TARTU ÜLIKOOL
Arvutiteaduse instituut
Informaatika õppekava

Ingrid Kaasik

Käsuklotsisüsteemi prototüüp arenduskeskkonnale Thonny

Bakalaureusetöö (9 EAP)

Juhendaja(d): Aivar Annamaa

Tartu 2017

Käsuklotsisüsteemi prototüüp arenduskeskkonnale Thonny

Lühikokkuvõte:

Töö eesmärgiks oli luua programmeerimiskeskkonnale Thonny prototüüp käsuklotsisüsteemist. Käsuklotsisüsteem on algajatele mõeldud programmeerimiskeskkond, kus on valikus programmeerimiskäskudele vastavad klotsid ning neid saab lohistada ja seeläbi luua programme. Prototüübis on võimalik luua lihtsamaid programme klotse lohistades, tõlkida need programmeerimiskeelde Python ning muuta Pythoni kood klotsideks.

Võtmesõnad:

Thonny, Python, käsuklotsisüsteem, programmeerimine

CERCS: S281 - Arvuti õpiprogrammide kasutamise metoodika ja pedagoogika, P170 - Arvutiteadus, arvutusmeetodid, süsteemid, juhtimine (automaatjuhtimisteooria)

Prototype of Block-based Programming System for Thonny Integrated Development Environment

Abstract:

The purpose of this paper is to create prototype of block-based programming environment to Thonny. Block-based programming environment is an environment that allows user to construct a computer program with draggable blocks. With the prototype it is possible to create simple computer programs with blocks, translate these blocks into programming language called Python and turn the program in Python into predefined blocks.

Keywords:

Thonny, Python, Block-based programming system, programming

CERCS: S281 - Computer-assisted education, P170 - Computer science, numerical analysis, systems, control

Sisukord

Sissejuhatus

Programmeerimisega alustajal on palju raskusi. Enamikule alustajatest on võõras programmeerimiseks vajalik töökeskkond ja programmeerimiskeele süntaks. Lisaks sellele on raskuseks ka soovitud käskude kirjutamine programmiks ehk enda soovide väljendamine programmeerimiskeeles.

Süntaksi probleemi aitab kergemaks teha käsuklotsisüsteem, mis on laiemalt tuntud lastele ja noortele mõeldud programmeerimiskeskkonna Scratchi kaudu. Scratchist tuleb rohkem juttu alampunktis 1.1.1. Käsuklotsid on visuaalsed klotsid, millega väljendatakse programmeerimiskeele konstruktsioone.

Thonnyt kasutatakse peamiselt Tartu Ülikooli veebikursusel "Programmeerimisest maalähedaselt" ja kursusel "Programmeerimine". Kumbki kursus ei eelda eelnevaid teadmisi programmeerimises ning nende peamine eesmärk on õpetada programmeerimise põhitõdesid. Praegune versioon Thonnyst on töökindel, kuid vajab täiendusi, et muuta alustavatele programmeerijatele programmide tööprotsess kergemini mõistetavaks.

Thonny arendamisega tegi algust Aivar Annamaa. Autori ülesandeks on luua Thonny jaoks prototüüp käsuklotsisüsteemist, mis oleks visuaalselt võimalikult sarnane süsteemiga Scratch. Kuna Scratchi õpetatakse paljudes Eesti koolides, siis tuttavlikkus klotsidega muudab õppeprotsessi kiiremaks.

Käsuklotsisüsteemiga saab õppija panna olemasolevatest klotsidest kokku programmi ning pärast tõlkida selle Pythoni koodiks. See aitab õppijal mõista, kuidas muutub visuaalselt ilmutatud kujul antud programmi struktuur korrektse süntaksiga koodiks. Sel juhul ei pea algaja programmi loomisel pöörama tähelepanu korrektsele süntaksile ning veateadetele, mis tunduvad esmapilgul hirmuäratavad.

Pythoni koodi on võimalik ka muuta käsuklotsideks. Niimoodi saab õppija muuta keeruliselt struktureeritud ning pika koodi visuaalselt mugavamaks ja algajale programmeerijale kergemini mõistetavaks. See funktsionaalsus muudab õppijale koodi lugemise hõlpsamaks ning vähendab koodi lugemisega seotud hirmu.

Käsuklotsisüsteeme on väga palju. Kuna enamik neist on suunatud lastele, siis need on mängulised ja keskenduvad peamiselt mängude loomisele. Mõned käsuklotsisüsteemid on lisaks mängudele suunatud ka natuke vanemale vanusegrupile ning neil on juurde toodud klotside tõlkimine

tekstilisteks programmeerimiskeelteks ning kasutajal on võimalik vahetada klotsivaate ja koodivaate vahel. Uue käsuklotsisüsteemi jaoks on aga vajadus olemas, kuna praegu ei leidu ühtegi süsteemi, mida saaks integreerida Thonny süsteemi. Käsuklotsisüsteemi ja Thonny ühendamine on õpilasele mugav, kuna siis pole vajadust alla laadida ja kasutada erinevaid programme klotside tõlkimiseks ning hiljem tekstilise programmeerimise alustamiseks.

Töö koosneb neljast peatükist. Esimeses peatükis on erinevate käsuklotsisüsteemide võrdlus. Teises peatükis tuuakse välja selliste süsteemide eelised ja puudused. Programmi allalaadimise ja käivitamise juhend on kolmandas peatükis ning neljandas kirjeldatakse kasutajaliidest ja kuidas seda kasutada. Viies peatükk kirjeldab tööprotsessi ning kuidas prototüüpi edasi arendada, et seda saaks pistikprogrammina Thonnys kasutusele võtta.

