

Tallinna Ülikool  
Digitehnoloogiaste instituut

# **RAKENDUSE ARENDUSE ÕPPEMATERJAL iOS PLATVORMILE**

Bakalaureusetöö

Autor: Marianne Kristiina Orusalu

Juhendaja: Hans Põldoja

Autor:..... ”.....” ..... 2018

Juhendaja:..... ”.....” ..... 2018

Tallinn 2018

## **Autorideklaratsioon**

Deklareerin, et käesolev bakalaureusetöö 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.....	5
1. Programmeerimisalaste õppematerjalide õpidisaini põhimõtted.....	8
1.1. Videod.....	8
1.2. Enesekontrollitendid.....	10
1.3. Simulatsioonid.....	12
1.4. Mängupõhise õppe lahendused.....	13
1.5. Kogukonnas õppimist toetavad lahendused.....	14
2. Metoodika.....	17
3. Õppematerjali kavandamine.....	20
3.1. Õppematerjali sisu ja struktuuri kavandamine.....	20
3.1.1. Rakendus ja koodinäited.....	20
3.1.2. Õppematerjali sisu ja harjutuste kavandamine.....	21
3.2. Platvormi valik.....	22
3.3. Õppematerjali litsentsivalik.....	26
4. Õppematerjali väljatöötamine.....	27
4.1. Õppematerjali loomine.....	27
4.2. Videote ja testide loomine.....	28
5. Õppematerjalile hinnangu andmine.....	31
5.1. Ekspertide hinnang õppematerjali kvaliteedile.....	31
5.2. Uuringutulemuste ülevaade.....	32
Kokkuvõte.....	38
Kasutatud kirjandus.....	39
Summary.....	41

LISAD.....	42
Lisa 1. Kursuse sisu kavandi tabel seminaritööst .....	43
Lisa 2. WordPress platvormil õppematerjali ekraanipilt .....	45
Lisa 3. LORI hindamismudel ekspertidele .....	46
Lisa 4 Intervjuu ekspertidega.....	47
Lisa 5. Ekraanipilte GitHub platvormis olevast õppematerjalist.....	54

## Sissejuhatus

Programmeerimiskeeli on olemas mitmeid erinevaid ja arendajana nendest kõikidest mingi teadmise omamine on väga suureks eeliseks ning samas tuleb programmeerimisoskus kasuks mingil määral kõigile, sest see arendab loogilist mõtlemist, probleemide lahendamise oskust ja loovust. Suurt rolli uute programmeerimisalaste teadmiste omandamisel mängib algteadmiste tase, millest alustati, seega on oma tähtsus ka selles millist programmeerimiskeelt esimesena õppida ning kuidas seda tehakse. Alexander Petkov (2018), MyFirstITJob<sup>1</sup> asutaja, tegi 2018. aasta alguses läbi uuringu ning tõi sellega välja edetabeli programmeerimiskeeltest. Tegemist on objektiivse ja praktilise ülevaatega praegusest seisust, suundumustest ja tulevikuproгноosidest IT vallas. Selliseid uuringuid on tehtud veel mitmeid ning igatühes neist on antud bakalaureusetöös kasutatav Swift programmeerimiskeel edetabelis 10 eelistatuma keele seas. Antud uuringud annavad hea ülevaate sellest, milliseid keeli hetkeseisuga ja tulevikuvaates rohkem eelistatakse ning milliseid programmeerimiskeeli oleks kõige praktilisem omandada nii tööturu kui ka arendaja tulevikuperspektiive arvestades.

Programmeerimise õpetamise arendamiseks üldhariduskoolides on Eestis käivitatud ProgeTiigri programm, mille raames on loodud õppematerjale, hangitud koolidele robotikaseadmeid, koolitatud õpetajaid ning muuhulgas on seal olemas ka autori jaoks kasulik digitaalse õppematerjali loomise juhend ja soovitused (HITSA, 2015).

Eesti koolides pööratakse hetkel enim tähelepanu programmeerimise algõpetusele ja veebiprogrammeerimisele ning robotikale ja vähem tähelepanu saavad selle kõrvalt programmeerimisvõimalused mobiilirakenduste arendamiseks. Samas puutuvad noored igapäevaselt kokku just erinevate mobiilirakendustega. Nagu iga programmeerimisalase teadmise õpetamisega, on ka mobiilirakenduse arenduse

---

<sup>1</sup><https://www.myfirstitjob.com>

õpetamises väljakutseks teha nende teadmiste omandamine hästi arusaadavaks ning huvitavaks. Android arenduse mänguliseks ja põnevaks õpetamiseks on loodud MIT App Inventor<sup>2</sup> ning iOS arenduse õpetamiseks on näiteks loodud Swift Playgrounds<sup>3</sup>.

Antud bakalaureusetöö on jätkuks seminaritööle (Orusalu, 2018), mille raames autor uuris programmeerimisalaste õppematerjalide koostamise põhimõtteid ning leidis ühe tulemusena, 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 ja 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 keskendus õppematerjali algele kavandamisele ning õppematerjali loomine ja testimine toimub käesolevas bakalaureusetöös. Seminaritöö üheks tulemuseks oli ka kirjanduse analüüs multimeediumi ja interaktiivsete lahenduste kasutusvõimalustest programmeerimisalaste õppematerjalide koostamisel, kus autor vaatas üle ka teiste üliõpilastöödena valminud programmeerimisalaste õppematerjalides kasutatud head võimalused ja kasutamata jäetud lahendused. Käesolevas bakalaureusetöös on esitatud kokkuvõtte seminaritöös koostatud kirjanduse analüüsist ning toodud välja nende kasutamine käesolevas õppematerjalis.

Antud bakalaureusetöö eesmärgiks on läbi teha äpi arendus iOS platvormile ja selle põhjal luua eelnevalt valminud seminaritöö põhjal lihtsa eestikeelse iOS arenduse õppematerjal ning läbi teha tagasiside uuring ekspertidega, et kontrollida õppematerjali kvaliteeti.

---

<sup>2</sup> <https://appinventor.mit.edu>

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

Eesmärgi saavutamiseks püstitas autor järgmised ülesanded:

- koostada lihtne iOS mäng, mis on aluseks õppematerjali loomisele;
- koostada iOS rakenduse arenduse õppematerjal loodud mängu näitel;
- hinnata ekspertidega koostatud õppematerjali kvaliteeti ja panna kokku soovitud õppematerjali parandamiseks.

Bakalaureusetöö koosneb viiest peatükist. Esimeses peatükis toob autor välja järeldused ning ülevaate kirjanduseanalüüsi tulemustel leitud programmeerimisalaste õppematerjalide õpidisaini põhimõtetest ning tutvustab, mida ja kuidas ta kasutab enda õppematerjalis. Teises peatükis teeb autor ülevaate õppematerjali koostamiseks valitud meetodikast ning loob pildi sellest, mis valmis seminaritöös ja mida tehakse bakalaureusetöös. Kolmandas peatükis loob autor ülevaate õppematerjali kavandamise protsessist ning otsustest. Neljandas peatükis kirjutab autor õppematerjali väljatöötamise protsessist. Viendas osas teeb autor läbi õppematerjalile hinnangu andmise ekspertidega ning toob välja saadud tulemused.

Bakalaureusetöö käigus valminud õppematerjal on kättesaadav aadressilt <https://github.com/ios-rakenduse-arendus/EsimeApp/wiki>.

# **1. Programmeerimisalaste õppematerjalide õpidisaini põhimõtted**

Seminaritöö raames läbi viidud kirjanduse analüüsis selgus, et parimate õpitulemuste tagamiseks peaks kaasaegne programmeerimisalane õppematerjal sisaldama õpilasi kaasavaid multimeediumi võimalusi. Samuti tuleks digitaalsete õppematerjalide loomisel arvestada interaktiivsuse lisamisega. Autor tegi järeldused sellest, milliseid multimeediumi võimalusi saab kasutada luues programmeerimisalast õppematerjali iOS rakenduse arenduse kohta ning kuidas kasutada neid õpilaste motiveerimiseks nii, et nad ei oleks kõigest passiivsed õppematerjali tarbijad vaid aktiivsed kasutajad ja õppijad.

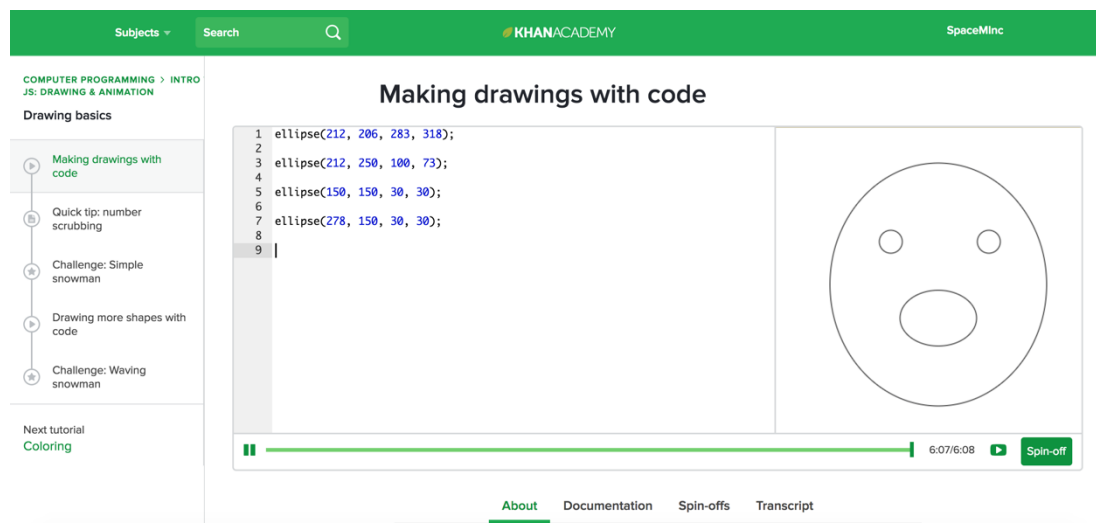
Järgnevalt on loodud kokkuvõtte leitud multimeediumi ja interaktiivsete lahenduste võimalustest programmeerimisalastes õppematerjalides ning nende kasutamisevajadus ja kasulikus antud iOS rakenduse arenduse õppematerjalis.

## **1.1. Videod**

Üheks parimaks ning levinuimaks multimeediumivahendiks, mida kasutada õppematerjalides, võib pidada videoid. Õppevideote kasutamine on eriti edukas algajatest programmeerijate seas ning samas eelistatud õppevorm ka teiste õpilaste hulgas. Edukas õppevideo peaks olema ajaliselt lühemas formaadis, sisukas ning seostama uut informatsiooni eelnevate teadmistega. Selliste õppevideote populaarsusele osutab suurelt ka nende massiline hulk YouTube'i kanalitel (Aureliano, 2014). USA Drake Ülikoolis uuringute põhjal selgus, et videopõhine õppimine paneb õpilasi nägema programmeerimist lõbusa ning huvitavana ka nende inimeste seas, kes sellega otseselt kokku ei ole puutunud (Manley & Urness, 2014).



Seminaritöös väljatoodud programmeerimisalastest õppematerjalidest kasutasid õppevideoid näiteks DataCamp<sup>4</sup> ja Khan Academy<sup>5</sup> (Joonis 1).



### ***Joonis 1. Khan Academy video kasutus***

Autor leiab, et videod on parim lahendus õpitava materjali kinnistumise tagamiseks, sest visuaalne jälgimine loob selge pildi ette õpitavast ning soovitatavast tulemusest ja samuti on võimalik videot seisma panna ning liikuda edasi ja tagasi materjalis kohta, mis võis veel segaseks jääda. Kõige paremaks õppimist toetavaks lahenduseks võib pidada lühivideoid, mida on lihtne jälgida. Lisaks sellele on ka hea ja tudengite poolt kiidetud lahendus, loengu või praktikumi lindistamine koos arvutiekraanil toimuvaga. See annab hiljem võimaluse õpitu kordamiseks ning puudulike teadmiste korral koheseks täiendamiseks. Videod on kasulik ja leidlik viis ka õpilastele kellel võib esineda keskendumisraskusi kirjaliku materjali või loengufurmaadis kursuse jälgimisega, kuid õpihimu on see-eest olemas. Seega otsustas autor iOS rakenduse arenduse õppematerjalis kasutusele võtta lühivideod eesmärgiga seletada lahti

---

<sup>4</sup> <https://www.datacamp.com/>

<sup>5</sup> <https://www.khanacademy.org/>

Xcode'i kasutamine ja GitHub'is<sup>6</sup> esitletud koodinäidete sisu ning lisaks õpetuse sellest kuidas laadid valminud rakendus oma seadmesse.

