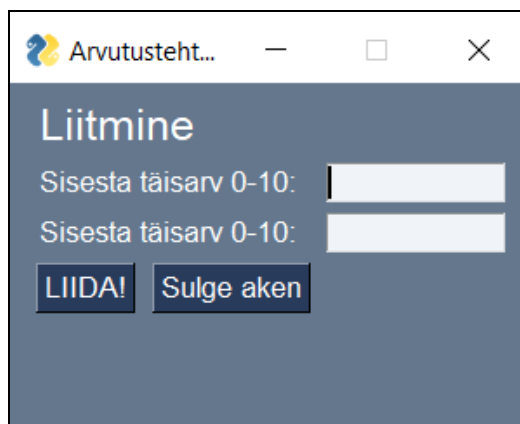
Seejärel tutvustatakse uut elementi, hüpinkakent (ingl *pop-up window*). Selle abil saab näiteks eelnevasse arvu ruudu arvutamise programmi lisada lisatingimuse, et kui sisestatav arv pole nõutavas vahemikus, kuvatakse ekraanile veateatega hüpinkaken (joonis 5).



Joonis 5. Kuvatõmmis loodud õppematerjali 2. peatükist (lisa 1).

²³<https://web.htk.tlu.ee/digitaru/tarkvara2/chapter/pysimplegui-nupuvajutus-pop-up-aken-ja-uee-graafikaakna-avamine/>

Kui programm peab aga järgmises aknas ka teatud arvutusi või muid operatsioone tegema, tuleb avada järgmine graafikaaken. Funktsiooni kasutamist ning hüplikakna ja uue graafikaakna loomist illustreeritakse näitega, kus esimeses aknas saab kasutaja sisestada kaks täisarvulist liidetavat vahemikus 1–10 (joonis 6).



Joonis 6. Kuvatõmmis loodud õppematerjali 2. peatükist (lisa 1).

Kui sisestatud arvud sobivad, kuvatakse pärast nupule “LIIDA!” vajutamist ekraanile järgmine aken, kuhu kasutaja saab sisestada lahutatavad nõutud vahemikus. Kui sisestatud arvud ei vasta nõuetele, ilmub ekraanile vastava teatega hüplikaken. Sobivate arvude korral kuvatakse viimases aknas arvutustehete vastused.

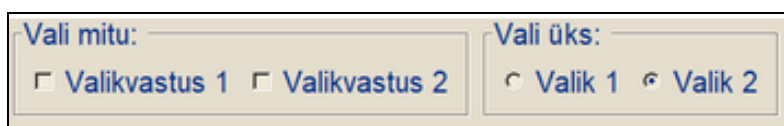
3.1.3 PySimpleGUI graafikaakna vidinad

Kuna eelmistes peatükkides on vidinatest käsitletud vaid “*Submit*” ja “*Cancel*” nuppe ning tekstisisestusvälja, siis peatüki „PySimpleGUI graafikaakna vidinad“²⁴ eesmärgiks on erinevate graafikaakna komponentide ehk vidinate (ingl *widgets*) ja võtete tutvustamise jätkamine lisaks eelmainitule. Õppematerjali 3. peatükis tutvub õpilane seega mitmete variantidega, kuidas oma programmile veelgi mitmekesisemat kasutajaliidest luua.

Esmalt antakse ülevaade erinevatest nuppudest. Näiteks käsitletakse mitmikvalikuvõimaluse nuppe, sealhulgas raadionuppu ja märkeruutu. Graafikaaknasse saab kuvada ka erilaadi menüüsid, näiteks rippmenüüd ning valikute kasti, liugurit ning suuremat tekstisisestusvälja, mis võimaldab kasutajal sisestada pikemaid tekste.

²⁴ <https://web.htk.tlu.ee/digitaru/tarkvara2/chapter/pysimplegui-graafikaakna-vidinad/>

Tihti on graafikaaknas vaja teatud infot muust eraldada, et muuta aken visuaalselt paremini hoomatavaks. Siinkohal saab kasutada elementide paigutamist raami, mille kohta on samuti digiõpikus näide toodud (joonis 7).



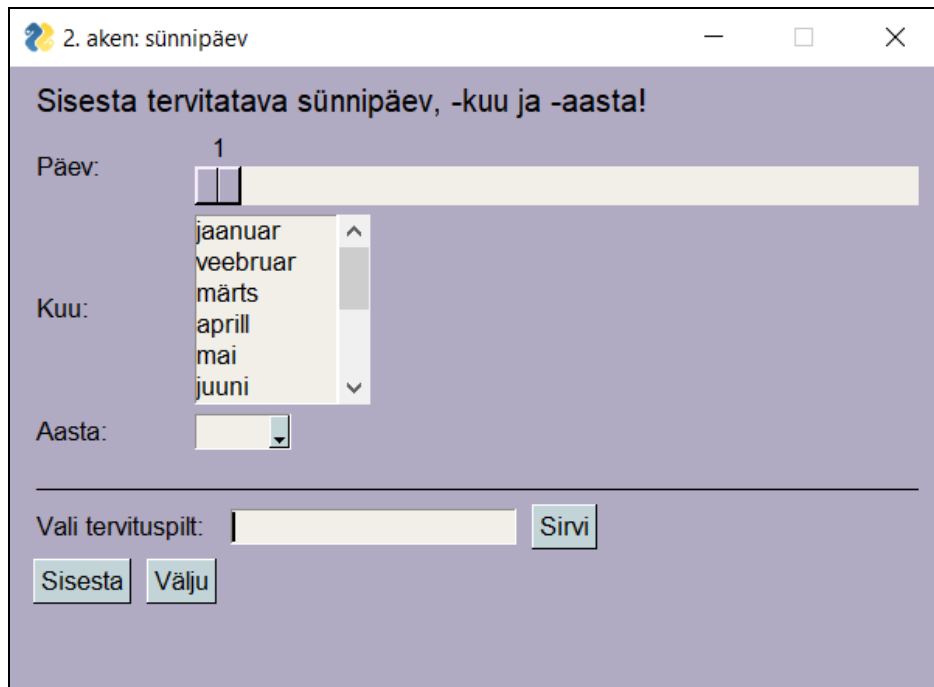
Joonis 7. Kuvatõmmis loodud õppematerjali 3. peatükist (lisa 1).