1. Erinevate käsuklotsisüsteemide võrdlus

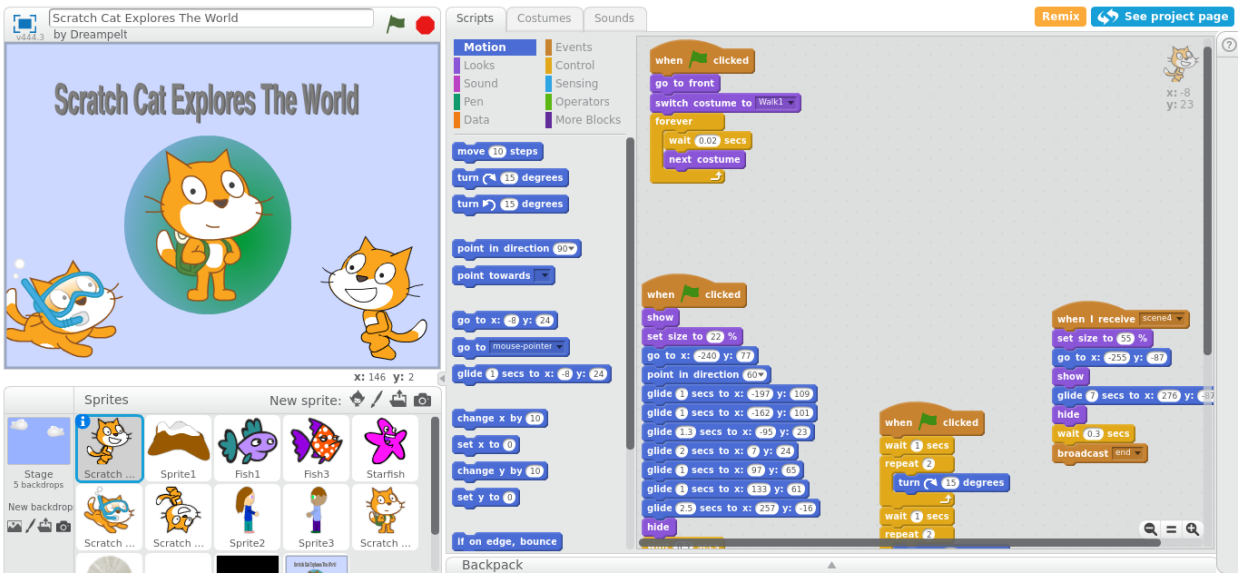
Käsuklotsisüsteemi kasutatakse väga palju ning neid on väga erinevaid. Autor toob välja pigem programmeerimise õpetamisele suunatud süsteemid. Lisaks õpetamiseks mõeldud süsteemidele on ka lihtsalt käsuklotsisüsteeme, millega saab luua oma mänge või programme ning mis ei ole suunatud otseselt programmeerimise õpetamisele (näiteks Stencyl [1], millega saab luua arvutimänge erinevatele platvormidele). Väga palju vanemaid süsteeme on kirjeldanud Kelleher ja Pausch [2].

1.1 Visuaalsed käsuklotsisüsteemid

Visuaalsed käsuklotsisüsteemid on programmid, mis muudavad algajatele programmeerimiskeele kergemini mõistetavaks. Nende süsteemide puhul ei keskenduta kindla programmeerimiskeele süntaksile, vaid üldiste programmeerimiskäskude ja -võtete mõistmisele.

1.1.1. Scratch

Scratch on kasutatuim programmeerimise õpetamiseks mõeldud käsuklotsisüsteem, millel on maailmas üle 15 miljoni kasutajaga [3]. Kasutaja saab klotse lohistades luua programmi, mida hakkab täitma valitud Sprite. Sprite on iseseisev kahe dimensiooniline pilt, mis on osa suuremast pildist. Scratchi kasutajaliidest näeb Joonisel 1.