## 1.2. Enesekontrollitestid

Enda teadmiste kontrollimisel on oluline osa igas õppevormis nii traditsiooniliste, kui ka digitaalsete õppematerjalide kasutamisel. Kõige lihtsamaks enesekontrolli testi tüübiks programmeerimisalastes õppematerjalides võib pidada valikvastustega küsimusi, mille 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, kuid koodi kirjutamine üldiselt on enesekontrolli peale toetuv tegevus. Nimelt on tagasiside omatud teadmiste kohta koheselt saadav selle pealt, kas rakendus, veebileht või programm töötavad ilma vigadeta või mitte.

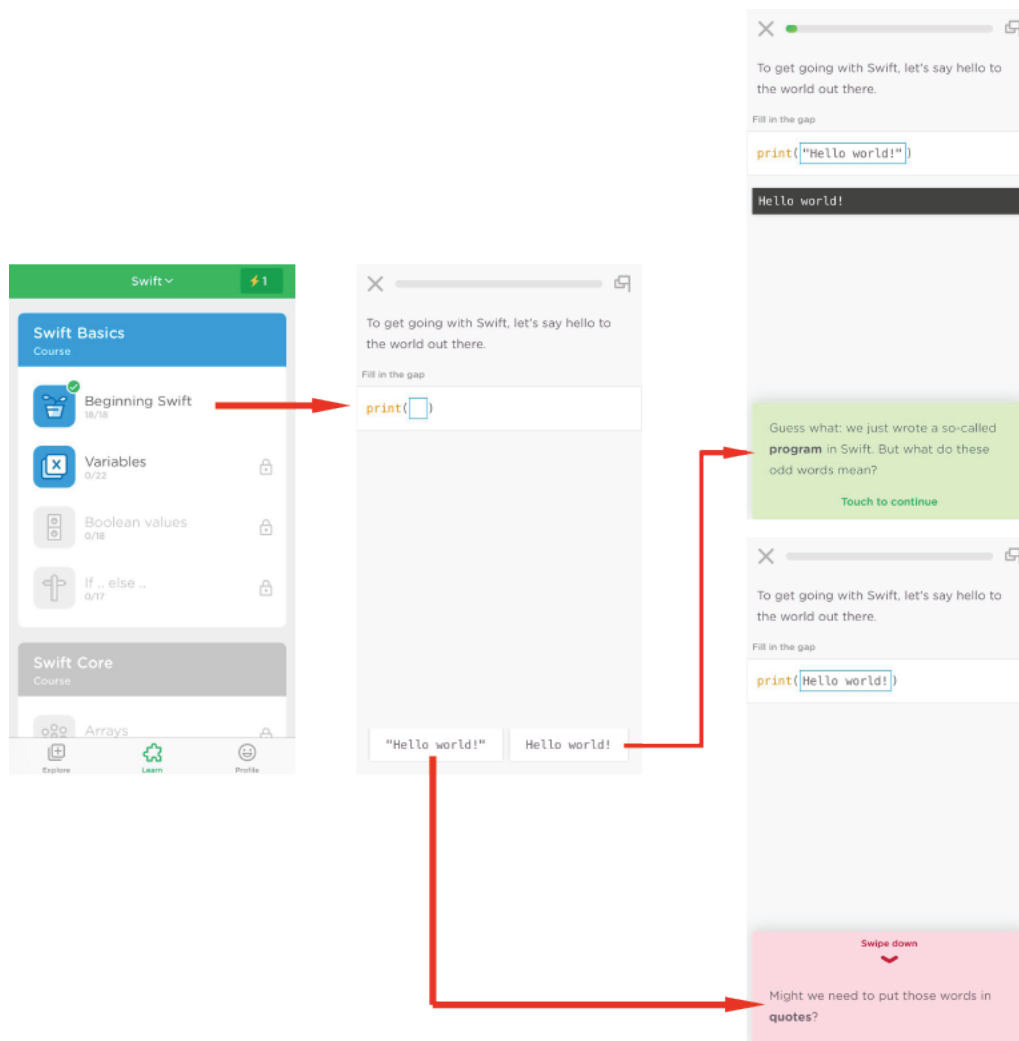
Seda arvesse võttes kasutavad enesekontrolli mingil määral kõik õppematerjalid seminaritöös väljatoodud programmeerimisalaste õppematerjalide analüüsis, kuid autori hinnangul neist parimaid lahendusi kasutavad Mimo<sup>7</sup> (Joonis 2) ning Py<sup>8</sup> äpid. Järeldus tuleneb nende kasutamislihtsusest ning selgest ülesehitusest, kus näiteks uues programmeerimiskeele õppimisel alustatakse nullist ja jagatakse materjal loogiliselt arenevasse järjekorda ning ülesandeid saab teha kiirelt ja mänguliselt, kus iga osa lõpus on teadmiste kontrolliks küsimused ja lünkade täitmised, millele järgneb kohene tagasiside ning vajadusel parandamine ja seletus.

---

<sup>6</sup>[https:// github.com/](https://github.com/)

<sup>7</sup> <https://www.getmimo.com/>

<sup>8</sup> <https://www.getpy.com/>



## Joonis 2. Mimo äpp

Autor leiab, et iOS rakenduse arenduse õppematerjalis saab õpilane enesekontrolli vastavalt koodi õigesti kirjutamisele iOS Simulator'i<sup>9</sup> käivitamisel, nähes, kas kood toimib sinnamaani rakenduses nii nagu peab. Samas on antud õppematerjalis enesekontrolliks kasutusele võetud ka iga praktilise osa lõpus leitav iseseisvalt lahendatav ülesanne tunnis valminud koodi täiendamiseks õpitu põhjal ning teadmiste kontrolli test.

<sup>9</sup> iOS Simulator tuleb kaasa iOS SDK'ga, kui laetakse alla Xcode ning on mõeldud äppide tõhusaks prototüüpimiseks, testimiseks, vigade parandamiseks ning visuaalse disaini optimeerimiseks kogu arendusprotsessi vältel. Simuleerida on võimalik näiteks erinevaid iPhone ning iPad vahendeid.

### 1.3. Simulatsioonid

Simulatsioon on reaalmaailma protsessi või süsteemi jäljendamine ajas, mis võimaldab õppijatel riskivabas keskkonnas oma uut teadmist rakendada ja uusi oskusi omandada.

Seminaritöös väljatoodud programmeerimisalastes õppematerjalides, ei kasuta simulatsiooni otseselt mitte ükski neist, kui just mitte võtta väga häguselt arvesse koodikirjutamist ning koodilünkade täitmist ning sellele järgnevat näidist sellest, mida vastavalt sellele ekraanil koodi järgi kuvatakse.

Käesolevas iOS rakenduse arenduse õppematerjalis võetakse simulatsioon kasutusele selle mõiste kõige sõnaotsesemas mõttes, kasutades selleks Xcode'i allalaadimisega kaasatulevat iOS SDK<sup>10</sup> osa, milleks on iOS Simulator (Joonis 3).



*Joonis 3. iOS Simulator*

---

<sup>10</sup> iOS SDK (Software Development Kit) on Apple'i poolt välja lastud tarkvaraarenduskomplekt, mis tagab arendajatele võimaluse luua, paigaldada, käivitada ning testida rakendusi nii iOS kui ka macOS platvormile.

Pärast õppematerjali läbimist on võimalik valminud rakendus ka enda seadmesse tõmmata ning katsetada selle toimimist ka seal, mida võib võtta, kui füüsilist simulatsiooni.

#### **1.4. Mängupõhise õppe lahendused**

Simulatsioonid on tihedalt seotud mängudega, nimelt mida rohkem mängulisi elemente simulatsioonile lisada, seda rohkem hakkab see endast kujutama midagi õpimängu sarnast. Otseseks õppevahendi tarbeks luuakse tõsimänge ehk õpimänge, mille loomisel kaasatakse nii programmeerijaid kui ka õpetajaid. 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. Õpimängu osadeks on ka punktide ja digitaalsete tunnustusmärkide kogumise võimalus, et haarata kaasa ning motiveerida sellega õpilasi. Tõsimängud annavad nii õpilastele kui õpetajatele võimaluse teha õppimisest lõbus, tulemusteni viiv ning sõltuvust tekitav protsess.

Seminaritöös väljatoodud programmeerimisalastest õppematerjalidest kasutavad mängupõhiseid õppelahendusi kaudselt Khan Academy oma digitaalsete tunnustusmärkidega, DataCamp oma kogemuspunktide süsteemiga ning sama punktide ja tunnustusmärkide süsteem on sees ka Mimo ning Py äppides. Otsest mängupõhist õppelahendust ehk programmeerimisalast õpimängu kasutavad CodeSpells<sup>11</sup> arvutimäng, CodeCombat'i<sup>12</sup> veebipõhine programmeerimismäng ning Colobot'i<sup>13</sup> (Joonis 4) arvutimäng.

---

<sup>11</sup> <https://codespells.org/>

<sup>12</sup> <https://codecombat.com/>

<sup>13</sup> <https://colobot.info/>



*Joonis 4. Colobot koodi näide*

Programmeerimist õpetava õpimängu loomine on aga aeganõudev ning mitme inimese koostööd eeldav projekt, seega ei ole autoril võimalust seda enda töös otseselt rakendada. Käesoleva iOS rakenduse arenduse õppematerjalis võib aga kaudselt mängulise elemendina välja tuua peale simulaatori kasutamise ka õppematerjali lõpptulemusena valmis mängurakendus, mida saab hiljem oma seadmest ka kasutada. Teadmiste kontrolliks loodud testid on samuti vähesel määral mängulise elemendiga.

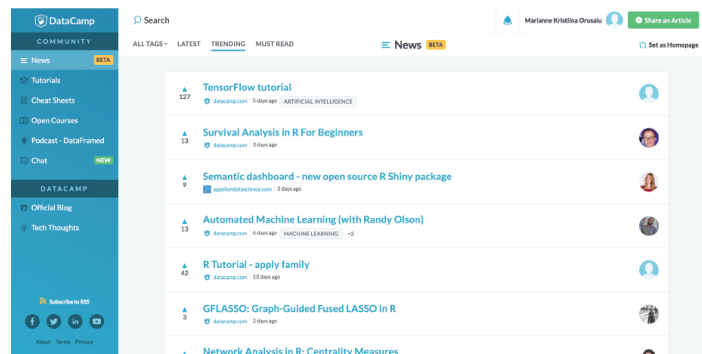
## 1.5. Kogukonnas õppimist toetavad lahendused

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 testide,

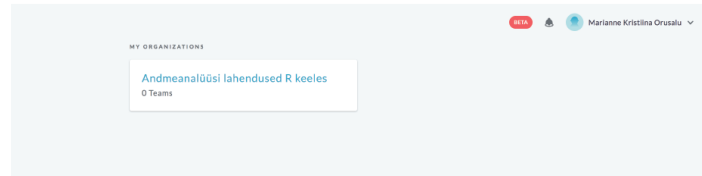
inimese ja arvuti vahelise suhtlemise ja mängude ning teiste multimeediumi võimaluste kaudu võib osutada mõnes kohas pingeliseks ning raskeks, kui on vaja kohest tagasisidet või selgitust millelegi, mida arvuti ei ole lihtsalt programmeeritud tegema. Selleks on aga olemas mõningaid lahendusi programmide ja veebilehete näol. Näiteks luuakse õppimist toetavad veebipõhised õpikeskkonnad ning kasutatakse kogukonnapõhiseid arenduskeskkondi nagu GitHub<sup>14</sup> või spetsiaalselt Python'i koodi õppimiseks ja jagamiseks mõeldud CodeSkulptor<sup>15</sup> (Warren, Rixner, Greiner, & Wong, 2014).

Seminaritöös väljatoodud programmeerimisalastest õppematerjalidest kasutavad kogukonnas õppimist toetavaid lahendusi DataCamp veebileht (Joonis 5), Codecademy<sup>16</sup> veebileht (tasuline juhendaja abi), Khan Academy veebileht ning mingil määral ka CodeCombat mäng, mis pakub meeskonna loomise võimalust.