Kui kasutajal on programmis vaja kasutada mingit faili, saab ka selle graafikaaknas vastavale nupule vajutades avada. Digiõpikus on kirjeldatud ka olukorda, kui failipuust tuleks eraldada faili nimi. Kuna selleks võib olla näiteks pilt, mida tahetakse graafikaaknas kuvada, on sellegi kohta näide toodud. Sobivaid pildiformaate on kirjeldatud täpsemalt graafilise kasutajaliidese õppemooduli 5. peatükis.

3.1.4 PySimpleGUI ja suurem näide

Neljandas peatükis „PySimpleGUI suurem näide“²⁵ võetakse seniõpitud teemad kokku, tuues näiteks ühe suurema programmi, mis hõlmab kõiki võtteid, mida on eelmistes peatükides tutvustatud. Valituks osutus tervituskaardi loomise ülesanne, mille käigus peab ekraanile kuvama üksteise järel kolm graafikaakent nõutud elementidega. Töö autor on digiõpikusse lisanud koodi näitelahendusega, et õpilane saaks koodi jooksutada ja aru saada, kuidas üks või teine element toimib. Näidislahenduse põhjal loodud programmi 2. graafikaakent kuvatakse joonisel 8.

²⁵ <https://web.htk.tlu.ee/digitalu/tarkvara2/chapter/pysimplegui-suurem-naide/>



Joonis 8. Kuvatõmmis loodud õppematerjali 4. peatükist (lisa 1).

Näidislahendus jätab aga piisavalt ruumi ja vabadust graafikaakende isikupärastamiseks. Nii saab õpilane sinna ka iseseisvalt vabalt valitud elemente lisada või kujundust muuta.

3.1.5 LISA: PySimpleGUI joonistamine

Digiõpiku mooduli viimane peatükk joonistamisest otsustati tuua õppematerjali hulka lisa-peatükina, sest antud teema ei pruugi kõikide õpilaste jaoks suurt väärtust pakkuda, aga olenevalt huvidest võib see mõne jaoks osutuda oluliseks.

Peatükk algab joonistamisele suunatud graafikaakna paigutuse põhimõtete selgitamisest, sealjuures tuletatakse meelde ka koordinaatteljestiku teemat, sest erinevaid elemente saab graafikaaknasse paigutada koordinaate kasutades.

Lisaks kujundite, näiteks ringi, sirgjoone ning ruudu lisamisele õpetatakse määrama elementidele erinevaid värve ning graafikaaknasse seadma ka taustapilti. Lisafunktsioonidena on õpikus välja toodud elemendi värvi ning asukoha muutmise võimalused, mis võimaldavad õpilasel oma joonistusi interaktiivsemaks muuta. Peatüki lõpus on mõned edasiarenduse võimaluse mõtted, kuidas õpilane saaks iseseisvalt loodud joonistust veelgi interaktiivsemaks muuta (joonis 9).

Katseta ja vaata, mis juhtub! Püüa pildile veel lisada (liigutatavaid) objekte, näiteks lisa veel juurde aknaid, korsten, katuse kaunistused, uksele aken (selle jaoks saad kasutada **sg.DrawOval**) jne. Näiteks võiks lõpptulemus välja näha selline:



Joonis 9. Kuvatõmmis loodud õppematerjali 5. peatükist (lisa 1).

Järgnevalt tutvustatakse ülesannete kogu, mis valmis paralleelselt õppemooduli materjalide loomisega.

3.2 Valminud ülesannete kogu

Ülesannete kogu on mõeldud õpetajatele, et nad saaksid õpilastele anda õppemoodulist PySimpleGUI saadud teadmisi kinnistavaid peale iga peatüki läbitöötamist. Ülesanded on kogutud Google Docs dokumenti, millele saab õpetaja parooli olemasolul ligipääsu läbi valikkursuse „Tarkvaraarendus. 2. trükk.“ veebilehe.