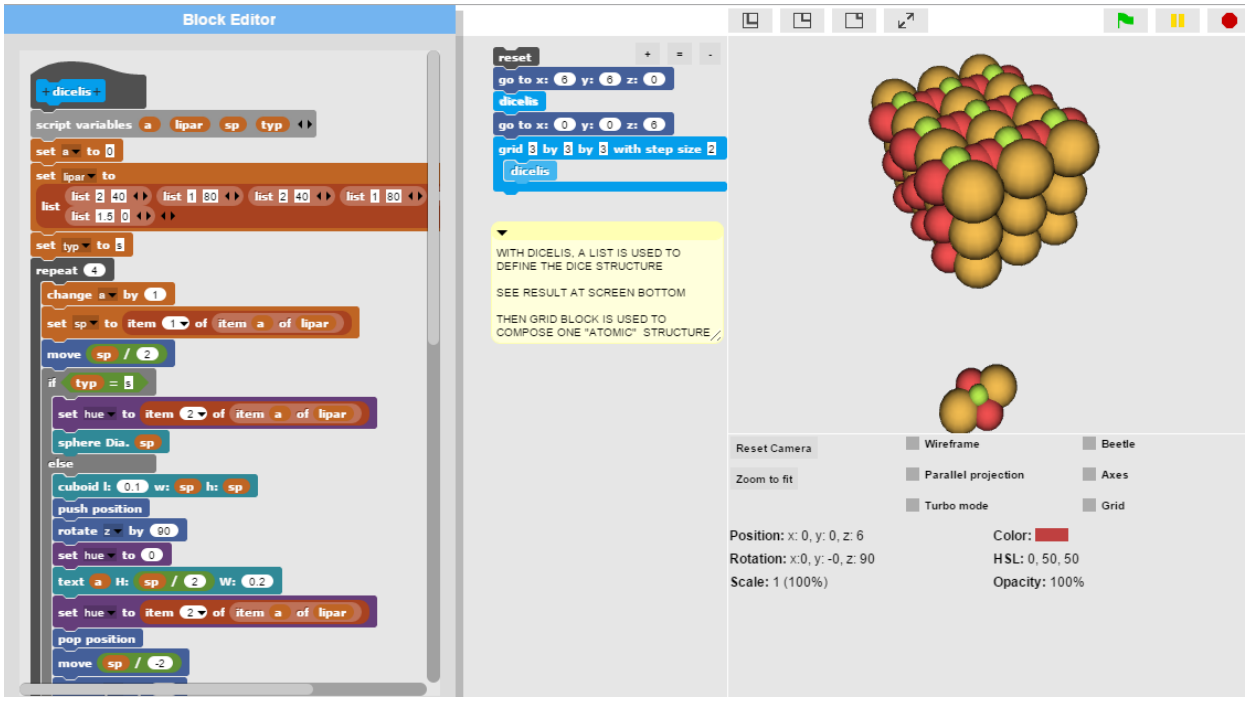
Joonis 1. Näide Scratchi projektist [4].

Scratch [5] sai alguse 2003. aastal Massachusettsi Tehnoloogiainstituudis Lifelong Kindergarden grupis. Scratch keskendub peamiselt lastele programmeerimise õpetamisele. Seega on selles programmis palju vahendeid visuaalsete projektide loomiseks. Projekti valmis saamisel on võimalus

vaadata ka loodud klotsiprojekti JavaScripti koodis. Lisaks oma projektide loomisele on võimalik kellegi teise nõusolekul tema projekti muuta ja edasi arendada. See loob võimaluse teineteise aitamiseks. Scratch muudab programmeerimise mänguliseks ja positiivseks kogemuseks.

1.1.2. BeetleBlocks

BeetleBlocks [6] on Scratchil põhinev süsteem, mis lisab tavalisele Scratchi funktsionaalsusele 3D graafika. Joonisel 2 on näha BeetleBlocksi kasutajaliidest ning valmiva mudeli näidist.



Joonis 2. Näide BeetleBlocksi projekti kasutajaliidesest ja selle loodud 3D mudelist [7].

Klotsid on üldiselt samad nagu Scratchis. Selleks, et lihtsustada laste jaoks ruumilist mõtlemist, käib programmi koostamine mardika liikumistee järgi. Kui liikumistee on paigas, muutub see tee ise 3D mudeliks. BeetleBlocksi arendus veel kestab.

1.1.3. Blockly

Google arendas oma klotsisüsteemi Blockly [8] aastal 2011. Blockly kasutajaliides ja näidisprojekt on näha joonisel 3.

Joonis 3. Näide Blockly projektist [9].

