

Tallinna Ülikool
Digitehnoloogiaste instituut

**PROGRAMMEERIMISALASE
ÕPPEMATERJALI ÕPIDISAIN iOS
ARENDUSE ÕPPEMATERJALI NÄITEL**

Seminaritöö

Autor: Marianne Kristiina Orusalu

Juhendaja: Hans Põldoja

Autor:..... ”” 2018

Juhendaja:..... ”” 2018

Tallinn 2018

Autorideklaratsioon

Deklareerin, et käesolev seminaritöö on minu töö tulemus ja seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on viidatud.

.....

(kuupäev)

.....

(autor)

Sisukord

Sissejuhatus.....	4
1. Programmeerimisalaste õppematerjalide õpidisaini põhimõtted.....	7
1.1. Õpidisaini protsess.....	7
1.2. Multimeedium ja interaktiivsed lahendused programmeerimise õppematerjalides.....	8
1.2.1. Videod.....	9
1.2.2. Enesekontrolli testid.....	10
1.2.3. Simulatsioonid.....	11
1.2.4. Mängupõhise õppe lahendused.....	11
1.2.5. Kogukonnas õppimist toetavad lahendused.....	13
2. Programmeerimisalased õppematerjalid.....	15
2.1. Programmeerimisõpetuse veebikeskkonnad ja äppid.....	15
2.2. Programmeerimisalased õpimängud ja simulatsioonid.....	23
2.3. TLÜ üliõpilastööde raames loodud programmeerimisalased õppematerjalid.....	26
3. Õppematerjali kavandamine.....	30
Kokkuvõte.....	34
Kasutatud kirjandus.....	35

Sissejuhatus

Tehnoloogia üks olulisemad rolle on lihtsus või õigemini elu lihtsustamine. Rutiinis võib jääda märkamatuks, kogu sujuvalt liikuvate süsteemide keskel, kuidas mõni väiksem või suurem asi oma funktsiooni täpselt täidab ning mis kõik selle süsteemi taga on. Lihtne on võtta iseenesestmõistetavalt igapäevaelus kasutatavaid digitaalseid vahendeid, mis töötavad ja ärkavad ellu just tänu läbimõeldud programmeerimisele. OSKA IKT¹ uuring toob ühe olulise arengutrendina välja seadmete mitmekesisuse kasvu (nt nutitelefonid, tahvelarvutid, nutitelerid, erinevad sensorid autodes jm) ning rõhutab tööjõuvajaduse kasvu tarkvaraarenduses (SA Kutsekoda, 2016). Programmeerimisoskus tänapäeva ühiskonnas ei ole vajalik ainult selles vallas töötavatele inimestele vaid tuleb kasuks igale ühele meist, arendades loogilist mõtlemist, probleemide lahendamise oskust ning loovust. Mida rohkem me teame sellest, kuidas meie tehnilised vahendid toimivad, seda turvalisem ja kindlam on meil neid ka kasutada. Samas on vajalik äratada noortes huvi programmeerimiskuse vastu, et kasvatada IKT valdkonnas õppijate arvu ning õpetada välja tulevasi tarkvaraarendajaid.

Igal aastal koostatav haridustrendide uuring Horizon Report toob üldhariduse lähiaastate kahe olulisema trendi hulgas välja programmeerimiskuse arendamise (Freeman, Adams Becker, Cummins, Davis, & Hall Giesinger, 2017). Eestis on programmeerimise õpetamise arendamiseks üldhariduskoolides käivitatud ProgeTiigri programm, mille raames on loodud õppematerjale, hangitud koolidele robotikaseadmeid ning koolitatud õpetajaid (HITSA, 2015). Seni on Eesti koolides programmeerimise õpetamisel pööratud enam tähelepanu hariduslikele programmeerimiskeeltele ja robotikale, kuid vähe on tegeletud mobiilirakenduste arendamisega. Samas puutuvad noored igapäevaselt kokku just erinevate mobiilirakendustega. Väljakutseks on see, kuidas õpetada mobiilirakenduste arendust

¹ OSKA – SA Kutsekoda poolt läbiviidud uuring OSKuste Arendamise koordineerimissüsteemi loomine

noorte jaoks hästi arusaadaval ja mängulisel viisil. Heade näidetena saab siin välja tuua Androidi arenduse õpetamiseks loodud MIT App Inventor² ning iOS arenduse õpetamiseks loodud Swift Playgrounds³. Programmeerimise õpetamist toetavad veebipõhised õppematerjalid, mida on võimalik huvitavaks teha põimides õppetöösse videod ja erinevad interaktiivsed ning mängupõhised lahendused.

Autor näeb probleemina, et eestikeelsed programmeerimisalased õppematerjalid ei kasuta piisavalt multimeediumipõhiseid, interaktiivseid ja kogukonnas õppimist toetavaid lahendusi. Lisaks sellele on autor jõudnud ka järeldusele, et eestikeelseid digitaalseid programmeerimisalaseid õppematerjale on üldiselt vähe. Kuna autoril on huvi iOS rakenduste arendamise vastu, otsustas ta luua kooliõpilastele ja mobiilirakenduste loomisest huvitatutele suunatud eestikeelse õppematerjali iOS arenduse kohta lihtsa rakenduse loomise näitel. Õppematerjali loomisega seotud tegevused on jagatud seminaritöö ja bakalaureusetöö vahel – seminaritöö keskendub õppematerjali kavandamisele, kuid õppematerjali loomine ja testimine jääb bakalaureusetöösse.

Käesoleva seminaritöö eesmärgiks on anda ülevaade erinevatest võimalustest programmeerimisalaste õppematerjalide loomisel ning kavandada lihtsa eestikeelse iOS arenduse õppematerjali ülesehitus.

Eesmärgi saavutamiseks püstitas autor järgmised ülesanded:

- analüüsida kirjanduse põhjal, millised on erinevad võimalused programmeerimisalaste õppematerjalide loomiseks;
- analüüsida näiteid programmeerimisalastest õppematerjalidest ning leida nende põhjal sobivaid lahendusi lihtsa iOS arenduse õppematerjali koostamiseks;
- kavandada lihtsa iOS arenduse õppematerjali ülesehitus.

² <http://appinventor.mit.edu>

³ <https://www.apple.com/swift/playgrounds/>

Seminaritöö koosneb kolmest peatükist. Esimeses peatükis seletab autor lahti õpidisaini mõiste ning analüüsib kirjanduse põhjal, millised on erinevad võimalused programmeerimisalaste õppematerjalide loomiseks, tuues sealjuures välja ka näiteid. Teises peatükis toob autor välja head näited mujal maailmas kasutatavatest programmeerimise õppematerjalidest, mille hulgas on nii mobiilirakendusi kui ka veebilehti. Samas peatükis analüüsib autor ka näiteid programmeerimisalastest õppematerjalidest, mis on valminud Tallinna Ülikoolis. Autor otsib välja nende kahe punkti põhjal sobivaid lahendusi lihtsa iOS arenduse õppematerjali koostamiseks. Kolmandas peatükis kavandab autor lihtsa iOS arenduse õppematerjali ülesehituse. Antud peatükis on välja toodud õppematerjali teemad, sisutüübid, esitamise platvorm jne. Seminaritöö on sisendiks bakalaureusetööle, mille raames siin kavandatud õppematerjal luuakse.

1. Programmeerimisalaste õppematerjalide õpidisaini põhimõtted

Käesolevas peatükis antakse kirjanduse analüüsi põhjal ülevaade õpidisaini mõistest ja õppematerjalide loomise protsessist. Samuti leiab autor näiteid multimeediumi ja interaktiivsete lahenduste kasutusvõimalustest programmeerimise õppematerjalides.

1.1. Õpidisaini protsess

Eduka õppematerjali koostamisel tuleks tähelepanu pöörata kindlale ülesehitusele, mis oleks praktiline, efektiivne ning kutsuks õppima ja ennast arendama. Selleks, et luua eeldused eduka e-õppematerjali koostamiseks oleks vaja lähemalt tutvuda õpidisaini (ingl *instructional design*) mõistega. Seda võib lahti mõtestada paljudel erinevatel viisidel, mis kõik viitavad kokkuvõttes õppeprotsessi kuuluvatele põhimõtetele ja vaatenurkadele. Õpidisain kujutab endast õpikeskkonna ja õppematerjalide loomise teadust, mis aitab õppijal omandada uued teadmised ja oskused kõige tõhusamal viisil. Õpidisain põhineb teoreetilistel ja praktilistel uuringutel tunnetuse, hariduspsühholoogia ja probleemide lahendamise alal (Siemens, 2002). Selleks, et teostada ning läbi teha üks õpidisaini protsess, on olemas mitmeid erinevaid protsessimudeleid, millest üks tuntumaid on ADDIE mudel (Branch, 2009), mille tugevateks külgedeks on selle paindlikkus, kontekstitundlikkus ja interaktiivsus. ADDIE mudeli nimetus on akronüüm, mis on moodustatud õpisüsteemide arenduse viie etapi järgi (Branch, 2009; Villems et al., 2015):

- Analüüsi etapp (ingl *Analysis*) – Analüüsi etapis kogutakse informatsiooni inimeste kohta kellele materjal suunatud on, õpitava ülesande kohta, kuidas antud inimesed materjali sisu näevad ja mis on projekti üldised eesmärgid. Lõpus saadud info klassifitseeritakse, et tagada sisu rakendatavus ning edu. Analüüsi etapi lõpus valitakse ka tehniline platvorm õppematerjali tegemiseks.
- Kavandamise etapp (ingl *Design*) – Selles faasis toimub õppematerjali loomine, kus pannakse paika materjali eesmärk ja õpitulemused ning sobivad

õpetamismeetodid. Kavandamise etapis on ette nähtud ka kasutatavate meediatüüpide valimine.