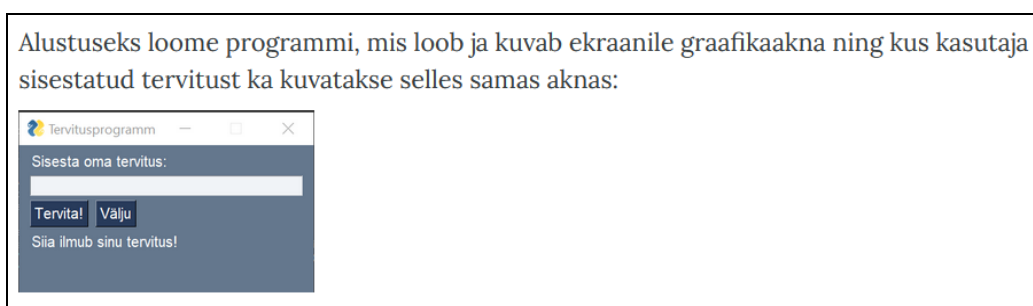
Iga peatüki kohta on toodud vähemalt üks autori poolt loodud ülesande variant, mida õpilane peaks pärast peatüki läbitöötamist oskama lahendada. Samuti on dokumendis näidatud ka üks võimalikest ülesande lahendamise versioonidest, mida kasutades saab õpetaja anda õpilasele tagasisidet.

Kuna õpilased peavad valikkursuse lõpetamiseks tegema ka suurema lõpuprojekti, siis on ka selle teema kohta toodud paar näidet, mida võiks õpilane luua, kui ta soovib oma programmile kasutajaliidest lisada. Siinkohal on näidislahendus toodud ühele variandile, milleks on kalkulaatori loomise ülesanne.

3.3 Materjalide koostamine Merrilli mudeli põhjal

Bakalaureusetöö peatükis 3.1 kirjeldatud õppemooduli peatükkide sisu ning peatüki 3.2 tutvustatud ülesannete kogu ülesannete loomisel lähtuti eelkõige peatükis 1.5 mainitud Merrilli mudelile. Järgnevalt tuuakse konkreetseid näiteid õppematerjalide osadest, mille koostamisel toetuti mudeli viiele punktile.

- 1) **Probleem.** Näite probleemipõhist lähenemist leiab juba õppematerjali 1. peatükist „PySimpleGUI ja esimene näide“ alguses. Enne õpetuste juurde asumist kuvatakse ekraanile näide graafikaaknast, mida õpilane õpib käesoleva peatüki läbimisel looma. Seejuures tagatakse parem orienteeritus õppematerjalides, sest asudes õpetusi lugema, saab õpilane iga hetk naasta näite juurde ja võrrelda õpetatavat potentsiaalse tulemusega. Näiteks on joonis graafikaaknast, millesse on paigutatud kõige elementaarsemad, kuid samas ka kõige vajalikumad graafilise kasutajaliidese elemendid - tekstisildid, tekstisisestuskast ning nupud (joonis 10).



Joonis 10. Kuvatõmmis loodud õppematerjali 1. peatükist (Lisa 1).

Seejärel alustatakse kõnealuse akna loomise seletamist põhimõtte järgi, et probleemile tuleb läheneda järk-järgult. Kuna üheks võimaluseks aknasse elemente paigutada on ridade kaupa, siis alustatakse ka akna loomise kirjeldamist rida rea haaval.

- 2) **Aktivatsioon.** Kuna loodud õppemooduli puhul on tegu suures osas ka iseseisvalt läbitava materjaliga, siis peab loodav sisu toetuma juba mingile teadmiste pagasile. Teisisõnu võib öelda, et uute materjalide omandamisel peaks toimuma varem omandatud info aktivatsioon.

Õpilane peab esimest graafikaakent luues rakendama teadmisi näiteks järjendite ja alamjärjendite olemusest, et mõista akna ülesehituse põhimõtet (joonis 11).

Seejärel paneme paika elemendid, mida aknasse planeerinud oleme. Loome muutuja `paigutus`, milleks on kahemõõtmeline järjend. Tuletame meelde, et näidisprogrammi aknas on neli rida, seega järjendi sisse peame lisama neli alamjärjendit, millest igaüks tähistab ühte rida.

```
paigutus = [[...], #1.rida
            [...], #2.rida
            [...], #3.rida
            [...]] #4.rida
```

Joonis 11. Kuvatõmmis loodud õppematerjali 1. peatükist (Lisa 1).

Varem omandatud teadmiste kasutamist vajab õpilane ka õppemooduli 2. peatükis „PySimpleGUI nupuvajutus, hüplikaken ja uue graafikaakna avamine“, kus rakendatakse funktsioonide ning sealjuures ka matemaatiliste tehete kasutamist programmeerimiskeeles Python (joonis 12).

Esmalt impordime PySimpleGUI klassid ja funktsioonid, sätestame akna värviteema ning koostamegi funktsiooni, mis arvutustehte sooritab. Tuleta meelde, kuidas Pythonis astendamine käis!

```
import PySimpleGUI as sg

sg.theme('Dark Blue 3')

#funktsioon arvutustehte sooritamiseks
def arvuta(arv):
    arvu_ruut = arv**2
    return arvu_ruut
```

Joonis 12. Kuvatõmmis loodud õppematerjali 2. peatükist (Lisa 1).