### Foorumi kasutamise võimalus:



### Õpigruppide loomise võimalus:



### Joonis 5. DataCamp kogukond

<sup>14</sup> <https://github.com/>

<sup>15</sup> <http://www.codeskulptor.org/>

<sup>16</sup> <https://www.codecademy.com/>

Kuna antud iOS rakenduse arenduse õppematerjal on ehitatud üles GitHub keskkonda, on autor sellega taganud kogukonnas õppimist toetava lahenduse. Näiteks saab koodist koopiaid teha ja edasi arendada ning teistel on võimalus koodi parandada ning täiendada.



## 2. Metoodika

Õppematerjali loomisel kasutas autor protsessimudelit nimega ADDIE mudel (Branch, 2009). Tegemist on ühe õppematerjalide väljatöötamiseks mõeldud levinuima mudeliga ning seda rakendatakse nii traditsiooniliste kui ka digitaalsete õppematerjalide kavandamisel, mille tõttu leidis autor ka selle sobivuse antud õppematerjali koostamiseks. 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 loodava projekti 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.

Autor viis oma seminaritöös läbi ADDIE mudeli järgi õppematerjali analüüsi ning osalise kavandamise etapi, andes endale sealjuures võimaluse valminud kavandit muuta, kohendada ja täiendada vastavalt vajadusele ja sobivusele antud bakalaureusetöös toimuvale väljatöötamise etapile.

Analüüsi etapis uuriti kirjandusallikate põhjal, mis võiksid olla õpilaste vajadused ja eelistused veebipõhiste programmeerimisalaste õppematerjalide täitmisel ning otsiti programmeerimisalaste õppematerjalide koostamiseks erinevaid multimeediumi võimalusi, et hiljem antud teadmisi võimaluse korral rakendada õppematerjali väljatöötamises.

Seminaritöös toimunud analüüsi tulemused:

- **Koolitusvajaduse kindlakstegemine:** Vähene või üldse puudulik eestikeelsete programmeerimisalaste õppematerjalide hulk iOS platvormile.
- **Õpieesmärkide määratlemine:** Õppija omandab algteadmised Swift keele kohta ning oskuse koostada koodinäidete eeskujul lihtsamaid iOS rakendusi.
- **Sihtgrupi analüüs:** Sihtgrupiks on infotehnoloogia suunaga tudengid ning programmeerimishuvilised inimesed.
- **Tehnilise platvormi valik:** Töö toimub Mac arvutites Xcode programmis, kasutades Swift keelt.

Seminaritöös sai analüüsi etapi tegevuste all ära mainitud ka see, et õppematerjalidele saab ligi WordPress veebikeskkonnas ning GitHub'is. Seminaritöö kaitsmisel õppejõust retsensendi soovitusel ning autori hilisema endapoolse uurimuse tulemusena tehti selles osas muudatus nii, et antud programmeerimisalane õppematerjal on rajatud üheselt GitHub'i keskkonda, millest räägitakse lähemalt antud bakalaureusetöös õppematerjali kavandamise all.

Seminaritöös toimunud kavandamise tulemused:

- **Õpiväljundite sõnastamine:**
  - 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.
- **Kursuse sisu kavandamine**

Seminaritöös tabelina (Lisa 1) esitletud kursuse sisu kavandile tegi autor õppematerjali luues muudatusi, millest kirjutatakse antud bakalaureusetöös õppematerjali kavandamise ning õppematerjali väljatöötamise all.

Lisaks kavandamise etapi täiendamisele ning väljatöötamise etapile viib autor antud bakalaureusetöös läbi ka hinnangu andmise etapi ekspertidega.

**Bakalaureusetöös läbi viidavad kavandamise etapi täiendamise tegevused:**

- kursuse sisu kohandamine ja täiendamine;
- struktuuri kohandamine ja täiendamine;
- tehnilise platvormi valik.

**Bakalaureusetöös läbi viidavad väljatöötamise etapi tegevused:**

- õppematerjali loomine;
- täiendava metoodika valimine või loomine;
- õppija juhendmaterjalide koostamine;
- muudatuste ja täienduste sisseviimine;
- piloottestimine.

**Bakalaureusetöös läbi viidavad hinnangu andmise etapi tegevused:**

- hinnangukriteeriumide määratlemine;
- hindamisvahendite valik;
- hindamise läbiviimine.

Õppematerjali hindamine viidi läbi LORI hindamismudeli põhjal (ingl Learning Object Review Instrument), mis on mõeldud õppematerjalide hindamiseks ekspertide poolt (Leacock & Nesbit, 2007). Antud hindamismudel sisaldab 9 kriteeriumit õppematerjali kvaliteedi hindamiseks. Hindamisel osales 3 eksperti, kellel on pikaajaline tarkvaraarenduse kogemus.

### **3. Õppematerjali kavandamine**

Lähtudes ADDIE mudeli teisest osast sõnastas autor eelnevalt oma seminaritöö kavandamisfaasis õpiväljundid ning nüüd viib autor antud peatükis läbi põhjalikuma ning täiendatud õppematerjali kavandamise etapi. Kavandamise etapis käsitletakse ning kohandatakse loodava õppematerjali sisu ja struktuuri ning valitakse tehniline platvorm.

#### **3.1. Õppematerjali sisu ja struktuuri kavandamine**

Õppematerjali sisu struktuuri kavandades lähtus autor tarkvara ning Swift keele funktsionaalsuste õppimise järjekorrast ning mängu loogilisest ülesehitamisest. Samuti kandis autor siin edasi sarnases loogilises järjestuses praktiliste tundide struktuuri nagu oli ingliskeelses materjalis, mille järgi autor seda endale selgeks tegi.

##### **3.1.1. Rakendus ja koodinäited**

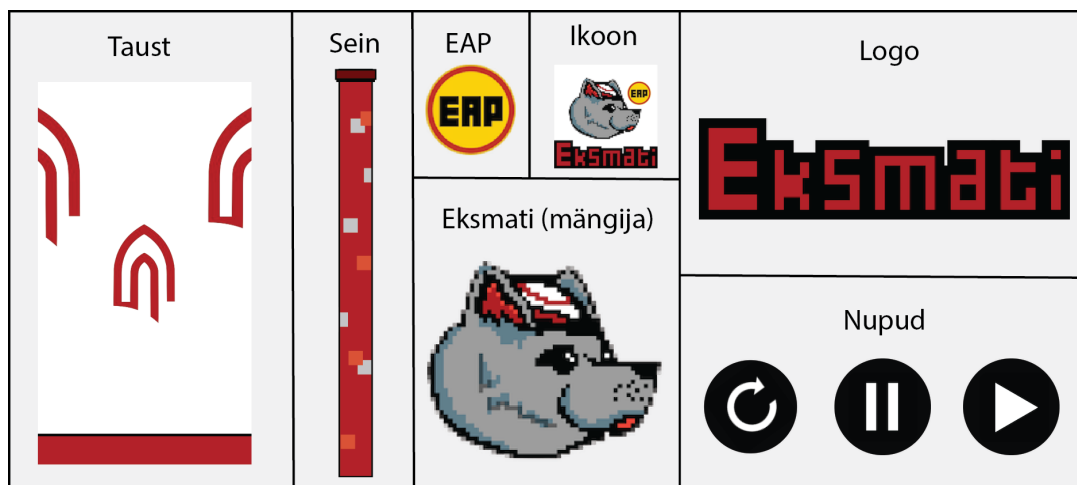
Kuna mobiilirakendusi on mitmeid erinevaid nii praktilisi, kui ka mängulisi otsis autor eestikeelse materjali puuduse tõttu erinevaid ingliskeelseid õpetusi nii videote, kui ka veebilehtede näol, kuni leidis piisavalt köitva ning arusaadava rakenduse loomise õpetuse, mida oleks huvitav nii õppida, kui ka edasi õpetada.

Rakendus mille autor välja valis, kogus oma tuntust 2013. aastal ning kandis nime Flappy Bird<sup>17</sup>. Ingliskeelseid materjale taolise mängu loomiseks on autori otsingute põhjal Swift 2 ning Swift 3 keele kasutusega ja loodud vanemal platvormil, kuid nende pealt algelise koodi põhimõtted on võimalik selgeks saada ning Xcode 9 aitab edasise koodi parandusega kaasa.

---

<sup>17</sup> <http://flappybird.io>

Enne äpi läbi tegemist ja endale Swift keele õpetamist otsustas autor luua uued mänguelemendid, valides teemaks Tallinna Ülikooli (Joonis 6). Selleks kasutati Eksmati tegelase valmistamiseks Pixilart<sup>18</sup> joonistamiseks loodud veebilehte kasutades ning kõigi teiste elementide valmistamiseks kasutas autor Adobe Illustrator'it.



*Joonis 6. Mängu elemendid*

Järgmiseks tegi autor läbi rakenduse loomise uute mänguelementidega ning kohandas ja muutis koodi Swift 4 programmeerimiskeelele ning soovitud mängutulemusel vastavaks, tehes samaaegselt endale selgeks Swift keeles muutujate, tsüklite, hargnevuste ja funktsioonide kirjutamine ning Xcode kasutamine koos simulaatoriga. Rakenduse loomise viimase sammuna, tõmbas autor valmis rakenduse enda iPhone seadmesse, et seda ka seal katsetada.

### 3.1.2. Õppematerjali sisu ja harjutuste kavandamine

Pärast rakenduse loomist ja kohandamist pani autor paika täiendatud ning kohandatud kursuse sisu kavandi eelnevalt seminaritöös valminud tabelile (Lisa 1).

<sup>18</sup> <https://www.pixilart.com/>

Kursuse vaatest on õppematerjal jaotatud loogiliselt kulgevalt viieks tunniks. Esimene tund on mõeldud sissejuhatavaks ülevaateks, kus räägitakse, mida antud õppematerjal endast kujutab, milleks see vajalik on, mis on eesmärgid, kellele see suunatud on ning mis platvormidel õppimine toimub. Tutvustatakse iOS platvormi, Xcode'i ja selle süsteeminõudeid ning paigaldamise/uuendamise juhiseid. Esimese tunni viimase osana tehakse ülevaade Swift keelest ning SpriteKit'ist.

Teise tunni sisuks on loodava rakenduse tutvustus ja arenduskeskkonna kasutamisele võtmine ning kohandamine vajadustest ja mugavusest lähtudes. Samuti tehakse teises tunnis algust rakenduse loomisega, kus loodud projekti sisse paigutatakse esmalt mängu elemendid ja seejärel puhastatakse Xcode'i poolt etteantud kood rakenduse loomiseks sobivaks. Teise tunni viimases osas kirjutatakse projekti sisse rakenduse jooksul kasutatavad muutujad, pannakse paika füüsiliste kehade funktsioonid ning lõpuks seatakse üles taust ning kontrollitakse esimest korda simulaatorist kas kõik kuvatakse nii nagu peab.

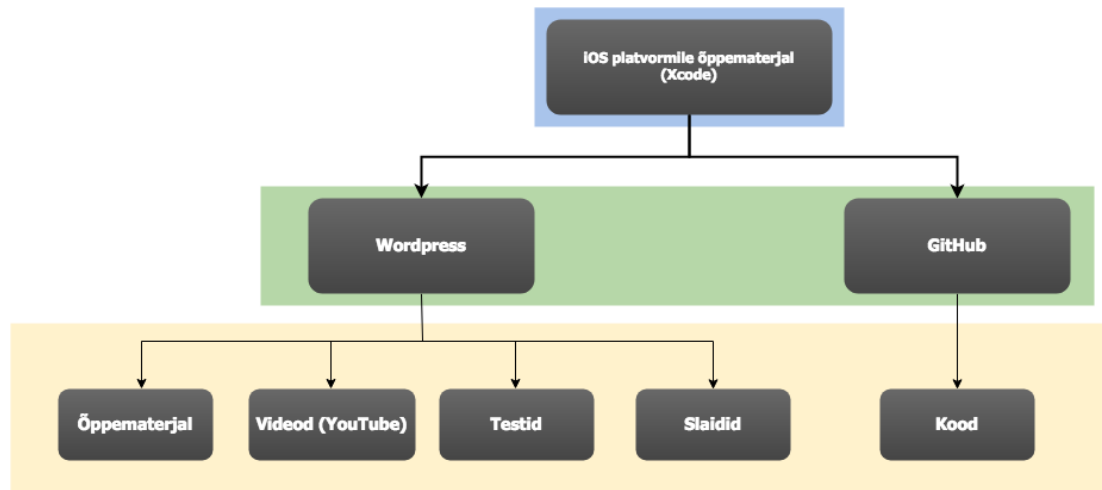
Kolmandas osas lisatakse eelmises tunnis loodud rakenduse muutujatest elementidele nende funktsionaalsused.