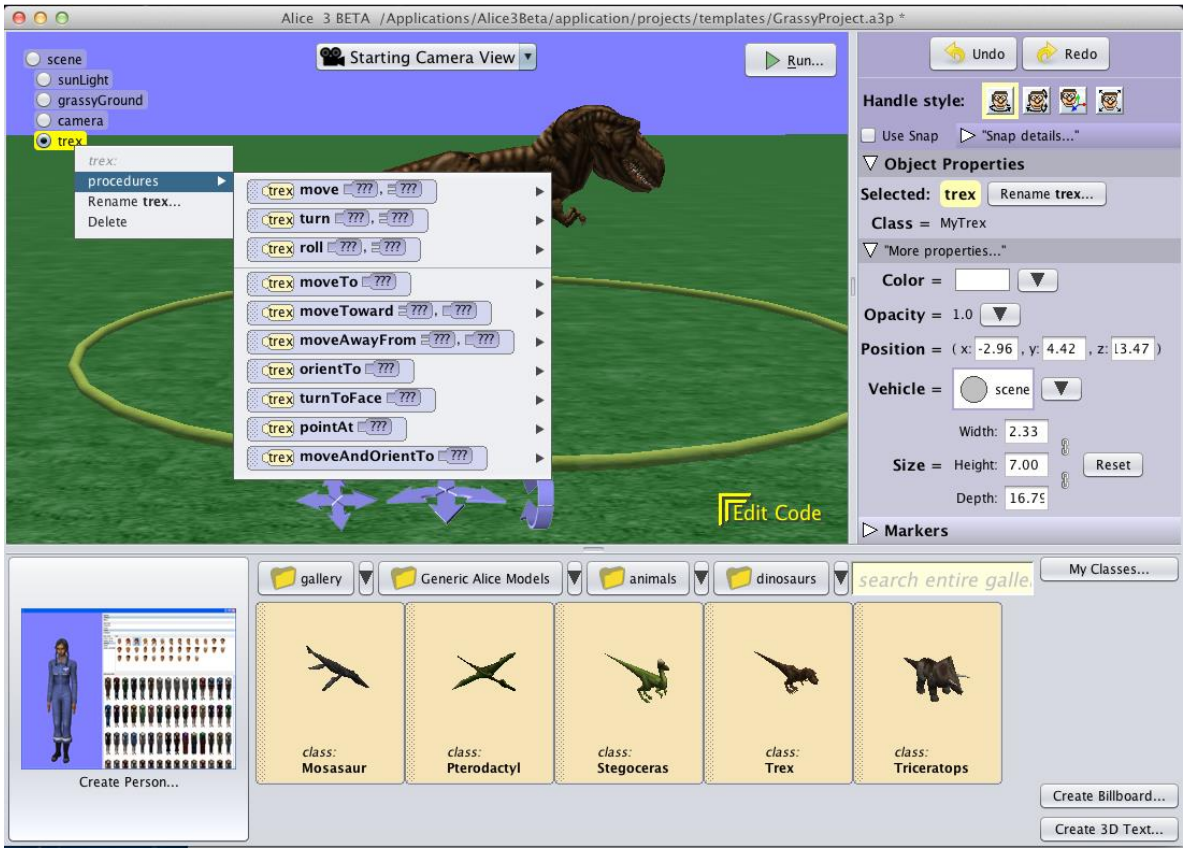
Visuaalselt erineb Blockly Scratchist oma klotside poolest. Blockly süsteemis saab lisada klotse nii vertikaalselt kui horisontaalselt. Viimast ei ole Scratchis võimalik teha. Hoolimata sellest, et Blockly on suunatud tulevastele arendajatele, on nende veebilehel ka õpetavaid mänge lastele. Erinevalt paljudest klotsisüsteemidest toetab Blockly klotsikoodi tõlkimist mitmetesse programmeerimiskeeltesse, sealhulgas JavaScript, Python, PHP ja Dart. Seda süsteemi kasutatakse laialt tuntud veebilehel Code.org [10], mille abil korraldatakse üle maailma projekti "kooditund". Blockly süsteemist arendati ka visuaal-tekstiline veebirakendus BlockPy [11], kus kasutaja saab vaadata klotsivaadet ja koodivaadet ning koostada programmi neis mõlemas. BlockPy kasutab tekstilise programmeerimiskeelena Pythonit, kuigi klotsid on muutmata kujul Blockly süsteemist.

1.1.4. Pocket Code

Pocket Code [12] on Scratchist inspireeritud programm, millega saab luua mänge, videosid ja tervituskaarte mobiilis või tahvelarvutis. Sellega arendatud programmid on vaadatavad ainult nutitelefonides ja tahvelarvutites ekraani suurusega kuni 7 tolli.

1.1.5. Alice

Süsteemi Alice [13] loomist alustati juba aastal 1997. Erinevalt paljudest teistest süsteemidest keskendub Alice objekt-orienteeritusele ning on suunatud 3D graafikale. Muidu on Alice üsna sarnane Scratchile. Alice'i [2] arendamine toimus mitmes etapis, mida autor kirjeldab järgmises lõigus. Joonisel 4 on näha viimase etapi Alice 3 projekti näidis.



Joonis 4. Programmi Alice 3 projekti näide [14].

Alice98 on sarnane programmeerimiskeelele Python. Juba esialgsel süsteemil Alice98 on 3D graafika võimalused ning objektpõhisus, kuid see ei ole veel käsuklotsisüsteem. Alice98 tagasisidest saadi teada, et trükkimine valmistab kasutajatele probleeme. Seega arendati Alice99, kus saab luua animatsioone klotside lohistamise abil. Peale tegelase lõuendile lohistamist ilmub valik selle tegelase tegevustest. Keerulisemate programmide puhul peab siiski kasutaja trükkima koodi. Järgnevas iteratsioonis, Alice 2, arendati klotse edasi ning klotside lisafunktsionaalsus annab kasutajale võimaluse ainult klotsidega luua keerukaid programme. Alice 3 [15] arendamisel muutus baaskeel Javaks ning lisati võimalus animatsioone Youtube'i üles laadida. Alice 3 muutus objektipõhisest süsteemiks objekt orienteeritud süsteemiks.

1.2 Füüsiliste klotsidega käsuklotsisüsteem Tern

Tern [16] on virtuaalsete ja füüsiliste robotite kontrollimiseks mõeldud programm, mis loodi Tufts'i ülikoolis. See süsteem on eelnevalt kirjeldatud süsteemidest väga erinev, kasutades vituaalsete klotside asemel päris puust klotse. Joonisel 5 on näha puust klotsidest loodud programm.

Joonis 5. Programmeerimiskeele Tern puidust klotsid [17].