- Väljatöötamise etapp (ingl *Development*) – Väljatöötamise lõpptulemuseks on valmis ja avalikustatud õppematerjal koos metaandmetega, milleks on vaja läbi viia sisu loomine, tehniline teostus ja testimine.
- Kasutamise etapp (ingl *Implementation*) – Antud etapp saab kõiki materjale testida, et määrata kindlaks nende funktsionaalsus ja sobilikkus.
- Hinnangu andmise etapp (ingl *Evaluation*) – Hinnangu faasis saab selgeks, kas materjalid täitsid oma ettenähtud eesmärgi.

1.2. Multimeedium ja interaktiivsed lahendused programmeerimise õppematerjalides

Suurimaid probleeme programmeerijaks õppimisel on koodist arusaamise oskus. Programmeerimine võib algajale tunduda ülimalt keerulisena ning selle õppimine hirmutavana. Tegelikult sõltub asja lihtsus hoopis õppemeetodist. Kuna tegemist on keeruka ja detailiderohke protsessiga, ei saa ka õpetamisel tugineda traditsioonilistele õppemeetoditele. Programmeerimise õppematerjalide eesmärgi edukaks saavutamiseks peaks õpetamises kasutama erinevaid multimeediumi võimalusi ja interaktiivseid lahendusi, mis täiendavad olemasolevat õppematerjali ning aitavad seda selgemalt ja edukamalt jälgida. Haridustehnoloogia- ning psühholoogiaalased uuringud näitavad, et õpilase praktiline töös osalemine on oluline osa multimeediumite hariduslikust efektiivsusest. Õpilane ei tohiks olla passiivne õppematerjali tarbija, vaid pigem aktiivne visualisatsiooni kasutaja. See aitab õppijal teemat teadlikumalt hallata tagades põhjalikuma teadmiste baasi õpitava vajalikkusest ning selle funktsioonidest (Sorva, 2012). Hea ülevaate annab ka Rinde (2010) oma magistritöös, kus väljatoodud õppimise püramiid⁴ näitab ära, et

⁴ NTL (National Training Laboratories, USA) poolt välja pakutud püramiid, mis näitab keskmist meeldejäetava materjali protsenti erinevate õppimis- ja õpetamismeetodite kasutamisel.

multimeediumivahendite kasutamine õpetamisel annab tavaliselt paremaid tulemusi kui traditsioonilised loengud ja tekstimaterjali lugemine.

1.2.1. Videod

Üheks tõhusaimaks õppimisviisiks võib pidada spetsialisti kõrvalt jälgimist ning seejärel kohest imiteerimist. See võib osutada aeganõudvaks protsessiks, kuna arvestama peab nii õppejõu kui ka õpilase ajaga ning kui tegemist ei ole individuaalse õpetamisega, siis loovad õppimise tempo teema kiiremad omandajad. Videote kasutamine õppevahendina, kas siis täiendava õppeosana või individuaalselt, võimaldab õpitavat visuaalselt jälgida ning kui tekib probleeme järgi jõudmisega saab alati video seisma panna või soovitud kohta uuesti vaadata. Programmeerimisalaste õppematerjalide puhul on videod eriti olulised graafiliste arenduskeskkondade puhul (nt Eclipse⁵, Xcode⁶, jne), kus lisaks koodi kirjutamisele on vaja ka teada, kuidas arenduskeskkonda kasutada. Videod ei asenda küll otsest kontakti ning lisaküsimuste või probleemide korral kohest tagasiside saamist kuid videod on hea viis õpitava materjali kinnistumiseks visuaalselt nii klassiruumis kui ka iseseisvalt.

Saksamaa Ålborgi Ülikoolis läbiviidud uuringud (Nørmark, 2014) näitasid, et sealseid programmeerimist õppivad õpilased suhtuvad väga positiivselt lühivideo loengutesse või isegi peavad neid hädavajalikuks osaks õppimises. Õpilaste tagasiside antud uuringus andis ülevaate ka sellest, mida õpilased videoloengutest ootavad. Teretulnud on teadlikult valitud, õppekava teemale vastavad ning koormust arvestavad videod. Lühivideod olid õpilaste seas populaarsemad, kui pikad (1–2 tunnised) traditsioonilised loenguvideod. Umbes viiendik õpilastest sooviks näha mõlemas stiilis (lüh- ja pikad videod) lühistusi traditsioonilisest loengust (Nørmark,

⁵ <https://www.eclipse.org/ide/>

⁶ <https://developer.apple.com/xcode/>

2014). Eestis on taolist loengute salvestamist suuremas mahus kasutatud näiteks IT Kolledžis⁷, mis nüüdsest on osa Tallinna Tehnikaülikoolist⁸.

Edukaimad õppevideod on ajaliselt lühemas formaadis ning põhjalikult läbi mõeldud ülesehitusega. Suur osa selliseid videoid tutvustavad õpetatavat teemat seostades uut informatsiooni eelnevate teadmistega. Selline õppeviis toimib eriti tõhusalt algajate programmeerijate seas nende iseseisva õppimise toetamiseks. Antud õppevideote populaarsusele osutab suurelt ka nende massiline hulk YouTube'i kanalitel või Khan Academy ja Code.org lehtedel (Aureliano, 2014).

USA Drake Ülikoolis uuringute põhjal selgus, et videote kasutamine programmeerimisõppes muudab ka programmeerimisega muidu mitte kokkupuutuvate õpilaste arusaama ning arvamust antud erialast. Videopõhine õppimine paneb neid nägema programmeerimist lõbusa ning huvitavana (Manley & Urness, 2014).

1.2.2. Enesekontrolli testid

Nagu koodi kirjutamises on vaja teha vahepealseid teste, et veenduda korrektsuses ning likvideerida vead, enne kui need kuhjuma hakkavad, siis sama süsteem võiks kehtida ka õppimises, eriti kui tegemist on programmeerimise õppimisega. Sellele tuginedes on edukaks materjali omandamise eelduseks kindlasti ka enesekontrolli testid. Kõige lihtsamaks enesekontrolli testiküsimuse tüübiks, mida programmeerimisalaste õppematerjalide juures kasutada saab, on valikvastustega küsimused. Selliste küsimuse abil on võimalik kontrollida arenduskeskkonna kasutajaliidese tundmist, koodi mõistmist, koodist vigade leidmist jms (Clark, 2004). Programmeerimisalastes õppematerjalides on enesekontrolliks kasutusel ka erinevate koodide parandamine ja lünkade täiendamine eelnevalt õpitu põhjal ning selle eest

⁷ <http://www.itcollege.ee/>

⁸ <https://www.ttu.ee/>

antakse ka kohest tagasisidet. Sellist meetodit kasutavad paljud veebipõhised õppelehed ning äpid, nende seast tuntuim näiteks DataCamp⁹.

1.2.3. Simulatsioonid

Simulatsioon on reaalmaailma protsessi või süsteemi jäljendamine ajas, mis võimaldab õppijatel riskivabas keskkonnas oma uut teadmist rakendada ja uusi oskusi omandada. Hariduslikud simulatsioonid võimaldavad uue teema õppimist läbi reaalse situatsiooni jäljendamise. Programmeerimise õpetamisel sobib simulatsioone kasutada näiteks lastele programmeerimise aluste õpetamisel või programmeerimises võhikutele, kes soovivad näha kohest seost probleemi lahenduse väljatöötamisel ja realiseerimisel.

Yuen (2006) kirjeldab programmeerimise põhimõtete õpetamist läbi liiklusmängu, kus esmalt antakse õpilasele ette multimeediat sisaldav programm, mis motiveerib neid õppima ning äratav uudishimu loomise, uurimise, testimise ja programmist arusaamise vastu. Antud simulatsioon annab pidevalt tagasisidet ning pakub väljakutset, mis ergutab õpilasi mitte alla andma. Yuen leiab samuti, et õpilased on oma õpingutes edukamad, kui nad osalevad aktiivselt programmi ehituse loomises, mitte passiivsel programmi jälgimisel.