Õppemooduli 2. peatükis pööratakse tähelepanu ka andmetüüpide teisendamisele. Siinkohal peavad õpilased varemõpitud meelde tuletama, mis tüüpi andmeid sisestab kasutaja ning mis tüüpi andmetega saab sooritada arvutustehteid (joonis 13).

Kui aga vajutati 'Arvuta!' nuppu, saame esmalt kätte sisendi. See on meil salvestatud sõnastikku ning sellele vastav võti on `sisestatud_arv`.

Jälgi kindlasti, et sisend tuleb (täis)arvuks teisendada, et arvutustehet sooritada `int(...)` – vastasel juhul saame veateate!

```
if sündmus == 'Arvuta!':
    sisend1 = int(v22rtused['sisestatud_arv'])
```

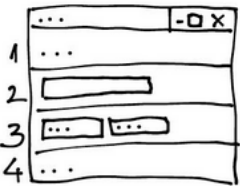
Joonis 13. Kuvatõmmis loodud õppematerjali 2. peatükist (Lisa 1).

Seega Merrelli mudeli 2. punktile toetumise eesmärgiks on koostada selline õppematerjal, mida omandades saab õpilane rakendada juba omandatud teadmisi, mis teevad omakorda uue info õppimise lihtsamaks.

3) **Demonstratsioon.** Digitaalse õppematerjali loomisel oli üheks olulisimaks punktiks teoreetiliste õpetuste illustreerimine näidetega. Seetõttu toodi igas peatükis uue infokillu juurde kas vastav illustratsioon või koodinäide.

Õppematerjali 1. peatükis on graafikaakna planeerimist demonstreeritud joonisega loodavast graafikaaknast (joonis 14). Sellega juhitakse tähelepanu asjaolule, et esmalt tuleks graafikaakna kujundus ja funktsioonid läbi mõelda, et programmeerimise protsess kulgeks sujuvamalt ning oleks selgelt eesmärgistatud.

Esmalt tuleks mõelda, millise kujundusega akent soovime teha. PySimpleGUI-s käib elementide paigutamine aknasse ridade kaupa kahemõõtmelise järjendi kaudu, kus iga rea elementid luuakse alamjärjendid, mis lisatakse omakorda ühte suurde järjendisse. Näiteprogramm on näiteks tehtud järgmise joonise põhjal:



Joonis 14. Kuvatõmmis loodud õppematerjali 1. peatükist (Lisa 1).

Õppemooduli 3. peatükis „PySimpleGUI graafikaakna vidinad“ illustreeritakse iga tutvustatud graafikaakna vidinat nii koodinäite kui ka kuvatõmmise abil. Sel juhul näeb õpilane kõikvõimalikke variante isegi enne, kui ta neid oma programmides rakendada asub, mis aitab õpilasel planeerida loodava akna elemente. Üheks vidinaks on näiteks liugur (joonis 15).

Liugur

Liugurit saab kasutada näiteks siis, kui on vaja valida arv vahemikus 1-10. Liuguri suund saab olla kas horisontaalne (*orientation='h'*) või vertikaalne (*orientation='v'*). Samuti saab määrata vaikeväärtuse.

```
sg.Slider(range=(1, 10), orientation='h', size=(20, 20), default_value=5)
```

Väljund:

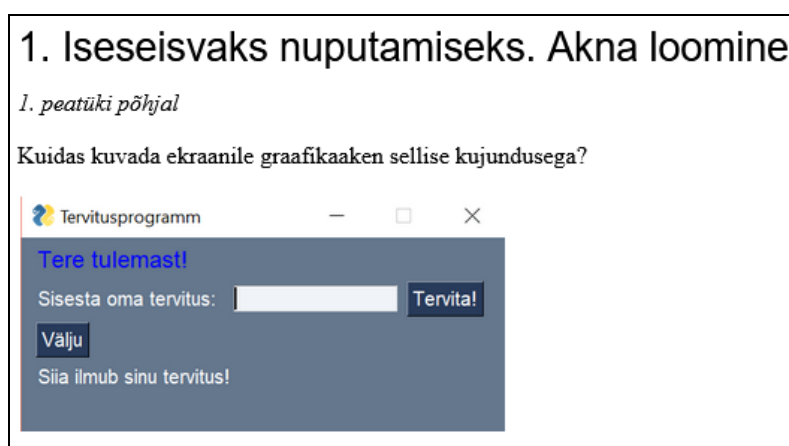


Joonis 15. Kuvatõmmis loodud õppematerjali 3. peatükist (Lisa 1).

Lisaks uute teadmiste demonstreerimisele peavad õpilased ka neid iseseisva harjutamise teel kinnistama, mille jaoks on loodud ülesannete kogu.

4) **Rakendamine.** Uute teadmiste rakendamiseks on eelkõige mõeldud eelmainitud ülesannete kogu. Kogus on hetkel nii 1.–3. kui ka 5. peatüki teemade kinnistamiseks loodud ülesanded.

Näiteks õppematerjali 1. peatüki „PySimpleGUI ja esimene näide“ põhjal loodud ülesanne võimaldab õpilasel luua graafikaaken ülesannete kogus toodud näite põhjal (joonis 16). Sellise lähenemise puhul peab õpilane jooniselt märkama, missuguseid graafikaakna elemente on kasutatud, kuidas nad on aknasse paigutatud ning mida peab programm kasutaja sisestatud tekstiga tegema.