Neljandas tunnis lisatakse rakenduse elementidele juurde liikumine ning toimingud, mis juhtuvad pärast kokkupuuteid. Selles osas kuvatakse ekraanile ka viimased kaks mängu elementi – EAP'd ja seinad.

Viimases tunnis viiakse läbi mängurakenduse viimistlemine, kus pannakse paika mängusisesed tegevused olenevat sellest, kus ja millal mängija ekraani vajutab. Samas lisatakse siin ka mängule heli, kui Eksmati puutub kokku EAP'ga või kaotab mängu. Lisaks tehakse selles tunnis läbi ka valmis rakenduse seadmesse tõmbamine.

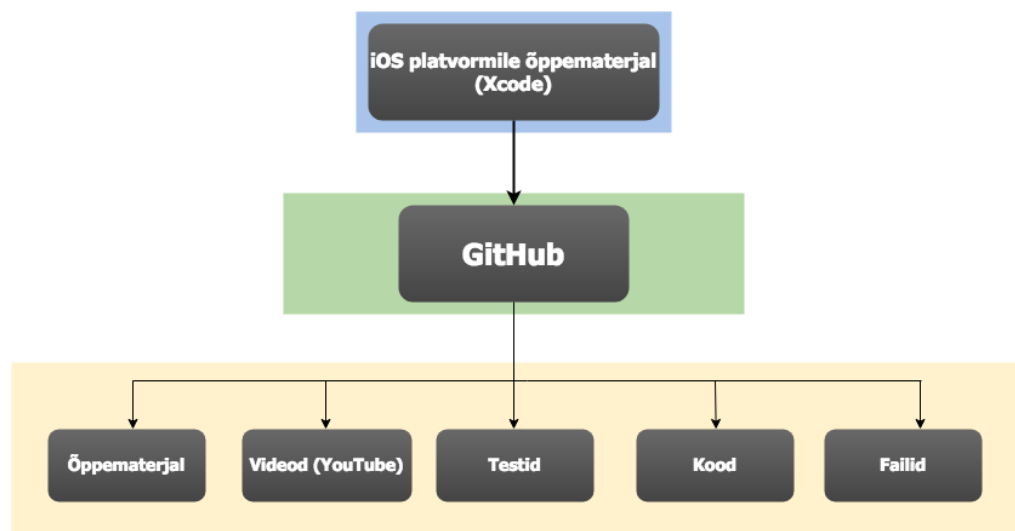
## **3.2. Platvormi valik**

Seminaritöös, õppematerjali kavandamise osas, oli autor algselt võtnud vastu otsuse luua õppematerjal üles kombineeritud platvormi peal kasutades selleks WordPress'i ning GitHub'i (Joonis 7) – WordPress'i oli plaan panna üles õppematerjal, õppevideod, enesekontrollitendid, slaidid ning lingid, mis viivad GitHub'i keskkonda, kus oleks olemas vajalikud koodinäited.



*Joonis 7. Õppematerjali esialgne platvormivalik*

Seminaritöö kaitsmisel suunas õppejõust retsensent tähelepanu aga ainult GitHub'i kasutamise peale ning siinse autori endapoolse uurimuse tulemusena tehti selles osas muudatus nii, et antud programmeerimisalane õppematerjal on rajatud üheselt GitHub'i keskkonda (Joonis 8).



*Joonis 8. Õppematerjali uus platvormivalik*

Algne plaan rajada õppematerjal üles WordPress'i keskkonda blogi stiilis tulenes selle mugavusest ning kasutajasõbralikusest. WordPress tagab mitmeid erinevaid lehe ülesehitamise ja kujundamise võimalusi ning kirjastiilide kasutamise tüüpe.

Samas on WordPress'i veebilehel mugav ülesse panna slaide ning videoid, sest seal ei viida kasutajat järgmisele lehele vaid videoid saab vaadata õppematerjali siseselt. WordPress on ka laialdasemalt tuntud platvorm, mis tähendab, et programmeerimishuviline, kes muidu on sellel ala võhik ja ei tunne veel GitHub'i, oskab WordPress'i lehel mugavalt navigeerida.

GitHub'i kasutamine seevastu on väga oluline oskus programmeerimise valdkonnas ning samas on see hea platvorm siis, kui vaja peaks minema kogukonnas õppimist sest see pakub palju erinevaid suhtlus- ning jagamisviise lehesiseselt. GitHub'is saab näiteks koodi omavahel jagada ja täiendada ning organisatsioonidesiseselt ka üksteist aidata. Lisaks sellele mängis suurt rolli GitHub'i valikus ka selle koodi kirjutamise võimalus, kus Markdown<sup>19</sup> süntaks lubab sellel eristuda muust tekstist, muutes koodi värviliseks ning ilmelt koodile vastavaks. Võimalusi õppematerjali lehtede loomiseks on GitHub'is mitmeid, kuid autor võttes siinkohal eeskujuks Tallinna Ülikoolis Informaatika erialale suunatud kursustest: “Veebisaidi arendus ja haldus” (ainekood IFI6104.DT) ning “Eesrakenduste arendamine” (ainekood IFI6211.DT). Autor läbis antud kursused 2017/18 õa, kus kursuse vältel läbitav materjal oli õppejõud Romil Rõbtšenkovi poolt üles ehitatud GitHub keskkonnas. Kursuse materjali avalikustamiseks kasutatakse GitHub organisatsiooni<sup>20</sup> (ingl *GitHub Organizations*) loomist ning selle peale repositooriumite<sup>21</sup> (ingl *repositories*)

---

<sup>19</sup> Markdown on lihtteksti vormindamise süntaks, mida saab vajadusel samanimelise tööriistaga HTMLiks teisendada. Markdown on populaarne seletusfailides (readme), veebipõhistes foorumites või tekstiredaktorites kiireks vormindatud teksti kirjutamiseks.



<sup>20</sup> Organisatsioonid on jagatud kontod, kus ettevõtted ja avatud lähtekoodiga projektid saavad korraga koostööd teha mitmel projektil. Omanikud ja administraatorid saavad hallata liikme juurdepääsu organisatsiooni andmetele ja projektidele koos keerukate turvalisuse ja haldusega.

<sup>21</sup> Repositooriumid on kataloogid või mälu ruumid GitHub'is, kuhu paigutatakse projektid. GitHub'i kasutajate seas on need tuntud ka kui “repod”. Need võivad olla ühendatud arvutisisesese kaustaga või olla mälu ruumiks GitHub'is. Repositooriumites saab hoiustada koodifaile, tekstifaile, pilte jne.



ülesse ehitamist, kus iga loeng ja praktika on eraldi repositooriumis. Samuti on “Veebisaidi arendus ja haldus” kursuse raames läbi tehtud GitHub Wiki ülesse ehitamine.

**Tabel 1. WordPress ja GitHub võrdlus**

 <b>WORDPRESS</b>		 <b>GitHub</b>	
<b>Plussid</b>	<b>Miinused</b>	<b>Plussid</b>	<b>Miinused</b>
Lehe kujunduse suurem valik	Koodinäiteid halb üles panna	Markdown'i kood visuaalselt hea	Videoid ei saa käivitada otse lehelt
Videod mängivad samal lehel	Vähem võimalusi kogukonnas õppimiseks	Programmeerimisalane keskkond	Võhikule võõras
Laialdasemalt tuntum		Vajadusel kogukonnas õppimist toetav	Markdown süntaksi kirjutamine võib olla tüütu
Lihtsam sisu muuta ja täiendada		Toetab vajadusel mobiilivaadet	
		Failide manustamise ja hoidmise võimalus	

Autor katsetas materjali avalikustamist mõlemal lehel ning eelnevast võrdlustabelist lähtudes otsustas, et WordPress (Lisa 2) ei sobi koodipõhise õppematerjali avalikustamiseks (Tabel 1).

### 3.3. Õppematerjali litsentsivalik

Igat õppematerjali luues on oluline jälgida autoriõiguseid nii sisu kui ka kasutaja seisukohast lähtudes. Lähtudes Autoriõiguse seadusest on teiste autorite tööde kasutamine õppematerjali siseselt lubatud motiveeritud mahus kui selle tegevusega ei taotleta ärilist eesmärki (Autoriõiguse seadus, 2017).

Kõik õppematerjali jaoks koostatud tekst ning rakenduse valmistamiseks kasutatud pildid on autori originaallooming. Õppematerjali aluseks oleva mängu idee põhineb Flappy Bird mängul, kus kasutatakse Tallinna Ülikooli maskott Eksmatit, kui mängu tegelast. Koodinäidete koostamisel on eeskujuna kasutatud ingliskeelselt rakenduste loomise õpetuste veebilehelt nimega SweetTutos<sup>22</sup> leitavat Swift 3 keelt kasutavat Flappy Bird'i taolise mängu loomise õpetusest. Samuti kasutas autor endale koodi õpetamiseks Jared Davidson'i Swift 2 keeles avalikustatud YouTube õpetusvideot Flappy Bird'i taolise mängu loomisest. Mängurakenduses kasutatavad helifailid on võetud avatud litsentsiga veebilehelt Freesound<sup>23</sup> ning videod on autori enda loodud ja kokku töödeldud tömmised ekraanist. Õppematerjalide tekstid on laiema taaskasutamise tagamiseks avaldatud Creative Commons Attribution 4.0 International<sup>24</sup> litsentsi alusel (Joonis 8). Antud litsents võimaldab teistel materjali muuta ja edasi arendada juhul, kui autorile on korrektselt viidatud.

#### Litsents



Käesolev leht ja kõik teised <https://github.com/ios-rakenduse-arendus> materjalid on [Creative Commons Attribution 4.0 International](https://creativecommons.org/licenses/by/4.0/) Litsensiga.

#### *Joonis 8. Litsents õppematerjali veebilehel*

---

<sup>22</sup> <https://www.sweettutos.com>

<sup>23</sup> <https://freesound.org>

<sup>24</sup> <https://creativecommons.org/licenses/by/4.0>

## 4. Õppematerjali väljatöötamine

Antud peatükis kirjeldatakse iOS rakenduse arenduse õppematerjali loomise protsessi eelnevalt valitud platvormil ning toetudes eelnevalt valminud õppematerjali struktuurile.

### 4.1. Õppematerjali loomine

Õppematerjali väljatöötamist alustas autor GitHub platvormile organisatsiooni (ingl *organization*) loomisest. Organisatsiooni siseselt võeti kasutusele GitHub Wiki kuhu autor sisestas tundide ning peatükkide kaupa kogu õppematerjali kirjaliku teksti ja koodi materjali (Tabel 2), mille sisu sai paika pandud eelnevalt õppematerjali sisu ja struktuuri kavandamise all. Õppematerjali praktiliste tundide osade jaoks sai tehtud eraldi ka repositooriumid, et neid saaks vajaduse korral rakendada ka tundides ning lisada sinna vajalike faile. Rakenduse loomisel kasutatavad mänguelementide pildid ning helifailid võib leida repositooriumite kaustadest, kuhu suunavad ka lingid. Iga praktilise osa lõpust võib leida ka rakendust täiendava ülesande.

Õppematerjalide lehekülgede jaotust võib ülevaاتlikult näha nii Lisa 5 alt, kui ka järgnevast tabelist Tabel 2.

*Tabel 2. Õppematerjali leheküljed (Markdown)*

Õppematerjali leheküljed (Markdown)		
nr	Pealkiri	Teksti maht sõnades
1	Õppematerjalist ( <i>Home</i> )	231
<b>1. tund</b>		
2	1. iOS tutvustus	527
3	2. Xcode tutvustus ja paigaldamine	248
4	3. Swift ja SpriteKit tutvustus	518

<b>2. tund</b>		
<b>5</b>	4. Rakenduse tutvustus	46
<b>6</b>	4.1. Rakendus: Projekti algus ja tausta loomine	1322
<b>3. tund</b>		
<b>7</b>	4.2. Rakendus: Elementide lisamine	751
<b>4. tund</b>		
<b>8</b>	4.3. Rakendus: START e. liikumise lisamine	530
<b>9</b>	4.4. Rakendus: Viimistlemine	499
<b>5. tund</b>		
<b>10</b>	4.5. Rakendus: Äpi paigaldus	254

## 4.2. Videote ja testide loomine

Selleks, et õppematerjalist leitavaid koodijuppe õpilastele paremini mõistetavaks teha, valis autor üheks multimeediumivahendiks lühikeste ja informatiivsete õppevideote (Tabel 3) lisamise materjalile, kus igas videos tehakse läbi tunnis toimuv ning seletatakse õppematerjali kirjalikule osale lisaks koodiridade tähendused ning funktsionaalsused.