1.2.4. Mängupõhise õppe lahendused

Simulatsioonid on tihedalt seotud mängudega, nimelt mida rohkem mängulisi elemente simulatsioonile lisada, seda rohkem hakkab see endast kujutama midagi õppimängu sarnast. Kui õppimine tundub millegi igavana, siis üks kindlaid viise selle põnevamaks muutmiseks on pöörata pilk mängumaailma. Stigma arvutimängude ümber on hakanud vaibuma ning ka meelelahutuslikuks kasutuseks mõeldud mängud õpetavad otsuste tegemist, meeskonna juhtimist ja leidlikkust, sest nad pakuvad turvalist keskkonda läbikukkumiseks. Otseseks õppevahendi tarbeks

⁹ <https://www.datacamp.com/>

luuakse aga tõsimänge ehk õpimänge. Kõitva ning samas õpetliku tõsimängu loomiseks on vajalik nii programmeerijate kui ka õpetajate koostöö.

Õpimäng kujutab endast mänguelementide lisamist õppimisse, näiteks punktide ja digitaalsete tunnustusmärkide kogumise võimalus, et haarata kaasa ning motiveerida sellega õpilasi. Tulemuseks on õppija rahulolu, sarnaselt arvutimänge mängides võiduni jõudes, kui ta on programmeerimisprobleemi mängu kaudu lahendanud.

Selleks et mängu elemendid oleksid edukad, peaksid need toetuma järgneval motivatsiooni teoorial (Morrison & Disalvo, 2014):

- **Eesmärkide püstitamine (ingl *setting goals*):**
 - Konkreetsed mitte üldised – mängud, mis annavad pigem konkreetseid kui üldiseid eesmärke, peetakse motiveerivamateks.
 - Jõupingutuste näitamine – motivatsiooni hoidmiseks on oluline eesmärkide saavutamiseks vajalike jõupingutuste hindamise võimalus.
 - Lähieesmärgid – lühiajalised eesmärgid võivad parandada motivatsiooni ja õppimist paremini, kui pikaajalised eesmärgid.
 - Raskuste määramine – kui mängija leiab, et eesmärki on raskem saavutada, tahab ta selle kallal rohkem töötada, kui siis kui tegemist on liiga kerge eesmärgiga.
 - Keskendumisprotsess vs. tulemus – lapsed, kes töötavad probleemilahenduse eesmärgi nimel ja saavad vahepealset tagasisidet oma arengust eesmärgi saavutamiseks, on rohkem motiveeritud.
- **Tulemuse ootused (ingl *outcome expectation*):**
 - Sisemised ja välised ootused – nii sisemised kui ka välised tulemusootused tõstavad õpilase motivatsiooni.
 - Mudelid – õpilased on rohkem motiveeritud, kui neile on ette näidatud edukas mudel antud mängus.
- **Väärtused (ingl *values*):**
 - Tegevuse väärtus tagab motivatsiooni – õpilased, kes annavad tegevusele väärtuse, võime omada kontrolli selle üle mida nad suudavad saavutada, on rohkem motiveeritud üritama ja saavutama soovitud tulemust.

- Väärtused viivad õppimise ja õppimisest loobumiseni – väärtused võivad viia õppimiseni, sest õppiija väärtustab tulemusi või tegevusi, mida on antud ülesande õppimiseks vaja. Väärtused võivad õpilast ka õppimisest loobuma panna, sest tulemused või tegevused, mis on antud ülesande lahendamiseks vajalikud ei ühti õpilase enda väärtustega.
- **Sotsiaalne võrdlus (ingl *social comparisons*):**
 - Võrdlus sarnaste inimestega – sotsiaalne võrdlus innustab enesekontrolli, mis on motivatsiooni aluseks.
 - Raskete eesmärkide püstitamine – teadlased on kindlaks teinud, et elemendid, mis edendavad eesmärki, näiteks raskemate eesmärkide püstitamine, tagavad kõrgema motivatsiooni kui sotsiaalsed võrdlused.

Fotaris, Mastoras, Leinfellner, ja Yasmine (2015) tegid oma uurimustöö tulemusena kindlaks, et kasutades mitmedimensioonilisi mängulisi lähenemisi õppimisele, saavutatakse edu õpilastes motivatsiooni tõstmises ning teadmiste kogumiseks programmeerimises. Tõsimängud annavad nii õpilastele kui õpetajatele võimaluse teha õppimisest lõbus, tulemusteni viiv ning sõltuvust tekitav protsess.

1.2.5. Kogukonnas õppimist toetavad lahendused

Inimeselt inimesele kontaktil on tähtis osa kõikide teadmisvaldkondade omandamises, ka programmeerimise õppimine ei ole siin erandiks. Näha, et teised õpilased maadlevad sama probleemiga, aidates üksteist probleemide lahendamises ning jagada ühiselt võidurõõme on suureks motivatsiooniks. Klassiruumis, interaktiivses loengusaalis või praktilistes tundides osalemine annab õpilasele võimaluse vastastikuseks kontaktiks ja mõtete avaldamiseks ja vahetamiseks ning tagasiside saamiseks õppejõu ning ka kaasõpilastega. Iseseisvalt kodus ning internetis õppides võib juhtuda, et selline inimeselt inimesele kontakt on piiratud või üldse mitte eksisteerida. Õppides programmeerima arvutipõhises õppes testides, inimese ja arvuti vahelise suhtlemise ja mängude ning teiste multimeediumi võimaluste kaudu võib osutada mõnes kohas pingeliseks ning raskeks, kui on vaja

kohest tagasisidet või selgitust millelegi, mida arvuti ei ole lihtsalt programmeeritud tegema. Selleks on aga olemas mõningaid lahendusi programmide ja veebilehtede näol. Näiteks luuakse õppimist toetavad veebipõhised õpikeskkonnad ning kasutatakse kogukonnapõhiseid arenduskeskkondi nagu GitHub¹⁰ või spetsiaalselt Python'i koodi õppimiseks ja jagamiseks mõeldud CodeSkulptor¹¹ (Warren, Rixner, Greiner, & Wong, 2014).

¹⁰ <https://github.com/>

¹¹ <http://www.codeskulptor.org/>

2. Programmeerimisalased õppematerjalid

Hea programmeerimisalase õppematerjali loomiseks peaks üle vaatama kindlasti ka eelnevad sarnased materjalid. Nii saab võtta eeskuju ning vältida puuduseid või luua ettekujutus sellest, milline õppematerjal välja võiks näha ning kuidas seda tulevikus edasi oleks võimalik arendada. Järgnevas peatüki esimeses pooles on autor välja toonud valiku erinevaid programmeerimisalaseid õppematerjale mujalt maailmast veebikeskkondade, äppide ning mängude ja simulatsioonide näol.

Peatüki teine pool kajastab Tallinna Ülikooli üliõpilastööde raames loodud programmeerimisalaste materjalide võrdlust, tuues välja nendes leiduvad head lahendused ja kasutamata võimalused.

2.1. Programmeerimisõpetuse veebikeskkonnad ja äppid

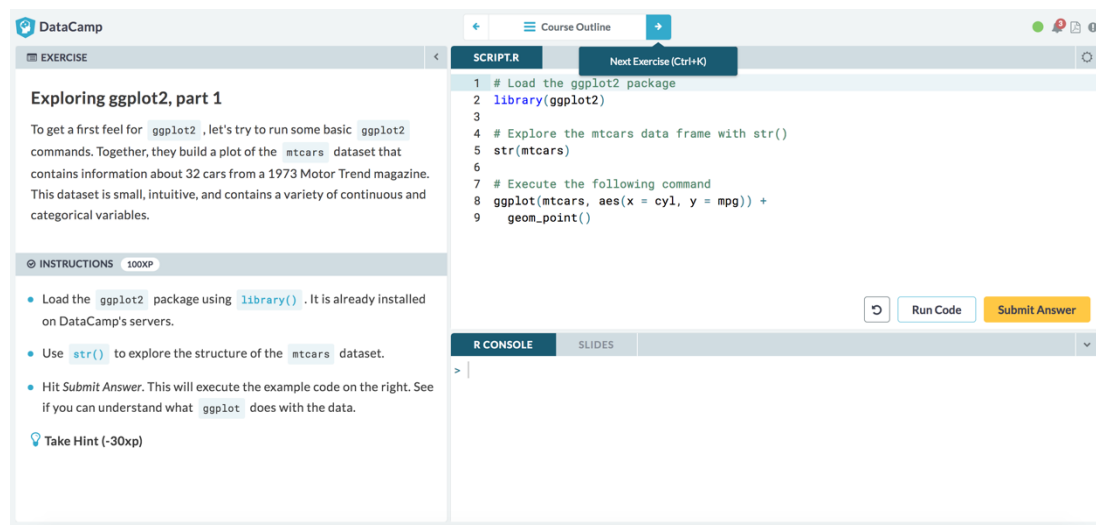
DataCamp (veeb, App Store, Google Play) – tegemist on veebikeskkonnaga, mis viib kokku õpilased ning oma ala spetsialistid, pakkudes neile keskkonda, kus luua põhjalikke tasuta ja tasuta avalike programmeerimisalaseid õppematerjale ning annab sellega võimaluse õppida SQL'i, R'i ja Python'it vastavalt siis andmeanalüüsi, statistikat või masinõpet. Antud rakendust saab kasutada nii veebikeskkonnas, kui ka mõningate erinevustega tasuta alla tõmmata Google Play¹² ning App Store¹³ poodides.