Joonis 16. Kuvatõmmis loodud ülesannete kogu 1. osast (Lisa 2).

Ülesannete kogu dokumendis on enamustele ülesannetele toodud ka lahendused, et õpilased saaksid lisaks õpetaja kommentaaridele loodud programmidele tagasisidet ka näiteprogrammide põhjal.

Merrilli mudeli 4. punkti põhjal saab väita, et informatiivne tagasiside on õpilasele oluline, sest see toetab efektiivset õppimist ja ei pärsi motivatsiooni uue materjali omandamisel.

5) **Integreerimine.** Ülesannete kogu ülesannete sisu loomisel lähtuti Merrilli mudeli 5. punktist, mille järgi peaks ülesannete sisu võimaldama paralleelsele tömmata pärise-luga.

Näiteks 3. peatüki ülesannete teemad on tihedalt seotud spordiga ning samas sisal-davad ka kõiki seniõpitud teemasid – graafika- ja hüplikakna loomist, kasutajalt info küsimist ning erinevate vidinate kasutamist.

Antud ülesannete puhul peab programm ka ühikuid ja andmetüüpe teisendama vastavalt sellele, mida on nõutud. Samal ajal saavad õpilased neid teemasid korrata ja rakendada, sidudes saadud teadmisi eluliste teemadega. 3. peatüki ülesannete variandid on järgmised:

- **Variant A.** Koosta programm, mis küsib kasutajalt eraldi akendes joostud kilomeetrite arvu ning aja minutites ning seejärel arvutab, mitu kilomeetrit jõuaks kasutaja potentsiaalselt tunnis joosta, kui ta jätkaks sama tempoga. Kasuta info küsimiseks erinevat tüüpi elemente!
- **Variant B:** Koosta programm, mis küsib kasutajalt eraldi akendes kehakaalu ja pikkuse ning seejärel arvutab kehamassiindeksi (KMI). Kasuta info küsimiseks erinevat tüüpi elemente! Samuti kuva hüppaknale info, kas saadud KMI on normide piires või mitte.

Merrelli mudeli 5. punkti rakendati ka digiõpiku 4. peatüki „PySimpleGUI suurem näide“ loomisel. Suureks näiteks valiti ülesanne, kus õpilane peab looma tervituskaardi. Ülesanne on punktide kaupa lahti seletatud ning on toodud ka nõuded, mida graafikaaken peab sisaldama ning kuidas programmi tegevus kulgeb:

1. Esmalt küsitakse tervitatava ees- ja perekonnanime. Aknas on ka tekstikast, kuhu saab kirjutada tervituse.
2. Järgmises aknas saab sisestada tervitatava sünnikuupäeva. Päeva, kuu ja aasta valimiseks on kasutatud erinevaid viise – näiteks liugurit, rippmenüüd või valikute kasti. Samuti saab failikataloogist valida ühe pildi, mille kaardile juurde panna.
3. Kolmandas aknas kuvataksegi tervituskaart – aknas kuvatakse kõiki andmeid ja tekste, mille kasutaja on sisestanud. Aknas on nähtaval ka pilt, mille kasutaja valis. Lisaks tuleb arvutada ja ekraanile kuvada ka tervitatava vanus (siin näites on kasutatud raskemat varianti – arvesse on võetud ka sünnikuud ja -kuupäeva. Arvutamiseks on loodud funktsioon).
4. Viimasena avaneb hüppaken, mis tänab kasutajat programmi läbimise eest.

Ülesandel on päriseluline väärtus, sest lisaks päriselulisele sisule saaksid õpilased ka loodud tervituskaarte omavahel vahetada ning loodud graafilisi kasutajaliideste disaini ja kasutatavust võrrelda.

Järgnevalt analüüsitakse valminud õppematerjalidele antud tagasisidet.

3.4 Tagasiside tulemused

Töö käigus sai autor kõige rohkem tagasisidet kommentaaride ja soovitude põhjal, mida lisati töörühma liikmete ja gümnaasiumiõpetajate poolt tööprotsessi ajal tekstiredaktorisse Google Docs või mida mainiti ühistel veebikoosolekutel.

Üks asi, millele tähelepanu pöörati, oli see, et eestikeelsetes materjalides võiksid olla eestikeelsed muutujanimed. Nende kasutamisega tagatakse ühtsus materjalide lõikes ning ka uute programmeerimise teemade õppimise parem arusaadavus õpilaste jaoks. Arutleti ka teemade sügavuse üle – millised teemad peaksid olema lisade all ja millised mitte. Tuginedes saadud tagasisidele, viidi sisse vajalikud muudatused ning lisati loodud õppemooduli 5. peatükk joonistamisest lisamaterjalide hulka.