Videod valmisid paralleelselt materjali GitHub'i ülesse panemisega.

Ekraanipildi salvestamiseks kasutas autor QuickTime Player'it ning hiljem monteeriti video kokku ja loeti tekst peale kasutades selleks iMovie nimelist, Mac arvutiga kaasnevat, videotöötlus tarkvara.

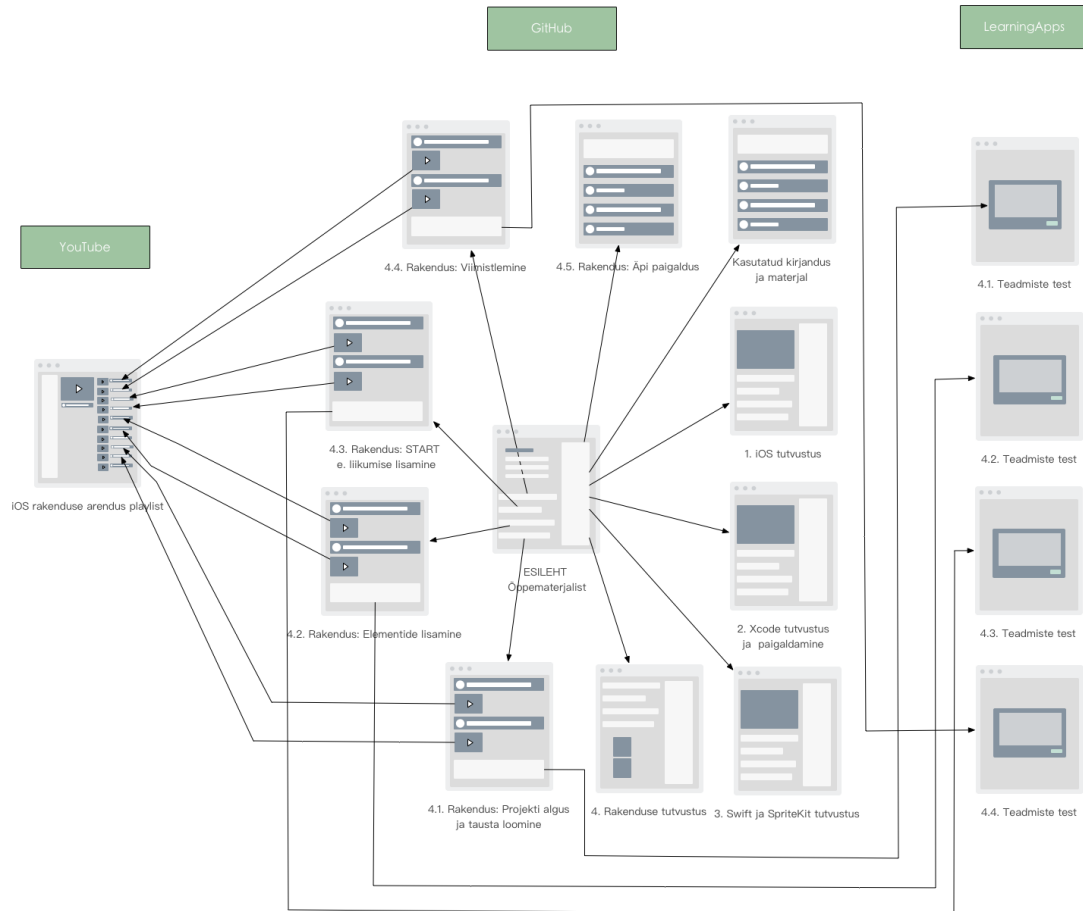
**Tabel 3. Õppevideod**

<b>iOS rakenduse arenduse õppematerjali YouTube'i õppevideod</b>		
<b>nr</b>	<b>Pealkiri</b>	<b>Pikkus</b>
1	Xcode tutvustus ja projekti loomine	2:03
2	Piltide projekti lisamine	1:41
3	Puhastamine // Kategooriad füüsilistele kehadele	2:27
4	Vaate seadistamine	1:43
5	Muutujate lisamine	1:41
6	Tausta kuvamine	3:13
7	Eksmati kuvamine	2:34
8	Teiste elementide kuvamine	2:27
9	Liikumise lisamine	2:13
10	Postid ja EAP'd	2:38
11	Viimistlus	4:03
12	Äpi seadmesse tõmbamine	1:55

Kollektiivselt võib videod leida järgmiselt leheküljelt:

<https://www.youtube.com/playlist?list=PLX0stWuI8VV0FA5lwT35bnLwqpefZmkqP>

Iga praktilise osa lõppu sai õpilastele tagasiside andmiseks LearningApps lehe abiga loodud ka mängulise elemendiga teadmiste kontrollid. Valminud õppematerjali sisukaart on esitatud joonisel 9.



**Joonis 9. Õppematerjali sisukaart**

## 5. Õppematerjalile hinnangu andmine

Kvaliteetne programmeerimisalane õppematerjal peaks olema õppimist toetav, kvaliteetse sisuga, motiveeriv, interaktiivne, kohandatav, tehniliselt korrektne ja ühilduv, kasutajasõbralik ning kättesaadav.

Seda arvestades otsustas autor õppematerjali kvaliteedi hindamiseks kasutada eeskujuna Romil Rõbtšenkovi (2014) bakalaureusetööd ning Liina Vaimla (2014) magistritööd, kus kvaliteedi hindamiseks kasutati LORI mudelit (Leacock & Nesbit, 2007).

Hinnangu andmiseks viis autor LORI mudeli põhjal läbi intervjuu 3 eksperdiga, kellest kaks olid tarkvaraarenduse temaatikaga tegelevad Digitehnoloogiate instituudi õppejõud ning kolmas töötab samas instituudis tarkvaraarendajana. Kõik eksperdid olid Mac kasutajad ning omasid kogemust Xcode keskkonnaga ja iOS arendusega. Ekspertidega läbiviidud uuringu tulemusena tuuakse välja järelused õppematerjali positiivsete ja parandamist vajavate punktide osas.

### 5.1. Ekspertide hinnang õppematerjali kvaliteedile

Õppematerjali kvaliteedile antakse hinnang antud mudeli järgi üheksale komponendile, mis eesti keeles lahtimõtestatuna on järgmised (Põldoja, 2014; Rõbtšenkov, 2014):

1. **sisu kvaliteet** – tõepärasus, täpsus, ideede tasakaalustatud esitus, sobiv detailsuse aste;
2. **õpieesmärkide kooskõla** – õpieesmärkide, tegevuste, hindamismeetodite ja sihtgrupi omavaheline kooskõla;
3. **tagasiside ja kohandumine** – kohandatud sisu ja tagasiside, mis põhineb õppija tegevustel ja mudelil;
4.  **motiveerimine** – võime õppijate sihtgruppi motiveerida;
5. **esitluse kujundus** – visuaalse ja audiaalse info kujundus õppimise ja mäluprotsesside toetamiseks;

6. **interaktsiooni kasutatavus** – navigeerimise lihtsus, kasutajaliidese etteennustatavus ja abivahendite kvaliteet;
7. **ligipääsetavus** – juhtimiselementide ja esitlusformaatide sobivus erivajadustega ja mobiilsetele kasutajatele;
8. **taaskasutatavus** – sobivus kasutamiseks erinevates õpituatsioonides ja erineva taustaga õppijatega;
9. **standarditele vastavus** – vastavus rahvusvahelistele standarditele ja spetsifikatsioonidele.

Hindamises osalenud ekspertidele edastati LORI hindamismudeli küsimustik ja õppematerjali veebilehe aadress ning paluti esmalt iseseisvalt õppematerjal läbi vaadata ning hinnata seda vastavalt küsimustele viiepallisüsteemis (Lisa 3). Hiljem viidi igaühega läbi intervjuu täiendavate kommentaaride saamiseks ning soovitusteks (Lisa 4).

## **5.2. Uuringutulemuste ülevaade**

Intervjuud toimusid 27. aprill 2018. aastal. Intervjuud toimusid iga hindajaga individuaalselt eraldi ning kestsid umbes 30 minutit. Alljärgnevas tabelis on visuaalselt välja toodud õppematerjali kvaliteedi hindamise 9 komponenti ning neile antud hinnangu keskmine viiepallisüsteemis. Hinnangu andsid igale komponendile kõik vastajad (Tabel 4).



**Tabel 4. Hinnangud õppematerjali kvaliteedile**

1. Sisu kvaliteet	★ ★ ★ ★ ☆
2. Õpieesmärkide kooskõla	★ ★ ★ ★ ★
3. Tagasiside ja kohandumine	★ ★ ★ ★ ☆
4. Motiveerimine	★ ★ ★ ★ ★
5. Esitluse kujundus	★ ★ ★ ★ ★
6. Interaktsiooni kasutatavus	★ ★ ★ ★ ★
7. Ligipääsetavus	★ ★ ★ ★ ☆
8. Taaskasutatavus	★ ★ ★ ★ ☆
9. Standarditele vastavus	★ ★ ★ ★ ☆

Järgnevalt on intervjuust kokkuvõtlikult välja toodud ekspertide märkused iga komponendi kohta.

**1. Sisu kvaliteet**

- Xcode seadistamise õpetusvideoid oli hea jälgida ka ilma helita – videote sisu oli kvaliteetne.
- Liigsed tühjad read venitavad koodi ja ei ole ilus lugeda aga samas võib see algajatel aidata koodi paremini eristada.
- Koodisisesed kommentaarid võiksid visuaalselt olla kahel real, siis ei häiriks vaadates see, et peab kerima.
- Idee on hea aga iga praktilise osa lõpus võiks olla ka link kohta, kus saaks kontrollimiseks sinnamaani tehtud koodi tervikuna, et seda oleks hea kasutada ka siis kui seda materjali kasutada tundides ning eelmises keegi puudus siis neil on materjal olemas mille peale edasi ehitada.

- Hästi lahtikirjutatud sisu ja kindel struktuur.

## 2. Õpieesmärkide kooskõla

- Kuna Swift keel on iseenesest suhteliselt keeruline, siis nii on selle õppimist isegi hea alustada tervikliku koodi lahtiseletamisega, mitte otse nullist. Siin peaks ainult mõtlema, et millised eelteadmised õpilastel kindlasti enne olema peavad.
- Enamasti kooskõlas, mingid algteadmised saab. Sõltub eelnevast programmeerimisalastest teadmistest, et mis määral seda mõistetakse. Oleks kindlasti vaja mingeid algteadmisi.
- Mängu tegemine on natuke edasijõudnud teema. Kui tahta täiesti nullist alustada, siis on reeglina lihtsamaid rakendusi mida tutvustada.
- Enne antud õppematerjali läbimist peaks äkki läbima mingite algteadmiste kursuse, mis näitab Xcode põhimõtteid, kuidas simulaator töötab jne. siis see materjal oleks järgmiseks sammuks.

## 3. Tagasiside ja kohandamine

- Õppematerjal on lineaarselt (järjest) ülesse ehitatud.
- Kohandatud sisu on keeruline üle kanda sellistes materjalides ehk sellised ülesanded tundides, mis toetaks eelmist ülesannet.
- LearningApps testid annavad lõpus küll mingit tagasisidet.
- Kui sa kirjutad koodi ja see töötab siis see ongi tagasiside õpilasele ehk simulaator ongi tagasiside.
- Õppematerjal on loogiliselt hästi sammude kaupa üles ehitatud.
- Tagasiside ei tule otseselt õppematerjali pealt vaid simulaatori käivitamisest ja katsetamisest.
- Kui on täiendavat tagasisidet vaja siis peaks selle panema kuskile mujale teises vormis.

## 4. Motiveerimine

- Multimeediaformaad – on olemas tekst ja video mis on motiveerivad.
- Teadmistekontrollid võivad ka siinkohal motivatsiooniks olla. Töö tundub nende pärast isegi tugevam.