Tuleb luua endale kasutajakonto ning seejärel pakutakse veebilehel erinevate õppematerjalide juurde sissejuhatavaid õpetlike videoid ja koodikirjutamise väljakutseid lünkade täitmise ja koodi täiendamise näol (vt joonis 1). Ülesannete täitmisel kasutatakse eelnevalt videos saadud teadmisi, kui ka sammulisi juhiseid, mida täiendust vajava koodi kõrval näha saab. Motivatsiooniks ning tagasisideks antakse ülesannete lahendamise eest punkte. Lehel saab luua ka erinevaid projekte,

¹² <https://play.google.com/store/apps?hl=en>

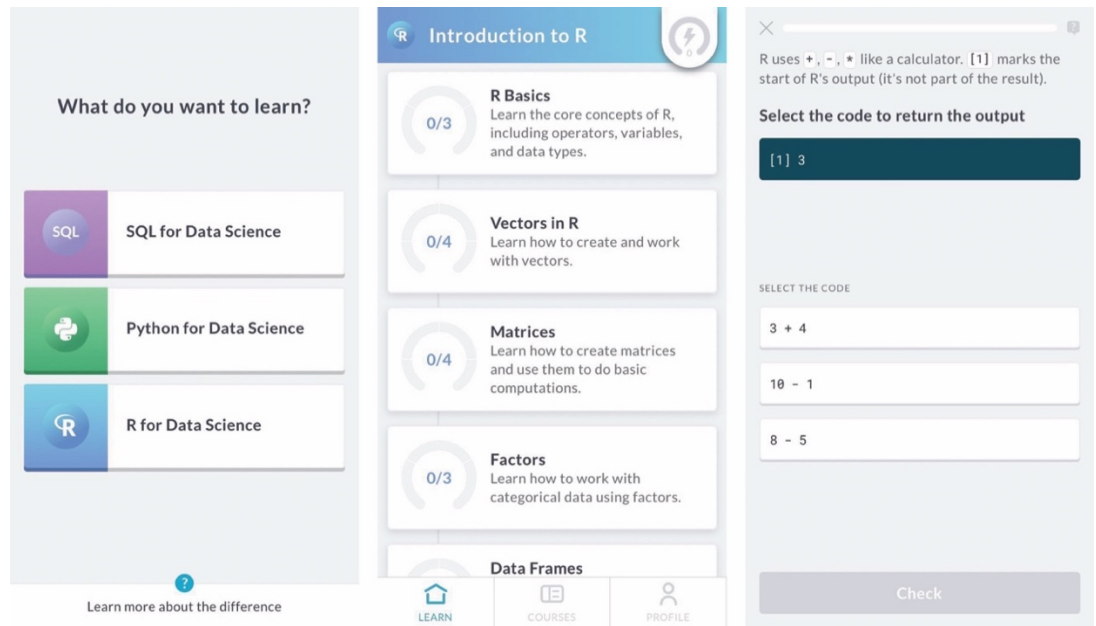
¹³ <https://itunes.apple.com/us/genre/mac/id39?mt=12>

millest on võimalik osa võtta ning samuti on võimalus moodustada õpigruppe. Lisaks on olemas foorumi keskkond, mis sisaldab erinevaid programmeerimisalaseid uudiseid, õpetusi, spikreid, avatud kursuseid, lindistusi ja blogipostitusi.



Joonis 1. DataCamp koodiülesande näide

Sama loodud kasutajakontoga kaudu saab siseneda ka läbi tahvelarvutitele ning nutitelefonidele mõeldud äppide. Küll aga on DataCamp äpi keskkond kompaktsemalt üles ehitatud. Saab valida, mis teadmisi omandada tahetakse ning ette antakse mini-ülesanded koodi parandamiseks ja täiendamiseks koos selleks vastavate juhistega, mida on võimalik lahendada 5 minutiga. Äpp näitab, kui palju on mingist ülesandest täidetud ning jälgib ka seda, kas päeva jooksul on õpilane mingi ülesande läbi teinud (vt joonis 2).

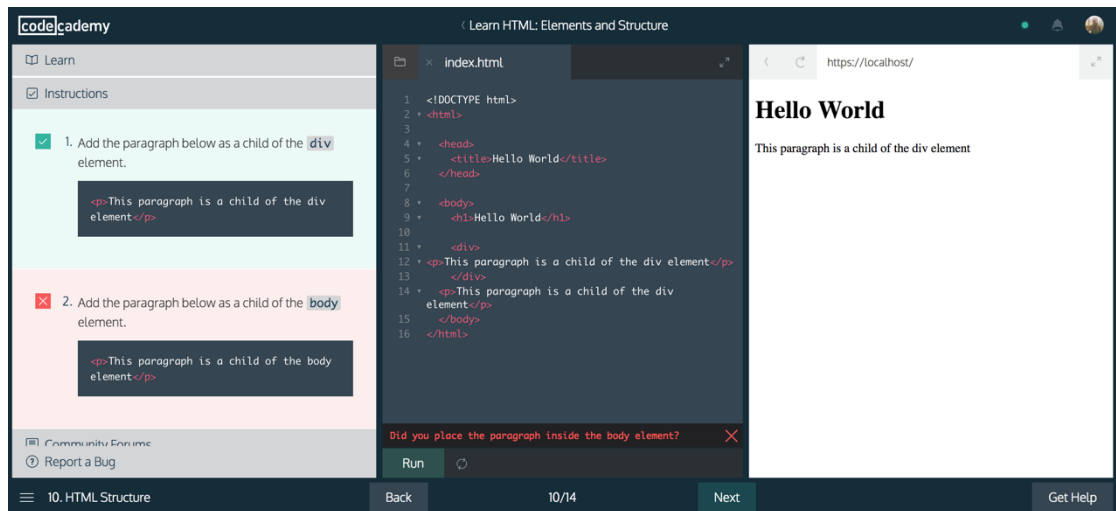


Joonis 2. DataCamp äpi valikud ja ülesande näide

Codecademy¹⁴ (veeb) – Codecademy on veebipõhine õppekeskkond, mis sarnaneb DataCamp'ile, kuid pakub lisaks SQL'ile, R'ile ja Python'ile veel Java, PHP, JavaScript'i (jQuery, AngularJS, React.js), Ruby, Sass'i ja HTML'i ning CSS'i programmeerimiskeele õpetusi. Küsimuste ning info edastamiseks on lehel olemas oma foorum. Codecademy pakub ka tasuta "pro" versiooni, mis annab õppijale ligipääsu isikustatud õppeplaanile, enesekontrolli küsimustikele, reaalsele projektidele ja kohest abi juhendajatelt.

Ülesande lahendamiseks on vaja eelnevalt läbi lugeda koodi kõrval olev sissejuhatav tekst ning järgida seejärel antud juhiseid, kui õpilane peaks koodi valesti sisestama saab ta sellele kohest tagasisidet (vt joonis 3).

¹⁴ <https://www.codecademy.com/>



Joonis 3. Codecademy ülesande näide

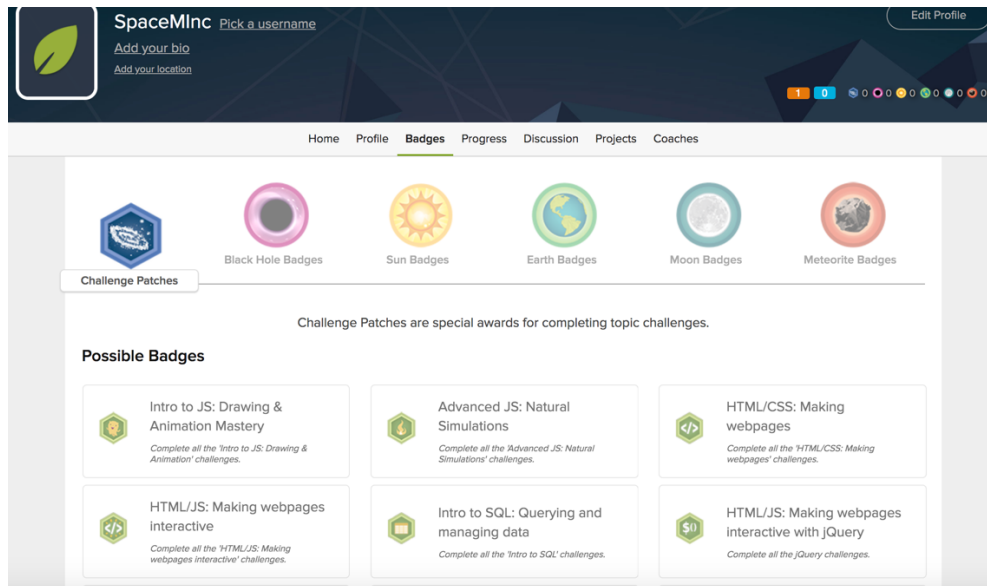
Khan Academy (veeb, App Store, Google Play) – Khan Academy on aastal 2006 õpetaja Salman Khani loodud mittetulunduslik haridusorganisatsioon, mis pakub lühiloenguid YouTube'i videote kaudu paljudest erinevatest valdkondadest ning täiendavaid praktilisi harjutusi ja materjale nii õpilastele kui ka õpetajatele. Õppematerjalid on valdavalt ingliskeelsed, kuid on olemas ka teistes keeltes materjale.