Töörühma liige, kes on ühtlasi ka tegev informaatikaõpetajana, retsenseeris valminud materjali 2020. aasta novembri lõpus pärast seda, kui lõputöö autor oli valmis saanud esimese versiooni õppematerjalidest. Põhiline aspekt, millele ta tähelepanu pööras oli see, et õppematerjali koostades peaks keerulisemad kohad veelgi üksikasjalikumalt lahti seletama. Samuti tõstis ta esile mõningaid sisulisi küsimusi, näiteks seda, et miks peab kasutama *while*-tsükli graafikaakna pidevaks ekraanil kuvamiseks. Õppematerjali 3. peatüki puhul soovitas ta kasutada ka erinevate graafikaakna vidinate juures näidispilte. Töö autor muutis vastavalt saadud tagasisidele õppematerjali sisu selgitamist selgemaks ja eakohasemaks ning visuaalselt paremaks, kasutades rohkem illustratiivseid näiteid. Need muudatused soodustavad õpilase arusaamist õppematerjalist, mis omakorda tõstab motivatsiooni õppida.

Kahe kooli õpilastelt ning informaatikaõpetajate konverentsilt saadud tagasiside küsimine piirdus vaid suusõnaliste kommentaaridega, mis olid positiivsed. Digiõpiku moodulit „PySimpleGUI“ kavatsetakse edaspidi hakata koolis kasutama, et teemast huvitatud õpilastele anda teadmisi graafilise kasutajaliidese loomise kohta.

3.5 Edasiarenduse võimalused

Potentsiaalseks õppemooduli edasiarenduse võimaluse suunaks on õppematerjali muutmine interaktiivsemaks. LORI mudeli järgi on interaktiivsus digitaalse õppematerjali juures oluline märksõna, mistõttu just selle arendamine oleks antud juhul esmatähtis.

Esimeseks võimaluseks, kuidas saaks graafilise kasutajaliidese õppematerjali muuta interaktiivsemaks, oleks lisada sellesse ka teema, mis tutvustab võimalust PySimpleGUIWeb.

Sellisel juhul saab loodud programmi käivitada ka veebibrauseris [31]. See on hea viis näiteks juhtida mingit ekraanivaba seadet, näiteks Raspberry Pi miniarvutit.

Magnar Vares on oma bakalaureusetööna koostanud miniarvutit Raspberry Pi ning operatsioonisüsteemi Linux tutvustavad õppematerjalid [32], mida saaksid soovijad kasutada, et õppida neid teemasid. Seejärel saab ühendada teadmised miniarvutist ja graafilise kasutajaliidese loomisest ning luua võimalust PySimpleGUIWeb kasutades kasutajaliides, mille nuppudele vajutades saab miniarvutit, mida on kasutatud näiteks roboti ehitamiseks, juhtida. 2020. aasta seisuga oli PySimpleGUIWeb aga veel arendamisel, mistõttu ei pruugi kõik funktsioonid veel töötada [31]. See oli ka peamiseks põhjuseks, miks hetkel antud teemat uues veebiõpiku moodulis ei tutvustatud. Erinevate õppematerjalide ühendamine soodustaks aga teemadevahelist lõimumist, mis toetaks õppematerjalide laiaulatuslikumat kasutamist. See omakorda aitab õpilastel siduda arusaamu erinevatest valdkondadest ning avardada IT-alast silmaringi.

Interaktiivsust saaks õppematerjalidesse lisada ka näiteks enesekontrolliküsimuste näol. Hetkel saavad moodulit läbivad õpilased lahendada praktilisi ülesandeid, mis leiduvad õpetajatele mõeldud ülesannete kogus. Bakalaureusetöö autor arvab ka, et õpetajatele mõeldud ülesannete kogusse võiks lisada rohkem (lisa)ülesandeid või ühele ülesandele enam valikuvariante. Sel juhul oleks suurem valikuvõimalus, milliseid ülesandeid lahendada. Samuti saaksid õpilased, kes huvituvad teemast rohkem või on kiiremini ettenähtud ülesanded sooritanud, lahendada lisäülesandeid.

Keerulisemate teemade puhul võiks rakendada ka õppevideote kasutamist. See võimaldab õpilasel jälgida reaajas mõne programmi koostamist, mille tekstilisest kirjeldusest võib väheks jääda. Interaktiivsust ei saanud bakalaureusetöö autor veel niivõrd mahukalt rakendada, sest materjalid olid aktiivselt loomis- ja katsetamisjärgus.

Loodud õppemoodul graafilisest kasutajaliidesest PySimpleGUI on hetkel huvilistele kättesaadav. Materjale võiks tulevikus aga veel katsetada, et õpilastelt koguda spetsiifilisemat tagasisidet potentsiaalsete arendusvõimaluste kohta.

Kokkuvõte

Informaatikaalased oskused, sealhulgas ka programmeerimise oskus, on tänapäeval ja eriti Eestis tähtsal kohal, sest aina enam lahendusi on digitaalsel kujul. Programmeerimist õpitakse enne kõrgkooli astumist gümnaasiumis valikkursustena. Samas on selgunud, et õpetavad informaatika kursused ei toeta üksteist, mistõttu vajaks informaatika ainekava uuendamist.

Ainekava uuendamise jaoks korraldas HITSA hanke, mille käigus uuendati muuhulgas ka valikkursust „Tarkvaraarendus“. Bakalaureusetöö autor ühines Tartu Ülikooli informaatika didaktika töörühmaga, et aidata õppematerjalide uuendamist täide viia. Valikkursust otsustati täiustada programmeerimiskeele Python graafilise kasutajaliidese moodulitega, millest üheks osutus PySimpleGUI.