- Kuna tegemist on mängurakenduse tegemisega ja mängud meeldivad kõigile siis see on motiveeriv.
- Piisavalt kompaktsed ja arusaadavad sammud, et olla motiveeriv materjal ja äpp lõpuks valmis ka saada.

## 5. Esitluse kujundus

- Rõhutati video olemasolu.
- Kood on tänu Markdown formaadile värviline Swift märgise kaudu.
- Ekraanipildid olid sobivad.
- Oli olemas tekst video ja interaktiivne tekst koos.
- Materjali struktuur hästi ülesse ehitatud ja sektsioonid on loogilised.
- Erilist kujundust pole sest see on GitHub Wiki ning seal on võimalused selleks piiratud.

## 6. Interaktsiooni kasutatavus

- Videote järgmine töötas ilma helita ka hästi.
- GitHubi Markdown on standardne ja arusaadav.
- Lihtne navigeerida ja arusaadav.
- Iga lehe lõpus võiks olla lingid eelmisele ülesandele, järgmisele ülesandele ja esilehele.

## 7. Ligipääsetavus

- Markdown'i eripära ja GitHub Wikit on hea mobiilivaates näidata.
- Videoid ei saa kohe lehel mängida vaid suunab lehelt edasi, sest Markdown ei võimalda teisiti. See võib olla hea selles mõttes, et mobiilivaates (näiteks, Android telefonides) avab lehtedel olevaid videoid eraldi veebiplatvormina, mis häirib.
- GitHub on õige platvorm selle õppematerjali jaoks.
- Ei oska seda väga hinnata sest ei ole otseselt ligipääsetavuse kohta uurinud aga ei meeldi, et viskab telefonis *desktop* liidese ette.

- Äkki see ei peaks üldse olema GitHub Wikis vaid GitHub Pages<sup>25</sup>, kus ta oskab õiget versiooni vastavalt seadme tüübile ette visata.

## 8. Taaskasutatavus

- Taaskasutatavust võib pidada heaks, kuna õppematerjal on tehtud GitHub'is siis seda on võimalik üle kanda.
- Erineva tasemega õpilastele on seda keeruline sobivaks teha. Mingi asi peaks olema siis lihtsam ja mingi keerulisem kuid seda on raske teha.
- Siin saab ka mainida iga tunni ülesannete lõpptulemuste jagamist, et hiljem oleks kus jätkata kui ei osanud ülesannet lahendada või puuduti eelmisest tunnist.
- Kommentaarid eestikeeles on sobivad aga eestikeelseid muutujaid ei tohiks kasutada.
- Kui õppija on natuke edasijõudnud, siis temajaoks oleks see kasulik, eriti kui ei ole kokku puutunud mängude arendamisega.
- Erineva taustaga inimeste jaoks peaks täiesti sobima.
- Sisu on lihtne võtta ja mujale sisestada seega on hästi taaskasutatav.

## 9. Standarditele vastavus

- Standarditele otseselt ei vasta, sest ei ole metaandmestikku kaasa.
- Samas GitHub Wiki ja Markdown on iseenesest ka standardid, kuid mitte otseselt selle punkti kirjelduse järgi.
- GitHub'i kasutamine on ka standard, sest see on rahvusvaheline programmeerimise standard.
- Antud keskkond, ei toeta erinevaid projektide standardeid.
- Samas on kasutatud GitHub Wikit ja Markdown'i mis on standardsed lahendused.

---

<sup>25</sup> GitHub Pages on staatiline veebilehe hostimise teenus, mis on mõeldud isiklike, organisatsiooni või projekti lehtede vastuvõtmiseks otse GitHubi hoidlast.

Autor võtab siinkohal arvesse ekspertide tagasisidet ning teeb parandusi õppematerjalis seal, kus see võimalik on.

Parandused ja täiendused õppematerjalis:

- Iga osa lõppu lisandub "eelmine", "järgmine" ja "sisukord" nupp.
- Õppematerjalile lisatakse juurde ka GitHub Pages veebilehe formaat, mida on võimalik kasutada vajadusel Wiki asemel.
- Koodisestest kommentaaride ridade loetavuse parandamine.

Algelisele GitHub Pages versioonis õppematerjali lehele saab ligi aadressil:

<https://ios-rakenduse-arendus.github.io/>

## Kokkuvõte

Bakalaureusetöö eesmärgiks oli läbi teha äpi arendus iOS platvormile ja selle põhjal luua eelnevalt valminud seminaritöö põhjal lihtsa eestikeelse iOS arenduse õppematerjal ning läbi teha tagasiside uuring ekspertidega, et kontrollida õppematerjali kvaliteeti.

Käesoleva bakalaureusetöö esimeses pooles teeb autor kokkuvõtte eelnevalt Seminaritöö raames tehtud kirjanduse analüüsist, kus vaadeldi erinevaid multimeediumi kasutamise võimalusi programmeerimisalastes õppematerjalides ning teeb järeldused nende kasutamisest iOS rakenduse arenduse õppematerjalis.

Teiseks viis autor antud bakalaureusetöö raames läbi õppematerjali kavandamise lõpliku protsessi, kus valiti muuhulgas ka uus tehniline platvorm ning lisati õppematerjalile litsents. Õppematerjali kavandamise tulemusi kasutati aluseks õppematerjali väljatöötamisel, mille tulemusena valmis GitHub'i keskkonnast kättesaadav iOS rakenduse arenduse õppematerjal, mis sisaldab endast koodijuppe, õppevideoid ning teadmiste kontrolli teste.

Viimases peatükis viidi läbi õppematerjalile hinnangu andmine ekspertidega. Hinnangu andmise eesmärgiks oli õppematerjali standardile vastavuse kontroll ning selle kvaliteedi määramine. Ekspertidega läbiviidud uuringu tulemusena tuuakse välja järeldused õppematerjali positiivsete ja parandamist vajavate punktide osas.

Antud bakalaureusetöö raames koostatud õppematerjal on kättesaadav aadressil <https://github.com/ios-rakenduse-arendus/EsimeneApp/wiki>.

Tulevikus on võimalik loodud iOS rakenduse arenduse õppematerjali täiendada erinevate rakenduste loomise õpetustega, mille seas on nii praktilisi äppe, kui ka teisi Xcode platvormi võimalusi kasutavaid rakendusi.

Lõputöö autorile andis kombineeritud seminaritöö ja bakalaureusetöö juurde uusi teadmisi programmeerimiskeelest Swift ning sellega rakenduste arendamisest. Samas sai autor antud töö käigus kasulike teadmisi ka programmeerimisalase õppematerjali ülesehitamisest digitaalsel platvormil ning ülevaate sellest milliseid.

## 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). Loetud aadressil <http://www.ppig.org/sites/ppig.org/files/2014-PPIG-25th-proceedings.pdf>
- Autoriõiguse seadus. (2017). RT I 1992, 49, 615. Loetud aadressil <https://www.riigiteataja.ee/akt/116062017008>
- Branch, R. M. (2009). *Instructional Design: The ADDIE Approach*. New York, NY: Springer.
- Clark, D. (2014). Testing Programming Skills with Multiple Choice Questions. *Informatics in Education*, 3(2), 161–178. Loetud aadressil [https://www.mii.lt/informatics\\_in\\_education/pdf/infe041.pdf](https://www.mii.lt/informatics_in_education/pdf/infe041.pdf)
- 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). Loetud aadressil [http://eprints.uwe.ac.uk/28599/3/From\\_Hiscore\\_to\\_High\\_Marks\\_Empirical\\_Stu%20%281%29.pdf](http://eprints.uwe.ac.uk/28599/3/From_Hiscore_to_High_Marks_Empirical_Stu%20%281%29.pdf)
- HITSA. (2015). Programm ProgeTiiger 2015–2017. Loetud aadressil [http://media.voog.com/0000/0034/3577/files/Programm%20ProgeTiiger%202015\\_2017.pdf](http://media.voog.com/0000/0034/3577/files/Programm%20ProgeTiiger%202015_2017.pdf)
- Leacock, T. L., & Nesbit, J. C. (2007). A Framework for Evaluating the Quality of Multimedia Learning Resources. *Educational Technology & Society*, 10(2), 44– 59. Loetud aadressil [http://www.ifets.info/journals/10\\_2/5.pdf](http://www.ifets.info/journals/10_2/5.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.

- Orusalu, M.K. (2018). *Programmeerimisalaste õppematerjali õpidisain iOS arenduse õppematerjali näitel* (seminaritöö). Loetud aadressil <http://www.cs.tlu.ee/teemaderegister/>
- Petkov, A. (2018, 16. jaanuar). Here are the best programming languages to learn in 2018. *Freecodecamp*. Loetud aadressil <https://medium.freecodecamp.org/best-programming-languages-to-learn-in-2018-ultimate-guide-bfc93e615b35>
- Põldoja, H. (2014). Õppematerjalide koostamise protsess ja kvaliteet. Loetud aadressil <http://oppematerjalid.wordpress.com/lugemismaterjalid/oppematerjalide-koostamise-protsess- ja-kvaliteet/>
- Rõbtšenkov, R (2014). *Üliõpilaste kaasamine digitaalsete õppematerjalide koostamisse* (bakalaureusetöö) Loetud aadressil <http://www.cs.tlu.ee/teemaderegister/>
- Vaimla, L (2014) *Interaktiivse sisupaketi loomine udutu abil: võimalused ja kitsaskohad* (magistritöö). Loetud aadressil <http://www.cs.tlu.ee/teemaderegister/>
- Villems, A., Aluoja, L., Pilt, L., Naulainen, M.-M., Kusmin, M., Rogalevitš, V., & Tokko, U. (2015). Digitaalse õppematerjali loomise soovitus: 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>



## Summary

Title: Learning Resource about Application Development for iOS Platform

The purpose of this Bachelor's Thesis was to develop a simple mobile application using Swift 4 programming language on Xcode 9 IDE (integrated development environment). The developed application – a simple mobile game inspired by the Flappy Bird app – was used as a basis for developing a learning resource about application development for iOS platform. The thesis is based on the seminar thesis, in which the author did a literature review and completed the analysis and design phase for developing the learning resource.

The aims of the current research were:

- to develop a simple iOS game to be the foundation for the learning resource;
- to create a learning resource about application development for iOS platform based on previously made game;
- to evaluate the developed learning resource with experts.

This Bachelor's Thesis consists of five chapters. The first chapter of the thesis provides an overview and conclusions of previously done literature review that was done in seminar thesis. In the second chapter, the author introduces ADDIE model, summarizes the results of the first steps that were done in seminar thesis and presents the steps that are going to be done in this Bachelor's Thesis. In the third chapter, the author discusses the design phase of the learning resource. Development phase is explained in the fourth chapter. In the fifth and final chapter the author evaluates the learning resources with three experts using LORI evaluation instrument.

# LISAD

## Lisa 1. Kursuse sisu kavandi tabel seminaritööst

<b>iOS ÕPPEMATERJAL ÄPI LOOMISEKS</b>	
<b>Tund 1. Ülevaade</b>	
<ol style="list-style-type: none"> <li>1. Sissejuhatus</li> <li>2. Arenduskeskkonna ja süsteemi algne tutvustus ja paigaldamine</li> <li>3. Programmeerimiskeele tutvustus</li> </ol>	<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"> <li>• tekst WordPress lehekülgedena;</li> <li>• Xcode keskkonda tutvustav ekraanivideo;</li> <li>• enesekontrolli testid Swift keele süntaksi kohta.</li> </ul>
<b>Tund 2. Algsed teadmised</b>	
<ol style="list-style-type: none"> <li>1. Arenduskeskkonna tutvustus</li> <li>2. Arenduskeskkonna kohandamine</li> <li>3. Projekti loomine</li> <li>4. Simulaatori tutvustamine</li> </ol>	<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"> <li>• tekst WordPress lehekülgedena;</li> <li>• Xcode keskkonda tutvustavad ekraanivideod (Xcode keskkonna kohandamine, projekti loomine, simulaatori kasutamine).</li> </ul>
<b>Tund 3. Sisu loomine</b>	
<ol style="list-style-type: none"> <li>1. Projekti arendamine</li> <li>2. Funktsionaalsuse tagamine</li> </ol>	<p>Kasutajaliidese loomine, millele lisatakse vajaminevad elemendid ning funktsionaalsused.</p> <p>Õppematerjali sisutüübid:</p> <ul style="list-style-type: none"> <li>• tekst WordPress lehekülgedena;</li> <li>• koodinäited GitHubis;</li> <li>• allalaetav Xcode projekt.</li> </ul>
<b>Tund 4. Testimine</b>	<p>Ülesannete andmine ja teadmiste testimine.</p> <p>Õppematerjali sisutüübid:</p> <ul style="list-style-type: none"> <li>• enesekontrolli testid.</li> </ul>

## Lisa 2. WordPress platvormil õppematerjali ekraanipilt



## Lisa 3. LORI hindamismudel ekspertidele

### Lugupeetud eksperdid!

Palun täitke alljärgnev tagasisideankeet iOS rakenduse arenduse õppematerjali kohta. Küsitluse eesmärgiks on välja selgitada Teie arvamused GitHub keskkonnas ülesse ehitatud iOS rakenduse arenduse õppematerjalist, mis on suunatud infotehnoloogia suunaga tudengitele, kuid ka kooliõpilastele ning teistele programmeerimishuvilistele inimestele. Antud küsitluse tulemusi kasutatakse ainult TLÜ informaatika bakalaureusetöös.