Antud veebikeskkond pakub laias valikus õppematerjale paljudest erinevatest valdkondadest, sealhulgas ka lai valik programmeerimisalaseid materjale, näiteks SQL, HTML, CSS, JavaScript, Python ning kuidas neid kasutada veebilehtede, mängude ja jooniste tegemiseks (vt joonis 4).

Joonis 4. Khan Academy koodi kirjutamis ülesanne

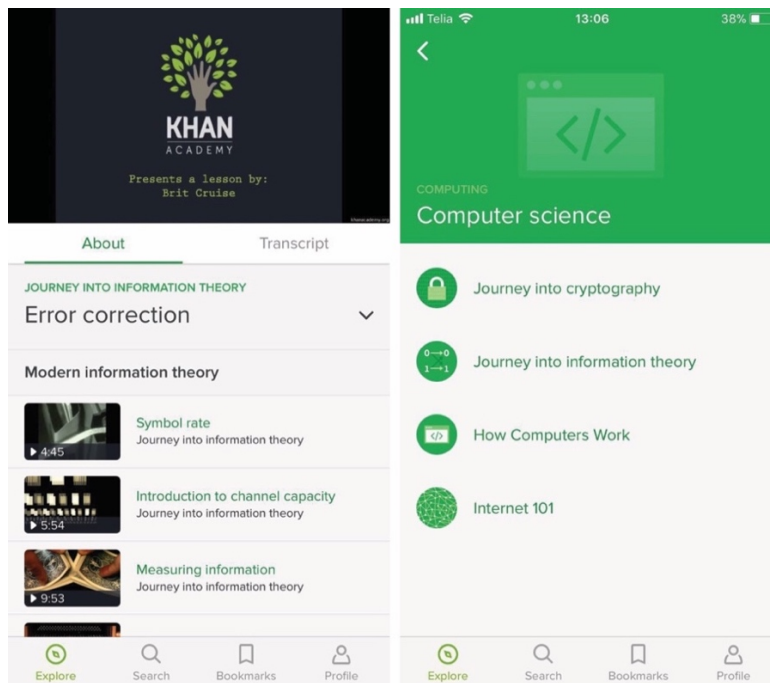
Khan Academy’l on olemas foorum, kus õpilane ning õpetaja saavad omavahel suhelda. Vanemad ning õpetajad saavad luua veebisiseseid klassiruumi ning lisada sinna õpilasi ja seejärel jälgida nende õppeprotsessi, anda ülesandeid ning tagasisidet (vt joonis 5). 2010. aastal lisati veebilehele ka palju erinevaid digitaalseid tunnustusmärke ehk õpimärke ning punktidekogumissüsteem, mis lisab õppimisele mängulisust ning sellega ka õpilastele motivatsiooni juurde (vt joonis 6).

Joonis 5. Khan Academi "klassiruumi" võimalused



Joonis 6. Khan Academy digitaalsed tunnustusmärgid

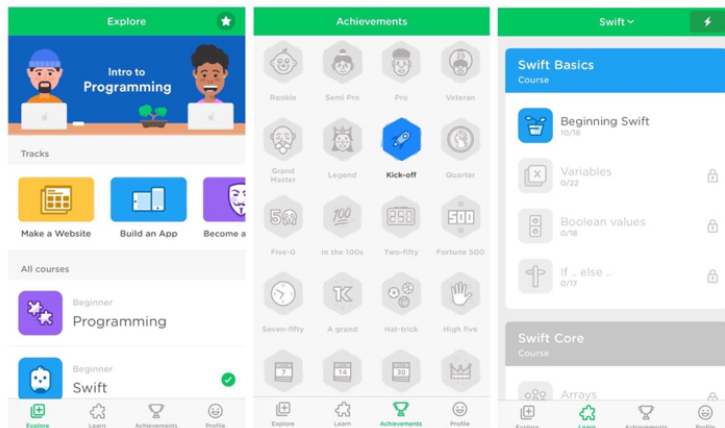
Khan Academy äpp võimaldab vaadata õpetlikke videoid erinevatel meelepärastel teemadel, kuid programmeerimisalastest õppematerjalidest on äppi toodud väga väike valik võrreldes veebilehel olevate teemadega (vt joonis 7).



Joonis 7. Khan Academy äpp

Mimo¹⁵ (App Store) – Mimo on ingliskeelne sissejuhatav äpp programmeerimisest algajatele ning õpetab äppide ja veebilehtede loomist, infoturvet ja palju muud taolist, tehes seda leidlikul ning interaktiivsel viisil ja andes õppijale võimaluse läbida ülesandeid kus iganes ta ka ei viibiks. Tegemist on väikeste, lihtsate programmeerimisülesannetega, mis ei võta rohkem kui minuti. Iga ülesanne sisaldab endas sissejuhatavat teksti ja seejärel koodi, mida juhistele vastavalt töödelda ning pärast koodi esitamist ka tagasisidet tehtud töö kohta. Äpi kasutaja läbib ülesandeid, saab selle eest punkte ja digitaalseid tunnustusmärke (vt joonis 8).

Äpi sisu pidevalt uuendatakse ja täiendatakse ning sellega on lubatud tulevikus sisse tuua teemasid nagu tehisintellekt, andmeteadus, riistvara, masinõpe. Teatud osa ülesandeid on äpis kõigile tasuta ligipääsetavad ning kui tahetakse rohkem juurde õppida antud äpi kaudu, siis lisandub sellele kas kuu- või aastamaks. Äpis on võimalik õppida programmeerimise aluseid, äppide, mängude, ja veebilehtede loomist, ülesannete automatiseerimist, küberkaitset ning erinevaid keeli nagu Swift, HTML, CSS, JavaScript, SQL, Python, Ruby, PHP, Git, Java, C#, C++ ja R.



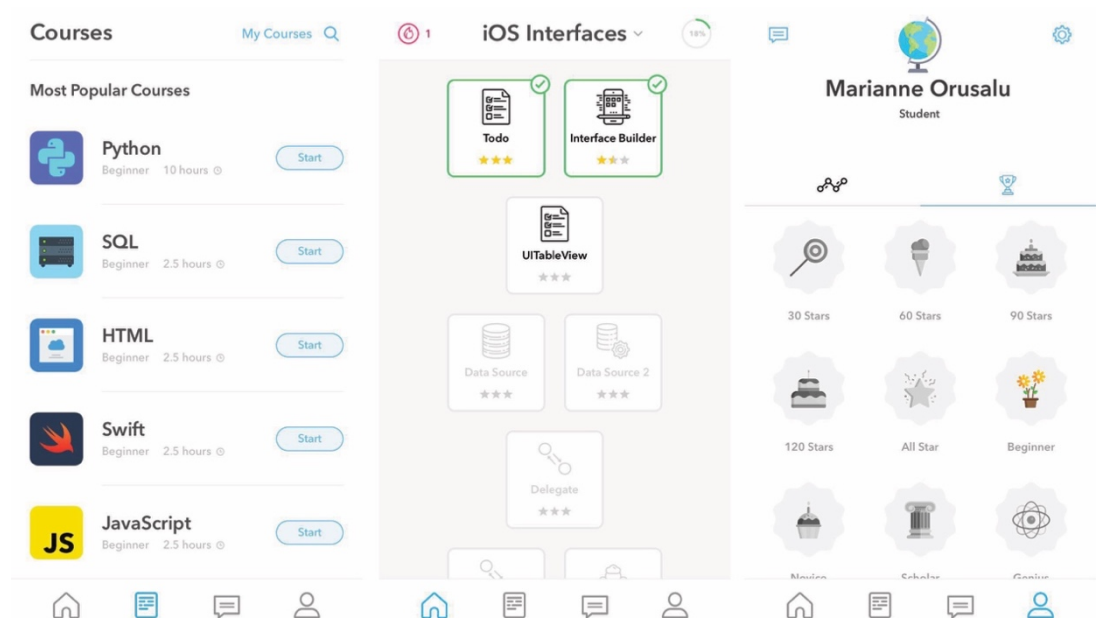
Joonis 8. Mimo äpp

¹⁵ <https://getmimo.com/>

Py¹⁶ (App Store ja Google Play) – Py on ingliskeelne äpp, mis annab võimaluse õppida väikeste, lihtsate programmeerimisülesannete näol kuidas luua äppe, veebilehti, analüüsida andmeid ning tutvustab sellega palju erinevaid programmeerimiskeeli. Py on sarnane ülesehituselt eelnevalt mainitud Mimo'ga, kus peab parandama ja täiendama koodi, kirjutama programme ja vastama õpitu põhjal küsimustele. Py äpp üritab välja selgitada õpilase nõrgad küljed ning pöörab nendele rohkem tähelepanu ning äpis on võimalus ka seada endale päevane eesmärk ning koguda ülesannete lahendamise eest punkte ja digitaalseid tunnustusmärke (vt joonis 9).