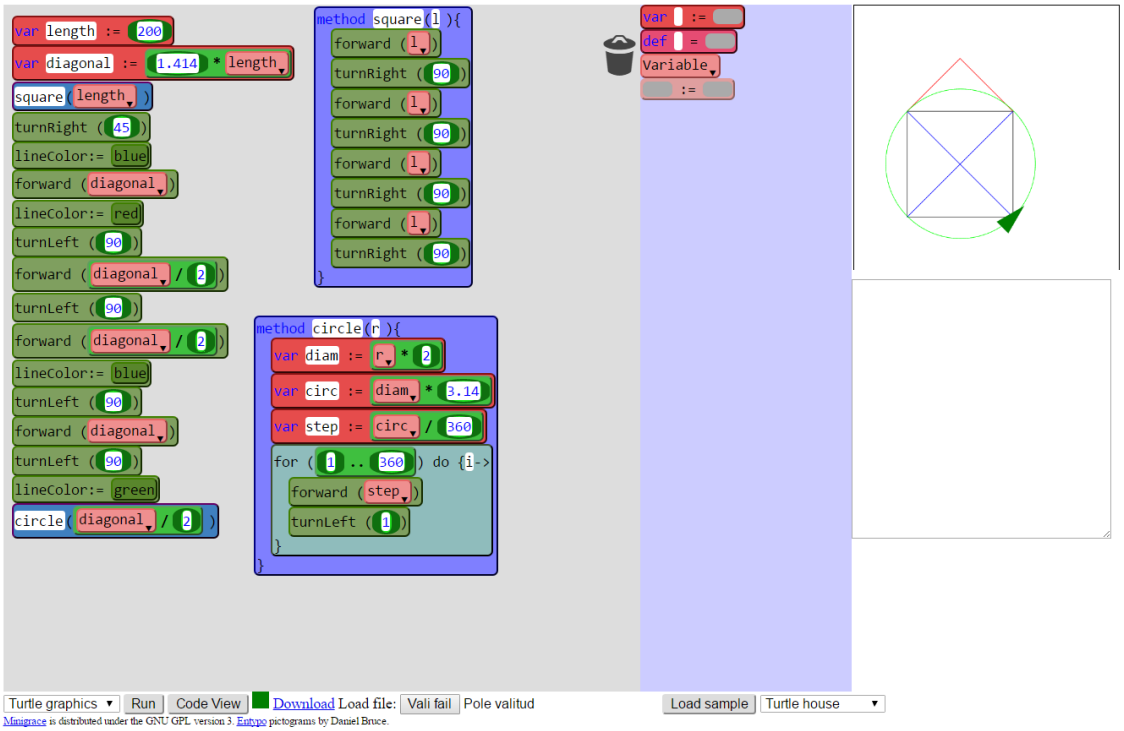
Kasutaja paneb puuklotsidest kokku käskude jada, suunab veebikaamera klotsidele ja arvutist programmi käivitades hakkab robot neid käske täitma. Kuna veebikaamera loeb sisse käsuklotsid, siis klotsidele ei ole lisatud elektroonikat, andureid ega sensoreid, tegemist on tavaliste piltidega puidust klotsidega.

1.3 Visuaal-tekstilised käsuklotsisüsteemid

Visuaal-tekstilistes käsuklotsisüsteemides on võimalik programmeerida nii klotsidega kui ka teksiga. Nende programmide eesmärk on aidata kasutajal klotsisüsteemidelt minna edasi tekstilisele programmeerimiskeeltele.

1.3.1. Tiled Grace

Tiled Grace [18] on veebipõhine programm, kus saab vahetada klotside ja koodi vaadete vahel. Joonisel 6 on näha klotsi vaade ning joonisel 7 on koodi vaade.



Joonis 6. Programmi Tiled Grace klotsi vaade.

Joonis 7. Programmi Tiled Grace koodi vaade.

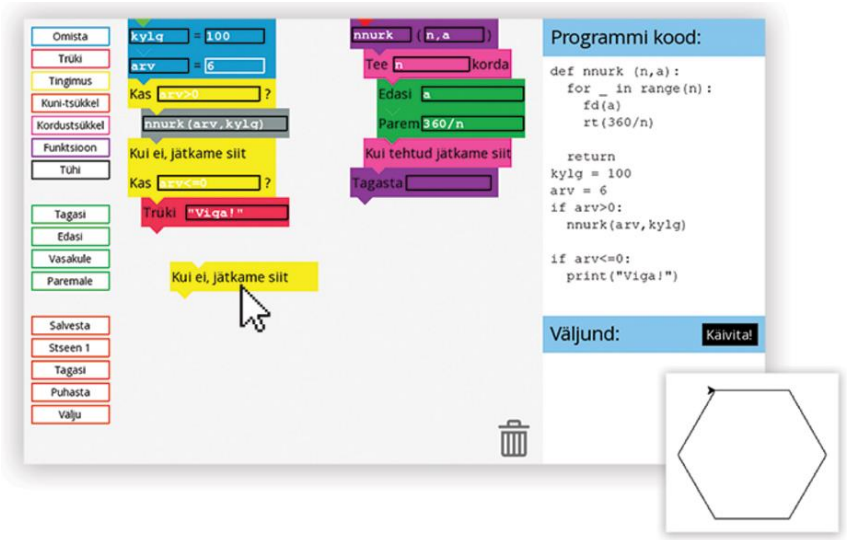
Mõlemas vaates saab teha muudatusi, mille saab ümber tõlkida teise vaatesse. Vaadete vahetamisel toimub animatsioon, mis muudab kas klotsid koodiks või koodi klotsideks, et kasutajal oleks võimalikult kerge näha seoseid klotside ja koodi vahel.

1.3.2. Greenfoot

Greenfoot [19] on programm, mille eesmärgik on õpetada programmeerimiskeelt Java mängude ja simulatsioonide kaudu. Selles programmis ei ole eraldi koodi ja klotsi vaadet, vaid muudetav kood asub klotsides. Parema hiireklõpsuga avaneb menüüriba ning kasutaja saab sealt valida sobivaid käske. Kui kasutaja kirjutab ise klotsidesse, siis Greenfoot soovitab reaalajas võimalikke meetodeid ja pakub automaatset lõpetust.