Bakalaureusetööna loodi kursuse „Tarkvaraarendus“ uuendatud versiooni juurde lisamaterjalid, mis tutvustavad PySimpleGUI kasutajaliidese loomist. GUI loomine võimaldab programme interaktiivsemaks muuta. Materjalide koostamise juures peeti silmas nii Merrilli mudelit kui ka digitaalse õppematerjali loomise soovitusi. Materjalide juurde kuuluvad õpetajatele mõeldud ülesanded ning nende lahendused.

Koostatud õppemoodulis õpetatakse looma graafikaaknaid erinevate vajaminevate graafikaakna komponentidega ning sinna lisama näiteks tekstivälju ja nuppe vastavalt sellele, millise programmiga on tegu. Samuti kasutab õpilane ülesannete tegemisel juba varemõpitud oskuseid – näiteks tsüklite ja tingimuslausete loomist ning funktsioonide rakendamist. Õppematerjali lõpuosas demonstreerib õpilane õpitud oskusi, luues suurema programmi, mis kasutaks graafilist kasutajaliidest. Ühtlasi sisaldab õppematerjal ühte suuremat näidet programmist koos graafilise kasutajaliidese, et anda õpilastele inspiratsiooni, milline võiks suurem programm välja näha.

Loodud õppematerjalidele koguti tagasisidet mitmel viisil. Lisaks töörühma liikme tagasisidele küsiti arvamusi ka kahest Eesti gümnaasiumist ning informaatikaõpetajate konverentsilt. Peamised kommentaarid, mida tagasisidel saadi, olid seotud sõnastuse ja teemade sügavusega. Samuti juhiti tähelepanu sellele, et eestikeelsetes õppematerjalides võiksid muutujanimed olla eestikeelsed. Saadud info põhjal viidi materjalidesse sisse veel viimased muudatused ning seejärel pandi need üles digiõpikusse.

Potentsiaalseteks edasiarenduste võimalusteks oleks muuta õppematerjali interaktiivsemaks. Selle eesmärgi täideviimiseks võiks luua näiteks veel üks lisapeatükk, kus tutvustatakse võimalust PySimpleGUIWeb. Samuti oleks variandiks loodud õppematerjali integreerida teiste materjalidega, milleks võiks olla näiteks Raspberry Pi. Töö autor pakub välja ka võimaluse lisada ülesannete kogusse õppevideoid ja rohkem (lisa)ülesandeid.

Viidatud kirjandus

- [1] OECD. Future Of Education and Skills 2030 Concept Note, 2019. https://www.oecd.org/education/2030-project/teaching-and-learning/learning/skills/Skills_for_2030.pdf
- [2] Gareis et al. e-Skills for Jobs in Europe – Measuring Progress and Moving Ahead: Final Report, 2014.
- [3] Tartu Ülikooli informaatika didaktika töögrupi koduleht, programmeerimise kursused. <https://didaktika.cs.ut.ee/progttl/kursusest/> (12.03.2021)
- [4] Riigihangete register, Gümnaasiumi informaatika valikkursuste arendus- ja koolitustegevus. (09.07.2020). <https://riigihanked.riik.ee/rhr-web/#/procurement/2145832/general-info> (05.05.2021)
- [5] Tuomi, P; Multisilta, J. Coding skills as a success factor for a society, 2018.
- [6] Arfé, B.; Vardanega, T.; Ronconi, L. The effects of coding on children’s planning and inhibition skills, 2020.
- [7] Tocháček, D.; Lapeš, J.; Fuglík, V.. Developing Technological Knowledge and Programming Skills for Secondary School Students through the Educational Robotics Projects, 2016.
- [8] Botirovich, B.D. et al. The importance of teaching algorithms and programming languages in the creation of electronic education resources, 2020. <http://www.jcreview.com/fulltext/197-1592823353.pdf?1615388092>
- [9] Orgmets, M-L. „IT valdkonnas valitseb jätkuvalt spetsialistide puudus“ ERR raadiouudised, 2017. <https://www.err.ee/608507/it-valdkonnas-valitseb-jatkuvalt-spetsialistide-puudus> (11.03.2021)
- [10] Startup Estonia, About Startup Estonia. <https://www.youtube.com/watch?v=8BLIRs8-y3o> (viimane külastus: 16.04)
- [11] Rein, E. Eesti gümnaasiumites õpetatavad programmeerimise kursused. TÜ arvutiteaduste instituudi bakalaureusetöö. 2020.
- [12] Kori, K. et al. IT oskuste arendamine Eesti koolides. Tartu Ülikool, haridusteaduste ja arvutiteaduste instituut. Wise-i tellitud uuringu raport, 2019. <https://wise.com/documents/IT%20oskuste%20arendamine%20Eesti%20koolides.pdf>