Loodud õppematerjali saate vaadata järgmiselt lingilt: <https://github.com/ios-rakenduse-arendus/EsimeneApp/Wiki>

Ette tänades,

Marianne Kristiina Orusalu

Digitehnoloogiaste instituudi informaatika (digitaalne meedia) bakalaureusetudeng.



- |  |   |   |   |   |   |               |
|--|---|---|---|---|---|---------------|
| 1. <b>Sisu kvaliteet</b> — tõepärasus, täpsus, idee tasakaalustatud ehitus, sobiv detailsuse aste                        | 1 | 2 | 3 | 4 | 5 | Vastus puudub |
| 2. <b>Õppieesmärkide kooskõla</b> — õppieesmärkide, tegevuste, hindamismeetodite ja sihtgrupi omavaheline kooskõla       | 1 | 2 | 3 | 4 | 5 | Vastus puudub |
| 3. <b>Tagasiside ja kohandamine</b> — kohandatud sisu ja tagasiside, mis põhineb õppija tegevustel ja mudelil            | 1 | 2 | 3 | 4 | 5 | Vastus puudub |
| 4. <b>Motiveerimine</b> — võime õppijate sihtgruppi motiveerida  | 1 | 2 | 3 | 4 | 5 | Vastus puudub |
| 5. <b>Esitluse kujundus</b> — visuaalse ja audiaalse info kujundus õppimise ja mälu protsesside toetamine                | 1 | 2 | 3 | 4 | 5 | Vastus puudub |
| 6. <b>Interaktsiooni kasutatavus</b> — navigeerimise lihtsus, kasutajaliidese etteennustatavus ja abivahendite kvaliteet | 1 | 2 | 3 | 4 | 5 | Vastus puudub |
| 7. <b>Ligipääsetavus</b> — juhtimiselementide ja esitlusformaate sobivus erivajadustega ja mobiilsetele kasutajatele     | 1 | 2 | 3 | 4 | 5 | Vastus puudub |
| 8. <b>Taaskasutatavus</b> — sobivus kasutamiseks erinevates õpituatsioonides ja erineva taustaga õppijatega              | 1 | 2 | 3 | 4 | 5 | Vastus puudub |
| 9. <b>Standarditele vastavus</b> — vastavus rahvusvahelistele standarditele ja spetsifikatsioonidele                     | 1 | 2 | 3 | 4 | 5 | Vastus puudub |

## Lisa 4 Intervjuu ekspertidega

Siin on toodud 27. aprillil 2018 aastal Tallinna Ülikoolis ekspertidega toimunud intervjuude üldine sisu. Õppematerjali hindamisse oli kokku kaasatud 3 spetsialisti kellest kõik võtsid osa hindamisest ning hiljem individuaalsest intervjuust. Vestluse üleskirjutuses kasutatakse lühendeid — 1. ekspert **E1**, 2. ekspert **E2** ja 3. ekspert **E3**.

### 1. Sisu kvaliteet – tõepärasus, täpsus, ideede tasakaalustatud esitus, sobiv detailsuse aste

**E1:** Sisu kvaliteedi osas tundus kõik nagu väga hea. Ma isegi tooks välja, et näiteks kasvõi seadistamist ainult teksti põhjal oleks palju keerulisem teha, mul oli kõrval video lahti, video pealt oli seadistamine parem, video andis sellele väga palju juurde. Kuna ma tegin seda samal ajal, kui ma olin koosolekul, siis video oli ilma helita ka piisav mu jaoks. Heli vajalikkus oleneb kindlasti ka sellest, kus sa õpid.

Mul on hea meel, et ei kasutatud eestikeelseid muutujaid. Mõningad nagu liigsed tühjad read raskendavad lugemist, aga samas võibolla see aitabki liigitada erinevaid koodijuppe. Ütleme niimodi, et koodifailid on juba niigi nii pikad, et nad ühele ekraanile äkki ära ei mahu. Samas äkki kui on väga suur ekraan, siis on mugavam.

Mõnes mõttes liigsed pikad read venitavad koodi pikemaks, aga samas aitavad äkki tõesti alguses eristada, et siin ma olen kahevahel.

Koodisisesed kommentaarid võiksid visuaalselt olla kahel real, siis ei häiriks Markdown'i vaadates kerimine.

**E2:** Kui sa jagad seda, tunni lõpus, peale ülesannet võiks olla mingid failid mingis vormis, kus on ülesanne tehtud. Üldiselt see struktuur on selline päris hea, sellise struktuuriga tunnid, et valmis tehtud asi, mingi kood, kus puuduvad osad, õppejõud räägib loengul mis funktsioonid mida teevad ja tunni lõpus sa ise teed seda ära, lõpetad koodi. Algajatele oleks kasulik, et nad kirjutavad koodi ja neil tekib küsimus ja neil on vaja kohta kuhu nad vaatavad, saan aru sul valmis äpis on need kohad olemas. Selles tunnis, kui sul

on valmis äpp ja sul on esimene tund ja sa avad selle, sa ei saa väga aru mida sa hetkel teed. Oleks kasulik ülesande failid või funktsioonid.

Idee iseenesest on päris äge.

**E3:** Ma arvan, et see sisu oli piisavalt hästi lahti kirjutatud. Mulle meeldis, et seal oli kindel struktuur, kõik oli ilusti leitav. Seal olid koodijupid, joonised/pildid mis näitavad, kuidas arenduskeskkonnas peaks asi välja nägema. See kõik toetas teineteist, selles mõttes oli hea.

## 2. **Õpieesmärkide kooskõla – õpieesmärkide, tegevuste, hindamismeetodite ja sihtgrupi omavaheline kooskõla**

**E1:** Ma arvan, et see üldiselt nagu on okei , et arvestades, et ma ise olen ka programmeerimist koolis õpetanud, siis ma usun, et see Swifti keel ise on nagu suhteliselt keeruline ja kui kuskilt otsast alustada, siis on nagu okei, kui sa ei hakka nagu puhtalt klasside ja objekti tasemelt rääkima, et mis see siis on jne. Et sellise terviklikuna koodi , et mulle näiteks meeldib küll tervikuna koodi rohkem õppida. Mina hindaksin, et tõesti sellist terviklahendust. Ütleme nii, et kui sa hakkasid programmeerimist nullist õpetama iOS'i peal, siis sama mahuga, mis sul siin on kuni äpi valmimiseni, siis tegelikult õpilastel saaks selgeks, et mis asi on objekt ja mis asi on massiiv, et mulle tundub, et hetkel see materjal on nagu selliste asjade jaoks nagu täitsa sobiv, aga siin peaks võibolla nüüd mõtlema, mis need eelteadmised võiksid olla.

**E2:** Mulle tundub, et enamasti kooskõlas, selles mõttes, et algteadmised saad. Sõltub, kas on sul mingi programmeerimis taust, algteadmised, tüüp-teadmised. Peavad olema mingid programmeerimis oskused või algteadmised selleks, et alustada.

**E3:** Kuna sa tood siin välja, et see materjal peaks andma algteadmisi Xcode programmeerimises, seda on võimalik nii ja naa hinnata. Kuna mängu tegemine on natuke edasijõudnud teema ikkagi. Kui sa tahad täiesti nullist alustada on seal reeglina lihtsamaid rakendusi, mida tutvustatakse. Ma oletaksin kui inimene hakkaks täiesti nullist ta suudaks küll midagi kokku panna, aga kas ta ikkagi suudab asjast lõplikult aru saada, see on omaette küsimus. Ma ütleksin võibolla enne seda peaks veel läbima sellise basic taseme tutoriali, mis näitab



Xcode põhimõtteid, kuidas simulaator töötab ja kõik need ja siis peale seda saaks juba sinu materjalide järgi selle rakenduse kokku panna.

### **3. Tagasiside ja kohandumine – kohandatud sisu ja tagasiside, mis põhineb õppija tegevustel ja mudelil**

**E1:** Mina arvan, et siin hetkel on kogu õppimisprotsess suhteliselt lineaarne, et järjest on ülesanded ülesse ehitatud.

Kohandatud sisu on üleüldse väga keeruline tagada sellistes veebimaterjalides, et näiteks, et ala stiilis, et kui ma esimese ülesande teen niimoodi ja valin sellise ülesande või teen seda siis järgmine ülesanne on seda toetav. Et tegelikult nad on kõik ikka suhteliselt ette antud, et mida ja kuidas sa teha saad.

LearningApps annab ka lõpus mingit tagasisidet.

**E2:** See tagasiside, see on selles mõttes, et kui sa kirjutad koodi ja see töötab see ongi tagasiside. Selles mõttes simulaator ongi tagasiside.

**E3:** Ses mõttes, et see materjal on üles ehitatud nii, et selle abil on võimalik õppijat hästi suunata, et see on loogiliselt hästi sammude kaupa üles ehitatud. Samas tagasiside osas minu meelest seal väga nagu tagasisidet polnudki. Tagasiside ei tule materjali poole pealt vaid see tuleb otse kui paned käima ja katsetad. Mõtlen, et kui on täiendavat tagasisidet vaja siis peaks selle panema kuskile mujale teises vormis, või teist rakendust kasutama. GitHub'i Wiki poleks kõige parem kanal või vorm selle info edastamiseks. Võiks olla mõni teine keskkond selle kasutamiseks.

### **4. Motiveerimine – võime õppijate sihtgruppi motiveerida**

**E1:** Multimeediaformaadi tooks välja siin punktis, et sul on olemas tekst ja video ja tegelt ka mulle on päris mitu tudengit öelnud et neile meeldibki see video formaat, et tegelt kui me vaatame programmeerimisasju siis sealt on videod ka väga populaarsed või tegelikult päris hästi vaadatavad.

Tooks välja ka teadmistekontrolli motiveerimise all. Motiveerimine on nagu selles mõttes hea, et töö tundub teadmiste kontrolli pärast nagu palju tugevam.

**E2:** Sul on mäng ja kõikidele meeldivad mängud. Motiveerib. *To-do* listid pole nii huvitavad kui mäng, kui sa alustad mängu teemaga.

**E3:** Ma arvan, et see on hästi motiveeritud kuna need sammud olid piisavalt kompaktsed, arusaadavad ja kui sa läbid ühe sammu ja paned käima siis kohe näed tulemust. Hästi motiveeriv ja annab sulle motivatsiooni, et edasi järgmise sammu juurde liikuda ja lõpuks see mäng saaks valmis.

## **5. Esitluse kujundus – visuaalse ja audiaalse info kujundus õppimise ja mäluprotsesside toetamiseks**

**E1:** Siin ma rõhutaks jälle seda sama video olemasolu ning seda, et kood on Markdown'is värviline eraldi Swift märgise kaudu ja tegelikult need samad ekraanipildid olid hästi sobivad.

**E2:** Panid nagu igasugused asju nagu tekst, video ja interaktiivne test.

**E3:** Materjali struktuur hästi üles ehitatud sammude kaupa ja seksioonid on loogilised. Erilist kujundust pole, kuna see on GitHub'i Wiki ja seal on sellised võimalused piiratud, aga no ikkagi saab koodilõike eraldi vormistada ja välja tuua. Piiratud võimalusi on siin ikkagi võimalik kasutada ja praegu võibolla isegi piisavalt.