Py tasuta versioon pakub kohest personaalset abi õpetajatelt ükskõik mis kell ning suuremamahulisemat kogumit õppematerjalidele, millele jooksvalt juurde lisatakse.

Äpis on võimalik õppida äppide, mängude, ja veebilehtede loomist ning erinevaid keeli nagu Swift, HTML, CSS, JavaScript, SQL, Python, PHP, Git, Java, C#, C++ ja R.



Joonis 9. Py äpp

¹⁶ <https://www.getpy.com>

2.2. Programmeerimisalased õpimängud ja simulatsioonid

CodeSpells¹⁷ (Steam¹⁸: Windows & Mac OS X) – CodeSpells on tasuline mäng, mis kombineerib omavahel kujutlusvõime, loomingulise vabaduse ning arvutimängude kaasahaarava interaktiivsuse, tehes seda läbi koodi kirjutamise. Mängijaid võib vaadelda kui õpilasi, kes selmet kasutada eelnevalt valmis tehtud mänguelemente, saavad proovida kätt enda valmiskirjutatud loitsudega (ingl *spells*). Loitsude kirjutamine toimub Javascript keeles visuaalse vahendi nimega Blockly¹⁹ abil (vt joonis 10; joonis 11). Kuna tegemist on varajaselt väljastatud mängiga ei ole selles kõiki funktsioone, mida mängu valmistajad plaaninud on.



Joonis 10. CodeSpells mängu ekraanitõmmised



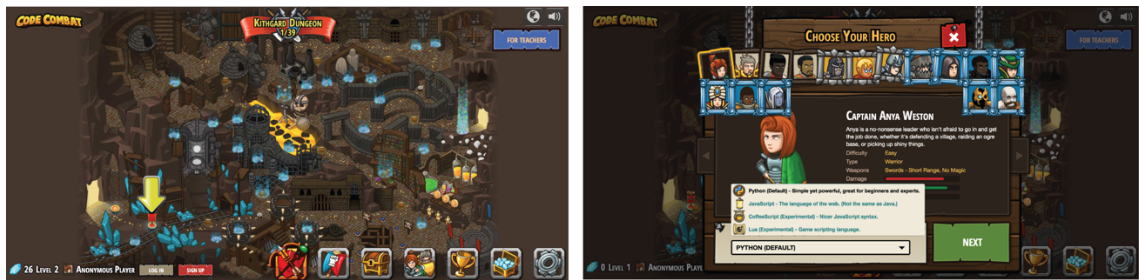
Joonis 11. CodeSpells loitsude valmistamise (koodi kirjutamise) ekraanitõmmised

¹⁷ <https://codespells.org/>

¹⁸ <http://store.steampowered.com/>

¹⁹ <https://developers.google.com/blockly/>

CodeCombat²⁰ (veeb) – Tegemist on 2013. aastal avaldatud veebipõhise rollimänguga, mis õpetab ühtlasi mängijaid Javascript ja Python keeles koodi kirjutama ning tutvustab programmeerimise aluseid (vt joonis 12). Mängus tuleb valida endale meelepärane tegelane ning programmeerimiskeel ning selleks, et mängus edasi jõuda peavad mängijad tõestama oma teadmiste kinnitumist koodi kirjutades (vt joonis 13), mida on võimalik teha nii üksikmängijana kui ka meeskonnas. Mäng on peamiselt suunatud põhikooliealistele õpilastele, kuid mängu- ja teadmiste omandamise rõõmu võib see siiski pakkuda kõigile. Hiljem lisasid mängu tegijad sisse ka tasemete loomise (ingl *level editor*) võimaluse, mis võimaldab kasutajatel luua ise mängu sisu. Mäng on kättesaadav tasuta, kuid kuumaksega on võimalik omandada ligipääs täiendavale mängu sisule.



Joonis 12. CodeCombat kaart ja valikud



Joonis 13. CodeCombat mängu käik

²⁰ <https://codecombat.com/>

Colobot²¹ (Windows ja Linux) – Colobot on reaaliajase 3D strateegiamäng, kus mängu elemente saab manipuleerida koodi kirjutamise teel CBOT keeles, mis on sarnane C++ ja Java keelele. Näiteks peab õpilane kirjutama lihtsa *forloop* koodi, et hävitada sihtmärgid, mis asetsevad võrdse vahemaaga üksteise kõrval. Colobot aitab kaasa koodis leiduvate süntaksite esiletõstmisega ning trükivigade ette näitamisega (vt joonis 14). Mängu missiooniks on leida asustamiseks uus planeet luues pinna baasinfrastruktuuri ja kõrvaldades missiooni ohustavate tulnukate eluvormid. Mäng leiab aset maal, kuul ja seitsmel väljamõeldud planeedil.



Joonis 14. Colobot koodi näide

²¹ <https://colobot.info/>

2.3. TLÜ üliõpilastööde raames loodud programmeerimisalased õppematerjalid

All väljatoodud Tallinna Ülikooli üliõpilastööd on vaadatud Digitehnoloogiate instituudi teemaderegistri²² lehelt (vt tabel 1). Iga töö puhul on välja toodud head lahendused õppematerjali esitamisel ning kasutamata jäetud võimalused.

Tabel 1. TLÜ üliõpilastööde analüüs

<u>Üliõpilastöö</u>	<u>Head lahendused</u>	<u>Kasutamata võimalused</u>
Romil Rõbtšenkov – Rakenduse loomine iOS operatsioonisüsteemiga seadme jaoks	<ul style="list-style-type: none"> • Arenduskeskkonna tutvustus • Koodinäited • Iseseisvad ülesanded • Ekraanitõmmised • Valmis realselt kasutatav, vajalik äpp • Testimise tagasiside 	<ul style="list-style-type: none"> • Videod • Veebikeskkond • GitHub
Erki Sidron – Semantic UI õppematerjal	<ul style="list-style-type: none"> • Koodinäited • Ülesanded • Testimise tagasiside 	<ul style="list-style-type: none"> • Visuaalne näide koodi järel, milline lõpptulemus välja peab nägema • Videod
Sven-Kristjan Kompus – Tarkvara ehitust automatiseeriva tööriista Gradle kohta õppematerjali loomine Java veebirakenduse põhjal	<ul style="list-style-type: none"> • Põhjalik sissejuhatus • Programmi paigalduse juhend • Koodinäited • Ülesanded 	<ul style="list-style-type: none"> • Videod • Tagasiside • GitHub • Veebikeskkond

²² <http://www.cs.tlu.ee/teemaderegister/>

Sander Leetus – Dota 2 Workshop Tools õppematerjal kohandatud mängude loomiseks	<ul style="list-style-type: none"> • Programmi paigaldamise juhend • Tööriistade tutvustus • Ekraanitõmmised 	<ul style="list-style-type: none"> • Videod • Tagasiside • Testid ja ülesanded õpitu põhjal • Veebikeskkond
Magnus Kvell – Lingvistika analüüs R-keele abil: õppematerjal	<ul style="list-style-type: none"> • Koodinäited 	<ul style="list-style-type: none"> • Tagasiside • Pildid • Videod • GitHub • Veebikeskkond
Priit Mattus - Xamarin ja Mvvmcross iOS ja Android rakenduste loomiseks. Õppematerjal.	<ul style="list-style-type: none"> • Põhjalik sissejuhatus • Videod • Tagasiside 	<ul style="list-style-type: none"> • Enese testimine • Veebikeskkond
Harry Kaarma – Sissejuhatav õppematerjal R-keelde	<ul style="list-style-type: none"> • Ekraanitõmmised • Põhjalik sissejuhatus • Koodinäited • Harjutus ülesanded 	<ul style="list-style-type: none"> • Videod • Laialdasem testimine • GitHub • Veebikeskkond
Lauri Valma – Solid Edge õppematerjal (Näidismudel „Segisti“ loomine)	<ul style="list-style-type: none"> • Ekraanitõmmised • Samm sammuline ülesehitus 	<ul style="list-style-type: none"> • Videod • Veebikeskkond
Toomas Häide – Marmalade raamistiku õppematerjal C++ keeles	<ul style="list-style-type: none"> • Põhjalik sissejuhatus • Ekraanitõmmised • Koodinäited 	<ul style="list-style-type: none"> • Tagasiside • GitHub • Veebikeskkond