1.4 Monty

Monty [20] loodi aastal 2014 Tartu ülikoolis loetavas "Automaadid, keeled ja translaatorid" aines. Joonisel 8 on Monty näidisprojekt. Montyga saab lohistada eestikeelseid klotse ning kõrval koodiaknas muutub kood reaalajas. Koodiaknasse kirjutamine pole aga võimalik. Montyt pole võimalik integreerida Thonny süsteemi, sest Monty kasutab PyGame'i teeki, aga Thonny jaoks on vajalik Tkinteri teek.



Joonis 8. Monty kasutajaliides [21].

2. Käsuklotsisüsteemi eelised ja puudused

Programmeerimist sissejuhatavad ained on algajatele programmeerijatele üsna keerulised. Programmeerimiseks on vaja täiesti teistsuguseid oskusi kui oleme harjunud kasutama. Tuleb omandada algoritmiline mõtlemine. Suurem osa õpilastest on harjunud mõtlema üldiselt, kuid oma mõtete programmiks tõlkimisel läheb vaja täpsust ja algoritmilist mõtlemist. Samal ajal peab aga õppima selgeks programmeerimiskeele süntaksi. See võib aga osutuda eriti keeruliseks, kuna vea korral näidatakse tihti veateateid, millega algajad ei ole kokku puutunud ning mida nad ei oska lahendada. Sulu või koma puudumisel viitab veateade tihtipeale järgmisele programmi elemendile mitte aga vea kohale.

Kuna inimesi heidutab mitme probleemiga korraga tegelemine, soovitavad A. Robins, J. Rountree ja N. Rountree [22] keskenduda ühe uue oskuse omandamisele korraga ning järjest vähendada probleeme, mis võivad tekkida. Nii läheb õppimine ladusamalt ning on näha, kus täpselt õpilastel probleeme tekib ehk eristada, kas õpilasel on pigem raskusi süntaksist arusaamisega või enda väljendamisega programmis [23].

D. Malan ja H. Leitner [24] uurisid, kuidas mõjutas klotsisüsteemiga Scratch alustamine õppijate motivatsiooni ja arengut. Tegemist oli arvutiteaduse esimese aasta Harvardi tudengitega. Scratch lihtsustab õppimist, kuna see kaotab süntaksi ja süntaksivigadest tekitatud probleemid ning õpilane saab keskenduda pigem programmeerimise põhitõde õppimisele. Uurimusest selgus, et õpilastel oli terve kursuse vältel kõrge motivatsioon, ka Java programmeerimiskeelele üle minnes jäi motivatsioon kõrgeks. Õpilased ise kommenteerisid, et kuigi nad ei teadnud, kuidas mõningaid Java konstruktsioone rakendada, siis neil oli teada, milliste meetoditega jõuda soovitud tulemusteni.

Scratchiga alustajatel on küll kõrge motivatsioon, kuid see ei pruugi vähendada väljakukkujate arvu kursuselt ning tavakursuse läbinutega võrreldes jäävad õpilaste tulemused sarnaseks. Sellele järeldusele jõudis I. K. Kereki [23], kes uuris Uruguay ORT Ülikooli arvutiteaduse tudengeid. Mõningatel juhtudel tõid õpilased välja, et Scratchiga alustamine hoidis neid tagasi ning nad oleks tekstilise keelega rohkem õppinud [24]. Samas uuringus osalenud algaja tõi välja, et Scratchi ja Java vahel on liiga suur erinevus ning see tekitas pettumust.

Klotsisüsteemi kasutamisel ilmnesid algajate seas laialt levinud halvad harjumused. Õpilased lohistasid nende meelest vajalikud klotsid programmiväljale ning hakkasid neid siis kokku sobitama. Nii lõid õpilased palju tsükleid ja if-lauseid, mida oleks saanud ühendada. Probleeme tekitas hiljem loodud koodi loogikavigade leidmine [25].

Õpilased, kes kasutavad ainult käsuklotsisüsteeme, on harilikult oma programmeerimisoskustes vähem enesekindlad [26]. Seetõttu on loodud ka mitmeid visuaal-tekstilisi süsteeme. Nendes on õpilasel võimalik vahetada klotsisüsteemi ja tekstilise programmeerimiskeele vahel. D. Bau [26] tõi välja oma uurimuses, et algajad küll eelistavad pigem klotsisüsteemi, kuid 26% õpilastest vahetas pidevalt klotsisüsteemi ja tekstilise kooditöötluse vahel. Nad lisasid, et enamik õpilasi läksid aja jooksul üle tekstilisele programmeerimisele, kuid vähem motiveeritud kasutajad, kes jäid sõltuma klotsisüsteemist lõpetasid hiljem programmi kasutamise.

Käsuklotsisüsteemil on nii eeliseid kui ka puuduseid. Tundub, et parimaid tulemusi saab selle süsteemiga, kui seda kasutada esimestel nädalatel ning suunduda siis tekstilise programmeerimiskeele juurde. Kuna esimesed nädalad on üsna heidutavad õpilastele, siis selle aja jooksul saab juba programmeerimise põhimõtetega tuttavaks ning on aeg hakata süntaksit õppima. Kui kasutada käsuklotsisüsteemi liiga pikalt, tekivad halvad harjumused.