- [13] Gümnaasiumi riiklik õppekava, 2011.
<https://www.riigiteataja.ee/akt/129082014021?leiaKehtiv> (02.05.2021)
- [14] E-õppepäev Tartu Ülikoolis. MOOCid. <https://epaev.ut.ee/moocid> (05.03.2021)
- [15] Leppik, C.; Haaristo, H-S.; Mägi, E. IKT-haridus: digioskuste õpetamine, hoiakud ja võimalused üldhariduskoolis ja lasteaias. HITSA tellitud uuringu raport, 2017.
https://media.voog.com/0000/0034/3577/files/IKThariduse_uuring_1%C3%B5pparuanne_mai2017.pdf
- [16] HITSA koduleht, gümnaasiumi informaatika ainekava. <https://www.hitsa.ee/ikt-haridus/progetiiger/gumnaasiumi-informaatika-ainekava> (03.05.2021)
- [17] Haridus- ja Noorteameti koduleht, gümnaasiumi informaatika ainekava. <https://projektid.edu.ee/pages/viewpage.action?pageId=81365502> (02.05.2021)
- [18] Küberkaitse valikkursuse veebileht, sissejuhatus kursusesse. <https://web.htk.tlu.ee/digitaru/kyberkaitse/front-matter/introduction/> (02.05.2021)
- [19] Omni.Sci koduleht, Graphical User Interface Definition. <https://www.omnisci.com/technical-glossary/graphical-user-interface> (02.05.2021)
- [20] Abdo, A.M. Graphical User Interface Features in Building Attractive And Successful Websites. Zakho Ülikool, Arvutiteaduste Instituut, Kurdistan, Iraak, 2016.
- [21] Python, GUI Programming in Python. <https://wiki.python.org/moin/GuiProgramming> (02.05.2021)
- [22] easygui 0.97 documentation, EasyGUI. (2020). <http://easygui.sourceforge.net/> (02.05.2021)
- [23] Beniz, D.B; Espindola, A.M. Using Tkinter of Python to create Graphical User Interface (GUI) for scripts in LNSL, 2016.
- [24] PySimpleGUI, PySimpleGUI User's Manual.
<https://pysimplegui.readthedocs.io/en/latest/> (02.05.2021)
- [25] Merrill, M.D. First Principles of Instruction, TR&D, Vol. 50, No. 3, pp 43–59, 2002.
<https://mdavidmerrill.files.wordpress.com/2019/04/firstprinciplesbymerrill.pdf>
- [26] Villems, A. et al. Digitaalse õppematerjali loomise soovitusel, 2014-2015.
<http://oppevara.hitsa.ee/kvaliteet/#tehnilised-pohimotted> (03.05.2021)

- [27] Tõnisson, E. jt. Tarkvaraarendus. 2. trükk. Kahemõõtmeline järjend. (2020). <https://web.htk.tlu.ee/digitalu/tarkvara2/chapter/kahemootmelinejarjend/> (03.05.2021)
- [28] Google, Google Docs. <https://www.google.com/docs/about/> (02.05.2021)
- [29] Tartu Ülikool arvutiteaduste instituut, hübriidkonverentsil tutvustatakse informaatika õpetamist koolides. (17.11.2020) <https://www.cs.ut.ee/et/uudised/hubriidkonverentsil-tutvustatakse-informaatika-opetamist-koolides> (04.05.2021)
- [30] Tõnisson, E.; Palts, T.; Tõnisson K.; Meier, H.; Säde, M.; Krusberg, S.M. Tarkvaraarendus. 2. trükk. Veebiõpik. <https://web.htk.tlu.ee/digitalu/tarkvara2/> (05.05.2021)
- [31] Python, PySimpleGUIWeb 0.30.0. (2020) <https://pypi.org/project/PySimpleGUIWeb/> (03.05.2021)
- [32] Vares, Magnar. Valikkursuse „Linux ja Raspberry Pi“ loomine ja pilootimine. TÜ arvutiteaduste instituudi bakalaureusetöö. 2020.

Lisad

I. Link loodud õppematerjalile „PySimpleGUI“

<https://web.htk.tlu.ee/digitalu/tarkvara2/chapter/pysimplegui-sissejuhatus/>

II. Link loodud ülesannete kogule

https://docs.google.com/document/d/1fNxrWkqj2wfaok5RghvxRdnsuIEiZvL5_vTi4TKck/edit

III. Litsents

Lihtlitsents lõputöö reprodutseerimiseks ja üldsusele kättesaadavaks tegemiseks

Mina, Säde Mai Krusberg,

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) minu loodud teose
“Gümnaasiumi valikkursusesse “Tarkvaraarendus” PySimpleGUI mooduli loomine”,

mille juhendaja on Tauno Palts, reprodutseerimiseks eesmärgiga seda säilitada, sealhulgas lisada digitaalarhiivi DSpace kuni autoriõiguse kehtivuse lõppemiseni.
2. annan Tartu Ülikoolile loa teha punktis 1 nimetatud teos üldsusele kättesaadavaks Tartu Ülikooli veebikeskkonna, sealhulgas digitaalarhiivi DSpace kaudu Creative Commons'i litsentsiga CC BY NC ND 3.0, mis lubab autorile viidates teost reprodutseerida, levitada ja üldsusele suunata ning keelab luua tuletatud teost ja kasutada teost ärieesmärgil, kuni autoriõiguse kehtivuse lõppemiseni.
3. olen teadlik, et punktides 1 ja 2 nimetatud õigused jäävad alles ka autorile.
4. kinnitan, et lihtlitsentsi andmisega ei riku ma teiste isikute intellektuaalomandi ega isikuandmete kaitse õigusaktidest tulenevaid õigusi.

Säde Mai Krusberg
07.05.2021