Siim Suu – Mängumootori Unity 3D õppematerjal	<ul style="list-style-type: none"> • Ekraanitõmmised • Koodinäited 	<ul style="list-style-type: none"> • Videod • GitHub
Hainer Savimaa – ZURB Foundation 5 Õppematerjal	<ul style="list-style-type: none"> • Ekraanitõmmised • Lisa veebiõpetuste soovitusel • Koodinäited • Ülesanded • Näidislehestik • Põhjalik tagasiside 	<ul style="list-style-type: none"> • Videod • GitHub • Veebikeskkond
Sander Leetus – AngularJS raamistiku õppematerjal	<ul style="list-style-type: none"> • Ekraanitõmmised • Koodinäited • Ülesanded 	<ul style="list-style-type: none"> • Videod • Tagasiside • GitHub • Veebikeskkond
Mirko Valkenpert – Source Filmmaker'i õppematerjal	<ul style="list-style-type: none"> • Ekraanitõmmised • Tagasiside • Põhjalik materjal 	<ul style="list-style-type: none"> • Videod • Veebikeskkond
Raner Piibur – WebGL'i kasutamine interaktiivsete graafikarakenduste loomiseks veebilehitsejas: õppematerjal	<ul style="list-style-type: none"> • Põhjalik sissejuhatus ja materjal • Koodinäited • Ekraanitõmmised • Lisamaterjal • Märkused • Ülesanded • Praktika • Juhised • Näitab, milline peab tulemus välja nägema (piltidega) 	<ul style="list-style-type: none"> • Videod • Veebikeskkond

Joonas Helde – Õppematerjal Adobe Edge Animate'i kasutamiseks interaktiivsete rakenduste loomisel	<ul style="list-style-type: none"> • Teiste materjalide ülevaade • Ekraanitõmmised • Ülesanded 	<ul style="list-style-type: none"> • Videod • Veebikeskkond
Tambet Paljasma – 3D mängude loomine XNA keskkonnas. Õppematerjal	<ul style="list-style-type: none"> • Intervjuu õppejõuga • Põhjalik testimine • Ekraanitõmmised • Koodinäited 	<ul style="list-style-type: none"> • Videod • GitHub • Veebikeskkond
Raimo Virolainen – Objektorienteeritud veebilahenduste loomise õppematerjal ASP.NET MVC raamistiku abil	<ul style="list-style-type: none"> • Olemasolevate materjalide väljatoomine. • Ekraanitõmmised • Koodinäited 	<ul style="list-style-type: none"> • Videod • GitHub • Veebikeskkond
Marten Rebane – Meteor.js raamistiku eestikeelne õppematerjal	<ul style="list-style-type: none"> • Videotele viitamine • Ekraanitõmmised • Koodinäited • Tagasiside 	<ul style="list-style-type: none"> • Ise tehtud videod • Laialdasem tagasiside • GitHub • Veebikeskkond

Valdavaks probleemiks autori poolt läbi vaadatud õppematerjalides on nende esitamine seminaritöö või bakalaureusetöö PDF failina, mis muudab nende praktiseerimise ebamugavamaks ning ei anna palju ruumi selleks, et kasutada kõiki võimalikke vahendeid õpitava parimaks omandamiseks. Sellise probleemi saaks lahendada õppematerjalile veebilehe loomisega. Samuti on PDF'ist ebamugav koodinäiteid kopeerida ning selleks oleks lihtsam kasutada õppematerjali osana GitHub'i. Vaadeldud õppematerjalide seas on ainult väheseid, mis kasutavad ekraanivideote võimalusi ning enesekontrolliteste.

3. Õppematerjali kavandamine

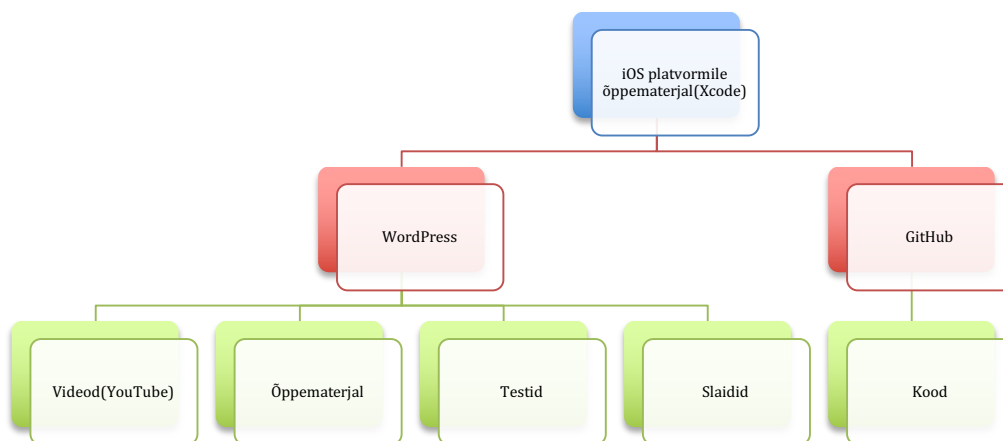
Antud seminaritöö põhjal on autoril plaanis luua bakalaureusetöö raames programmeerimisalane õppematerjal iOS platvormile, mis sisaldaks erinevaid multimeedia võimalusi (vt joonis 15). Selleks on autor esmalt loonud ADDIE mudelit kasutades lühikese tegevuskava õppematerjali koostamiseks (vt tabel 2).

Õppematerjali sissejuhatuses tuuakse välja kellele on õppematerjal mõeldud, milline on õppematerjali ülesehitus, millised peaksid olema vajalikud eelteadmised ning ülevaade õppematerjali käigus omandatavatest oskustest.

Esmalt tutvustab autor iOS operatsioonisüsteemi ning arenduskeskkonda Xcode ja annab ülevaate süsteeminõuetest ning kuidas programmi arvutisse paigaldada. Samuti tutvustab autor siin õppematerjali veebilehte ning GitHub keskkonda.

Järgmisena annab autor ülevaate õppematerjali lõpuks valmivast rakendusest ning sellele järgneb samm-sammuline äpi loomise õpetus, kasutades selleks õpetlikke videoid, pilte ja koodinäiteid ning viimaks iseseisvad ülesanded ning testid.

Lõpus teeb autor lühikese kokkuvõtte õppematerjalis loodud rakenduse ja õpilase omandatud teadmiste kohta ning annab edasiõppimise soovitusi ja toob välja õppematerjali edasiarendamis võimalusi.



Joonis 15. Õppematerjali sisukaart

Tabel 2. Õpimisaini tegevuste tabel

Tegevused õppematerjali koostamisel ADDIE mudelit kasutades	
Analüüsi etapi tegevused	<p>Koolitusvajaduse kindlakstegemine: Vähene või üldse puudulik eestikeelsete programmeerimisalaste õppematerjalide hulk iOS platvormile.</p> <p>Õpieesmärkide määratlemine: Õppija omandab algteadmised Swift keele kohta ning oskuse koostada koodinäidete eeskujul lihtsamaid iOS rakendusi.</p> <p>Sihtgrupi analüüs: Sihtgrupiks on infotehnoloogia suunaga tudengid ning programmeerimishuvilised inimesed.</p> <p>Tehnilise platvormi valik: Töö toimub Mac arvutites Xcode programmis, kasutades Swift keelt.</p> <p>Õppematerjalidele saab ligi WordPress veebikeskkonnas ning GitHub'is.</p>
Kavandamise etapi tegevused	<p>Õpiväljundite sõnastamine:</p> <ul style="list-style-type: none"> • oskab kasutada XCode arenduskeskkonda uue projekti loomiseks, koodi sisestamiseks ja koodi käivitamiseks simulaatoris; • tunneb Swift keele süntaksit; • on võimeline koostama koodinäidete põhjal lihtsamaid iOS rakendusi Swift keeles. <p>Kursuse sisu kavandamine (esitatud tabelis 3)</p>

Selles seminaritöös käsitletud andmete põhjal on autor kokku pannud lühikese kursuse sisu kavandi, mis edasises töös võib muutuda ja täieneda (vt tabel 3).