Üks võimalus mainitud probleemide lahendamiseks on lubada varasema programmeerimiskogemusega õpilastel käsuklotsisüsteemi vahele jätta ning kohe tekstilise programmeerimisega alustada. Niimoodi on algajatel motivatsiooni ja edasijõudnutel ei ole liiga lihtne.

3. Prototüübi käivitamine

Selles peatükis kirjeldab autor, kuidas alla laadidaendale käesoleva tööga valminud programmi. Lisaks toob ta välja, milliseid programme on selle tööks vaja ning kuidas seda käivitada. Autor soovitab kasutada Thonny IDE-t, kuna see teeb eeltöö hõlpsamaks.

3.1 Prototüübi käivitamiseks vajalikud programmid

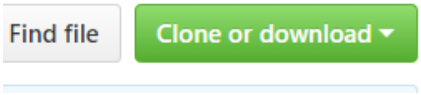
Prototüübi käivitamiseks on kasutajal vaja Python 3.5, mille saab tõmmata veebilehelt https://www.python.org/downloads/. Lisaks sellele on vajalik Pythoni IDE. Autor soovitab alla laadida Thonny [27], kuna Thonnyga on kaasas Python 3.5, ja nii pole Pythonit tarvis eraldi alla laadida.

3.2 Prototüübi tõmbamine

Prototüüp on avalikult saadaval aadressil: https://github.com/ingrid94/baka.

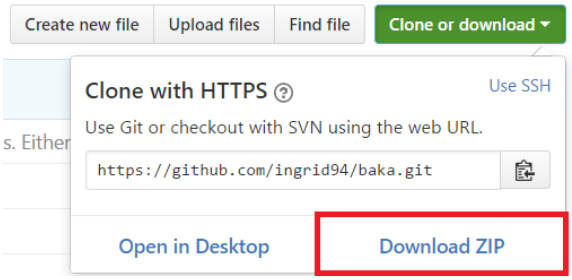
Projekti tõmbamiseks tuleb toimida järgnevalt:

1. Ava projekti link.
2. Vajuta rohelisele nupule akna paremal pool ‘Clone or download’. See nupp on kujutatud joonisel 9.



Joonis 9. Nupp ‘Clone or download’ githubis.

3. Avanenud aknas vajuta nupule “Download ZIP”, mis on joonisel 10 punase kasti sees.



Joonis 10. Punasega tähistatult nupule vajutades, laeb alla kasutaja endale projekti ZIP-failina.

4. Paki projekt lahti kausta, kust on seda lihtne leida.

3.3 Prototüübi käivitamine

Prototüübi käivitamiseks tuleb tööle panna eelnevalt installitud Pythoni IDE. Kuna programm on mõeldud Thonnyle, siis edaspidi eeldab autor, et tõmmatud Pythoni IDE on Thonny.

Järgmiseks tuleb avada IDEga projekt. Kui kasutaja ei kasuta Thonnyt, tuleb üle kontrollida, kas projekt käivitub õige Pythoni versiooniga ning vajadusel õige versioon valida.

Prototüübi käivitamiseks on vaja teha järgmised sammud:

1. Ava Thonny.
2. Vali põhimenüüst View → Files, mille juurde ilmub linnuke.
3. Vali aknas avanenud osas kaust, kuhu projekt on eelmises punktis salvestatud.
4. Ava Pythoni fail ‘main.py’.
5. Vajuta klaviatuuril klahvi F5 või tööriistaribal olevat rohelist nuppu, mis on piiritletud joonisel 11 punase ruuduga.

Joonis 11. Prototüübi käivitamise nupp, millel on punane kast ümber.

6. Nüüd peaks olema avanenud kolm uut akent.

4. Loodava süsteemi kasutamine

Selles peatükis kirjeldab autor kasutajaliidest, seletab lahti disainiotsuseid ning kirjeldab kuidas programmi kasutada.

4.1 Prototüübi aknad

Prototüübi käivitamisel avanevad kolm akent, mida on näha joonisel 12.

Joonis 12. Prototüübi käivitamisel tekkinud kolm akent.

Ülemises vasakpoolses aknas on kaks nuppu ‘To code’ ja ‘To blocks’. Alumises vasakpoolses aknas on näha kõik võimalikud valitavad klotsid prototüübis. Parempoolses aknas on alguses ainult prügikasti kujutis. Prototüüpi on kõige lihtsam sulgeda, kui panna nuppude aken kinni. Teiste akende sulgemisel kaob ainult konkreetne aken.

4.2 Klotside loomine ning kustutamine

Vasakus alumises aknas on valik klotsidest. Kasutaja valib sobiva klotsi, vajutades sellele peale, ning seejärel ilmub valitud klots parempoolsesse aknasse. Muutuja, muutuja väärtustamise, numbri ning sõne loomisel ilmub lisadialoog parema akna peale ning kasutaja saab kirjutada vastava

muutuja nime, numbri või sõne, mida soovib klotsile. Dialoogis on kirjeldatud, milline peab olema sisestatav tekst, et see vastaks reeglitele. Vale sisendi korral ilmub uus aken veateatega.

Parema akna üleval paremal nurgas on prügikasti ikoon. Kui kasutaja soovib klotsi või ühendatud klotse kustutada, siis lohistab ta klotsi prügikasti ikooni kohale kuni prügikasti ikooni ümber tekib punane joon. Sel hetkel hiirenuppu lahti lastes kustuvad klotsid.