## **6. Interaktsiooni kasutatavus – navigeerimise lihtsus, kasutajaliidese etteennustatavus ja abivahendite kvaliteet**

**E1:** Videote jälgimine töötas isegi ilma helita väga hästi.

**E2:** See interaktsiooni kasutatavus, sa ju räägid seal sellest, mis mida teeb. Nagu see tavaline GitHub'i Markdown on standardne ja arusaadav.

**E3:** Jällegi minu meelest oli seda hästi lihtne navigeerida ja kõik oli arusaadav. Ainuke asi, mis mul tuli meelde oli see, et kui sa selle peatüki läbid, siis alati pead tagasi kerima. Siin on see kõrvalmenüü, kus on need peatükid ja tavaliselt, kui sa läbid ühe peatüki, siis vähemalt mina olen harjunud sellega, et osa lõpus on viide järgmisele peatükile. Muidu sa pead alati alguses kerima ja otsima selle üles. Oleks palju mugavam ja kiirem. Muidu pead meelde tuletama, kas ma olen esimeses või teises peatükis ja mis järgmine peatükk peaks olema, võtab hetk aega. Aga muidu oleks link, et eelmine peatükk ja järgmine peatükk, oleks väga mugav. Või siis isegi on ka sellised variante, kus pannakse kolm linki: eelmine, järgmine ja sisukord.

## **7. Ligipääsetavus – juhtimiselementide ja esitlusformaatide sobivus erivajadustega ja mobiilsetele kasutajatele**

**E1:** Markdown'i eripära ja juba see, et see seal Wikina on juba päris hea, et seda tõesti saab nii mobiiliseadmes näidata jne. Kuna su videod Markdown'i piirangute tõttu siin ei ole siis mõnes mõttes on see hea ja mõnes mõttes halb, et sul on siin nagu ekraanipildina viide, et see on mobiiltelefonile tegelikult hea, sest kui sa avad video siis ta avabki selle YouTube playeri ja see töötab ka hästi, sest muidu on see, et kui sul on lehel video siis ta avab selle mobiilivaates alati veebiplatvormina, et vähemalt androidile on see hästi halb, et see on selle koha pealt hästi positiivne isegi. Ehk siis Markdown sobib mobiilile hästi ning on üleüldiselt ükskõik kus hästi vaadatav.

**E2:** Sa ju kasutad GitHubi ja Youtube'i mobiilis. Sa ju valisid Githubi, ma arvan see on õige platvorm selleks.

**E3:** Ausalt öeldes ma ei ole seda ligipääsetavust väga uurinud ja ma ei oska seda väga hinnata, aga praegu me katsetasime telefonis ja millegi pärast ta viskab selle desktop liidese ette. Et jällegi ma ei oska öelda millest see tingitud on, äkki see ei peaks olema üldse Wikis vaid GitHub Pages abil tehtud ja siis ta oskaks automaatselt juba õiget versiooni vastavalt seadme tüübile ette visata. Äkki see on lihtsalt Wiki omapära.

## **8. Taaskasutatavus – sobivus kasutamiseks erinevates õpituatsioonides ja erineva taustaga õppijatega**

**E1:** Minu arust see taaskasutatavus on selles mõttes hea, et kuna see on tehtud GitHub'is siis seda on võimalik kasvõi võtta ja vaadata mingi ühte tükki. Kui ma teeksin näiteks oma materjali ja ma teeksin enda materjali, siis seda ülesseadmise osa saab enamvähem tegelikult üle kanda sealt.

Kuigi see õppematerjal on nagu kohandatav siis erineva tasemega õppijatele on seda nagu keerulisem maha müüa, et seal peaks olema ikka mingi asi lihtsam ja mingi keerulisem aga sellepärast ei saa öelda, et materjal on halb aga seda on lihtsalt raske üle kanda. Ma isegi ei kujuta ette, et kui teha mingit kergemat varianti, et siis kas tõesti peaks olema mingi raskem ülesanne, et

tehke nüüd see ja siis kui sa tahad sellest üle mida siis siin on see valmis lahendus, aga see ei ole ka otseselt hea lahendus tegelikult.

**E2:** Ja see on võimalik, sest et sa nagunii koodis kirjeldad kus mis asub. Isegi kui pole oskusi üldse saad copy-paste teaha. Kommentaarid eesti keeles on okei aga muutujad ei tohi teistes keeltes kirjutada, halb praktika. Siia saab selle ettepaneku panna, et võib nagu tunni lõpus mingid faili või kausta tunnis tehtud ülesande kohta anda. Tehtud ülesanded pakkuda kasutajatele, et nad saaksid vaadata kas nad tegid samamoodi. Kui sa teed mitte kodus seda esimene tund esimene päev, teine tund teine päev, siis kui sul on mingi fail esimesest korrast esimene tehtud asi, järgmisel päeval sa alustad kui midagi läks nihu, sinu tehtud failist/kaustast. Ja kui ei õnnestu siis järgmine päev võtab jälle sinu faili/kausta lahti.

**E3:** Ma arvan, et siin on nii ja naa, et kui õppija on natuke edasi jõudnud, siis ikkagi tema jaoks oleks see kasulik, eriti kui ta pole kokku puutunud mängude arendamisega. näiteks mina olen iOS'i rakendusi teinud ja teistele õpetanud, aga ma ei ole otseselt selle mängumootoriga kokku puutunud, see oli minu jaoks selles mõttes huvitav. Selline nagu uus avastus. Erinevate tausta inimeste jaoks peaks täiesti sobima.

## **9. Standarditele vastavus – vastavus rahvusvahelistele standarditele ja spetsifikatsioonidele**

**E1:** Standarditele otseselt ei vasta, sest ei ole metaandmestikku kaasas, mis võimaldaks importe kuhugi või teha sellega midagi aga samas Wiki ja Markdown on iseenesest küll standard, et kui tahta seda üle tõsta või lugeda sest isegi Wikipedia kasutab ju ka Markdown'i põhimõtet, et selles osas on nagu standard tagatud. Et kuna ta õpiobjekti standarditele otseselt ei vasta ja siin on nagu suurem rõhk sellel, siis ma siin ei saa ka viite tähte anda.

**E2:** Sa võid siin öelda, et sa kasutad GitHub'i, see ongi, kindlasti see ongi standard vastavus, sul on programmeerimise koolitus, kasutab GitHub'i, sest see on rahvusvaheline standard kasutada koodi jaoks GitHub'i ja Wiki ongi standard.

**E3:** Siin muidugi see keskkond ei toeta minu teada erinevaid projektide standardeid, aga samas kuna siin on kasutatud Wikit ja Markdown'i mis on

sellised standardsed lahendused, siis ma eeldan, et seda sisu on väga lihtne siit keskkonnast alla laadida ja kuskile mujale sisestada, selles mõttes peaks olema see hästi taaskasutatav.

## Lisa 5. Ekraanipilte GitHub platvormis olevast õppematerjalist



This repository Search Pull requests Issues Marketplace Explore

ios-rakenduse-arendus / EsimeneApp Unwatch 1 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Insights Settings

### Home

Marianne edited this page 2 hours ago · 31 revisions

## Õppematerjalist

Antud iOS rakenduse arenduse õppematerjal on loodud Tallinna Ülikooli informaatika eriala seminaritöö ja bakalaureusetöö raames. Õppematerjal on ülesse ehitatud nii, et kui koheselt kirjalikus pildis koodi ei mõista siis on võimalik iga rea funktsionaalsust kuulata ja vaadata õppematerjali sisse lingitud videotest. Igas praktilises osas on vaja täita iseseisvalt mingi ülesanne ilma sealjuures oleva videoõpetuseta, et lisada sellega rakendusele funktsionaalsusi ning kontrollida, kas eelnevalt kirjutatud koodi on mõistetud.

Koostas: Marianne Kristiina Orusalu, [mkoinc@tlu.ee](mailto:mkoinc@tlu.ee)

Terviklik äpp on kättesaadav [EKSMATproject](#) kaustast. Seda võib kasutada hilisemaks kontrolliks.

### Sisukord

- 1. tund:
  - 1. iOS tutvustus
  - 2. Xcode tutvustus ja paigaldamine
  - 3. Swift ja SpriteKit tutvustus
- 2. tund:
  - 4. Rakenduse tutvustus
  - 4.1. Rakendus: Projekti algus ja tausta loomine
- 3. tund:
  - 4.2. Rakendus: Elementide lisamine

Pages 11

- Home
- 1. iOS tutvustus
- 2. Xcode tutvustus ja paigaldamine
- 3. Swift ja SpriteKit tutvustus
- 4. Rakenduse tutvustus
  - 4.1. Rakendus: Projekti algus ja tausta loomine
  - 4.2. Rakendus: Elementide lisamine
  - 4.3. Rakendus: START e. liikumise lisamine
  - 4.4. Rakendus: Viimistlemine
  - 4.5. Rakendus: Äpi paigaldus
- Kasutatud kirjandus ja materjal

#### 4. tund:

4.3. Rakendus: START e. liikumise lisamine

#### 5. tund:

4.4. Rakendus: Viimistlemine

4.5. Rakendus: Äpi paigaldus

---

**Õpieesmärgid:** Õppija omandab algteadmised Xcode integreeritud arenduskeskkonna käsitlemise ja Swift keele kohta ning algse oskuse mõista ning kasutada neid praktikas, mille lõpptulemusena valmib mängu äpp.

#### Õpiväljundid:

- oskab kasutada Xcode arenduskeskkonda uue projekti loomiseks, koodi sisestamiseks(vajadusel parandamiseks/kohandamiseks) ja koodi käivitamiseks simulaatoris;
- tunneb Swift keele süntaksit;
- on võimeline koostama koodinäidete põhjal lihtsamaid iOS rakendusi Swift keeles.

**Sihtgrupp:** Õppematerjali sihtgrupiks on infotehnoloogia suunaga tudengid, kuid ka kooliõpilased ning teised programmeerimishuvilised inimesed.

**Tehnilise platvorm:** Töö toimub Mac arvutites Xcode programmis, kasutades Swift keelt ja SpriteKit'i.



## Innukat õppimist!

#### Litsents



Käesolev leht ja kõik teised <https://github.com/ios-rakenduse-arendus> materjalid on Creative Commons Attribution 4.0 International Litsensiga.

---

+ Add a custom sidebar

Clone this wiki locally

[https://github](https://github.com)

Clone in Desktop

## 4.1. Rakendus: Projekti algus ja tausta loomine

Edit New Page

Marianne edited this page 22 hours ago · 4 revisions

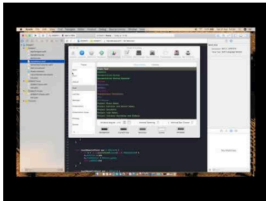
### 1. Ava Xcode ning tee uus projekt iOS mängu (Game) jaoks.

2. Anna mängule nimi ning kontrolli, et kasutad Swift keelt ning SpriteKit'i ja salvesta see endale meelepärasesse ja meeldejäädvasse kausta oma arvutis.

### 3. Kohandamine:

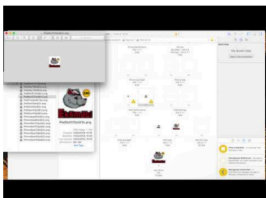
- Eemalda seadme keeramise funktsiooni linnukesed antud äpis
- Kontrolli, Xcode menüüst *Preference*'i alt, et kõik oleks ajakohane ja lae alla uuendused, kui on vaja
- Kui soovid, kohanda Xcode endale meelepärasek

VIDEO(1.,2.,3.):



4. Lae [siit](#) alla mängu elementide(Assets) kaust.
  - Paiguta sealt leitavad ikoonid(AppIcon kaustas) Xcode projekti sees olevasse Assets.xcassets kausta selleks ettenähtud kohtadesse vastavalt nimele.
  - Tõsta kõik ülejäänud pildid Assets kaustast Assets.xcassets kasuta

VIDEO(4.):



Pages 11

Home

- 1. iOS tutvustus
- 2. Xcode tutvustus ja paigaldamine
- 3. Swift ja SpriteKit tutvustus
- 4. Rakenduse tutvustus
  - 4.1. Rakendus: Projekti algus ja tausta loomine
  - 4.2. Rakendus: Elementide lisamine
  - 4.3. Rakendus: START e. liikumise lisamine
  - 4.4. Rakendus: Viimistlemine
  - 4.5. Rakendus: Äpi paigaldus
- Kasutatud kirjandus ja materjal

+ Add a custom sidebar

Clone this wiki locally

<https://github.com/ios-r>

Clone in Desktop