Tabel 3. Kursuse sisu kavand

iOS ÕPPEMATERJAL ÄPI LOOMISEKS	
Tund 1. Ülevaade	
<ol style="list-style-type: none"> 1. Sissejuhatus 2. Arenduskeskkonna ja süsteemi algne tutvustus ja paigaldamine 3. Programmeerimiskeele tutvustus 	<p>Esimeses tunnis antakse ülevaade, mida antud õppematerjal endast kujutab, milleks see vajalik on, mis on eesmärgid, kellele see suunatud on ning mis platvormidel õppimine toimub.</p> <p>Tutvustatakse Xcode'i, süsteeminõudeid ning antakse paigaldamise/uuendamise õpetus.</p> <p>Antakse ülevaade Swift keelest.</p> <p>Tutvustatakse lühidalt õppematerjali lõpuks valmivat rakendust.</p> <p>Õppematerjali sisutüübid:</p> <ul style="list-style-type: none"> • tekst WordPress lehekülgedena; • XCode keskkonda tutvustav ekraanivideo; • enesekontrolli testid Swift keele süntaksi kohta.
Tund 2. Algsed teadmised	
<ol style="list-style-type: none"> 1. Arenduskeskkonna tutvustus 2. Arenduskeskkonna kohandamine 3. Projekti loomine 4. Simulaatori tutvustamine 	<p>Tutvustatakse Xcode'i lähemalt, selle funktsioone ja kasutatavaid põhielemente.</p> <p>Õpetatakse Xcode'i endale sobivaks kohandama.</p> <p>Tutvustatakse iOS simulaatorit.</p>

	<p>Luuakse uus projekt äppi loomiseks.</p> <p>Õppematerjali sisutüübid:</p> <ul style="list-style-type: none"> • tekst WordPress lehekülgedena; • XCode keskkonda tutvustavad ekraanivideod (XCode keskkonna kohandamine, projekti loomine, simulaatori kasutamine).
Tund 3. Sisu loomine	
<ol style="list-style-type: none"> 1. Projekti arendamine 2. Funktsionaalsuse tagamine 	<p>Kasutajaliidese loomine, millele lisatakse vajaminevad elemendid ning funktsionaalsused.</p> <p>Õppematerjali sisutüübid:</p> <ul style="list-style-type: none"> • tekst WordPress lehekülgedena; • koodinäited GitHubis; • allalaetav XCode projekt.
Tund 4. Testimine	<p>Ülesannete andmine ja teadmiste testimine.</p> <p>Õppematerjali sisutüübid:</p> <ul style="list-style-type: none"> • enesekontrolli testid.

Õppematerjali platvormiks on plaan kasutada WordPress keskkonda, kus õppematerjali tekstiline osa esitatakse lehekülgedena, mille sees kuvatakse vajalikke faile, slide ja YouTube's avaldatud ekraanisalvestusi Xcode kasutamise kohta. Koodinäited pannakse ülesse GitHub'i, kus õppijad saavad alla laadida ka näidisprojekti Xcode formaadis. Enesekontrolli testide lisamise vahend täpsustub, kui on selge, milliseid küsimuste tüüpe testides kasutatakse.

Materjali läbimine võib õppijal aega võtta umbes 5-8 tundi.

Kokkuvõte

Käesoleva seminaritöö eesmärgiks oli anda ülevaade erinevatest võimalustest programmeerimisalaste õppematerjalide loomisel, mis kasutavad multimeediumi võimalusi ning kavandada lihtsa eestikeelse iOS arenduse õppematerjali ülesehitus.

Seminaritöö esimese tulemusena annab autor kirjanduse analüüsi põhjal ülevaate erinevatest võimalustest programmeerimisalaste õppematerjalide loomisel, keskendudes multimeediumi ja interaktiivsete lahenduste kasutusvõimalustele.

Teise tulemusena toob autor välja valiku programmeerimisalaseid õppematerjale, mis kasutavad erinevaid multimeediumi vahendeid, saades sellega põhjaliku ülevaate mujal maailmas kasutatud headest võimalustest programmeerimise õpetamisel. Samuti analüüsis autor TLÜ üliõpilastööde raames koostatud programmeerimisalaseid õppematerjale. Tehtud analüüs õppematerjalide headest lahendustest ning kasutamata võimalustest andis autorile ülevaate, mida paremini võiks teha eestikeelse programmeerimisalase õppematerjali loomisel.

Kolmandas peatükis loob autor kogutud infot arvesse võttes lühikese õppematerjali kavandi iOS platvormile. Kavandatud õppematerjal on jaotatud neljaks tunniks ning sisaldab teksti, ekraanivideoid, koodinäiteid ja enesekontrolli teste.

Autor näeb antud seminaritöö edasiarendamise võimalusena loodud õppematerjali kavandi täiendamist, mille eesmärgiks on valmis saada realselt rakendatav programmeerimisalane õppematerjal iOS platvormile, mis kasutab erinevaid sobilikke multimeediavahendeid.

Kasutatud kirjandus

- Aureliano, V.C.O. (2014). Self-explaining from videos as a methodology for learning programming. B. du Boulay & J. Good (toim), *Psychology of Programming Interest Group Annual Conference 2014* (lk 209–214).
<http://www.ppig.org/sites/ppig.org/files/2014-PPIG-25th-proceedings.pdf>
- Branch, R. M. (2009). *Instructional Design: The ADDIE Approach*. New York, NY: Springer.
- Fotaris, P., Mastoras, T., Leinfellner, R., & Yasmine, R. (2015). From hiscore to high marks: Empirical study of teaching programming through gamification. R. Munkvold & L. Kolås (toim), *Proceedings of the 9th European Conference on Games Based Learning ECGBL 2015* (lk 186–194).
http://eprints.uwe.ac.uk/28599/3/From_Hiscore_to_High_Marks_Empirical_Stu%20%281%29.pdf
- Freeman, A., Adams Becker, S., Cummins, M., Davis, A., & Hall Giesinger, C. (2017). *NMC/CoSN Horizon Report: 2017 K–12 Edition*. Austin, TX: The New Media Consortium.
- HITSA. (2015). *Programm ProgeTiiger 2015–2017*. Loetud aadressil
http://media.voog.com/0000/0034/3577/files/Programm%20ProgeTiiger%202015_2017.pdf
- Manley, E.D., & Urness, T.M. (2014). Video-based instruction for introductory computer programming. *Journal of Computing Sciences in Colleges*, 29(5), 221–227.
- Morrison, B.B., & Disalvo, B.J. (2014). Khan academy gamifies computer science. *Proceedings of the 45th ACM technical symposium on Computer science education* (lk 39–44). New York, NY: ACM.
<http://dx.doi.org/10.1145/2538862.2538946>
- Nørmark, K. (2014). Using Short Videos in an Introductory Programming Course. *International Conference on e-Learning '14* (lk 254–260). <http://elearning-conf.eu/docs/cp14/paper-38.pdf>
- Rinde, A. (2010). *Multimeediumipõhiste õppematerjalide loomise raamistik* (magistritöö). Loetud aadressil <http://www.cs.tlu.ee/teemaderegister/>
- SA Kutsekoda. (2016). *Tulevikuvaade tööjõu- ja oskuste vajadusele: info- ja kommunikatsioonitehnoloogia*. Loetud aadressil <http://oska.kutsekoda.ee/wp-content/uploads/2016/04/IKT-Raport-loplik.pdf>

- Siemens, G. (2002) *Instructional Design in Elearning*. Loetud aadressil <http://www.elearnspace.org/Articles/InstructionalDesign.htm>
- Sorva, J. (2012). *Visual program simulation in introductory programming education*. (doktoritöö). Loetud aadressil <https://aaltodoc.aalto.fi/handle/123456789/3534>
- Tillmann, N., Halleux, J., Xie, T., Gulwani, S. & Bishop, J. (2013). Teaching and Learning Programming and Software Engineering via Interactive Gaming. D. Notkin, B. H. C. Cheng, & K. Pohl (toim), *35th International Conference on Software Engineering* (lk 1117–1126). <http://taoxie.cs.illinois.edu/publications/icse13see-pex4fun.pdf>
- Villems, A., Aluoja, L., Pilt, L., Naulainen, M.-M., Kusmin, M., Rogalevitš, V., & Tokko, U. (2015). *Digitaalse õppematerjali loomise soovitused: Juhend digitaalse õppematerjali autorile*. Loetud aadressil <http://oppevara.hitsa.ee/kvaliteet/>
- Warren, J., Rixner, S., Greiner, J., & Wong, S. (2014). Facilitating human interaction in an online programming course. *Proceedings of the 45th ACM technical symposium on Computer science education* (lk 665–670). New York, NY: ACM. <http://dx.doi.org/10.1145/2538862.2538893>
- Yuen, A.H.K. (2006). Learning to program through interactive simulation. *Educational Media International*, 43(3), 251–268. <http://dx.doi.org/10.1080/09523980600641452>