4.3 Klotside ühendamine ja lahti ühendamine

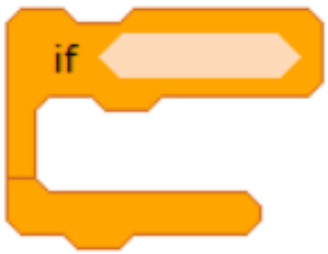
Paremas aknas on klotsid lohistatavad. Kui kasutaja lohistab klotsi teisele klotsile lähedale, siis ilmub teise klotsi külge nende ühenduse kohale roheline joon. Klotsi lahti lastes liigub klots ühenduskohale ning klotsid on ühendatud. Klotside lahti ühendamiseks tuleb lohistada klotsi, mida kasutaja tahab lahti ühendada, kuni roheline joon kaob. Sel juhul klots ei ühendu lahti lastes uuesti.

4.4 Klotside kujud

Klotsidest on olemas lausete klotsid ja sinised avaldiste klotsid. Lausete klotside seas on kollased tingimusklotsid ning punased avaldislause klotsid.

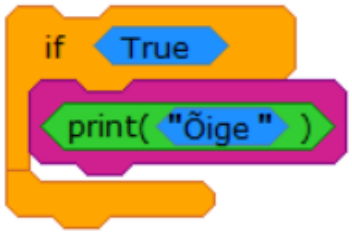
4.4.1 Tingimusklotsid

Tingimusklotsid on C-kujulised klotsid, mille eesmärgiks on kontrollida programmis antud tingimust. Esialgne idee oli luua see klots kastina, milles on tühimik, kuhu lisada käsud, mis käivituvad tingimuse täitumisel. Kuna tekkis mitteühilduvus Linuxis ja Windowsis, pidi jääma C-kujuliste klotside juurde. Tühi tingimusklots *if* on näha joonisel 13.



Joonis 13. Tühi tingimusklots if.

Scratchi arendustiim tõi välja, et C-kuju vähendab arendajate tööd, kuna C-klots ei pea kohanduma horisontaalselt teiste käsuklotside järgi. Klotsid, mis käivituvad tingimuse täitumise korral, lisatakse C-kuju vahemikku. Blockly [28] arendustiim tõi välja, et neil tekkis kogemata väike tühimik nende klotside ning tingimusklotsi alumise osa vahele. Seda tühimikku on näha joonisel 14 *print*-klotsi ja *if*-klotsi alumise osa vahel.



Joonis 14.Tingimusklots if, mis tingimuse täitmise korral väljastab sõne „Õige”.

See programmeerimisviga sai aga positiivset vastukaja, kuna kasutajad teadsid tühimiku pärast intuitiivselt, et tingimusklotsi sisse on võimalik lisada rohkem kui üks klots. Seetõttu rakendasin seda ka enda töös.

4.4.2. Avaldislause klotsid

Avaldislaused on laused, mille ülesandeks on mingi avaldise (näiteks funktsiooni väljakutse) väärtustamine. Selliste lausete väljendamiseks on loodud spetsiaalne tühi klots, kuhu saab sisestada avaldise klotsi. Joonisel 15 on klots funktsioonide välja kutsumiseks.

Joonis 15. Klots, mis on mõeldud funktsiooni välja kutsumiseks.

Muutujate väärtustamiseks on klots muutuja nimega, mida ei saa lahti lohistada, et kasutaja kogemata midagi muud sinna asemele ei paneks. Kuigi funktsioone ei ole prototüübis käsitletud, tegi autor *return*-klotsi, millega saab tagastada funktsioonide väärtusi. See loodi selleks, et tuua erinevus return- ja print-klotsi vahel. *Return*-käsk kuulub Pythoni keele juurde, aga *print*-käsk on eeldefineeritud meetod.

4.4.3. Avaldisklots

Erinevatele Pythoni avaldistele vastavad klotsid on äratuntavad kuusnurkse kuju ja sinise taustavärvi poolest. Joonisel 16 on võrdsuse väljendamise klots.

Joonis 16. Võrduse klots.

Neid klotse saab ühendada kõikide klotsidega, kus on heledam osa. Sisemised klotsid on võrdlemiseks, aritmeetilisteks arvutusteks, eeldefineeritud meetodite käivitamiseks, sõnede, numbrite, muutujate ning väärtuste *None, True* ja *False* väljendamiseks.

4.4.4. Eelkombineeritud klotsid

Prototüüpi on võimalik lisada ette valmistatud klotside komplekt. Praeguses programmis on olemas tühjast käsuklotsist ja sisemisest klotsist kokku pandud *print*-klots. Kuid võimalik on luua ka keerukamaid komplekte.

4.5 Nupud

Prototüübil on kolm nuppu. Esimene nupp on „To code", mis loob faili „generated.py" ning kirjutab sinna klotsid ümber vastavaks koodiks. Mitteühendatud sisemisi klotse programm ignoreerib. Selle all on nupp „To blocks", mis võtab pythoni koodi failist „to_blocks.py", mis on projekti kaustas olemas, ning loob vastavad klotsid parempoolsesse aknasse.

5. Töö protsess

Programmeerimine oli autorile alguses üsna keeruline, kuna ta ei olnud varem nii suurt kasutajaliidesega programmi teinud. Esimest korda kasutas ta ka Pythonit objekt-orienteeritud keelena, varasemalt pole tal Pythonis klassidega kokkupuudet olnud. Kogenematuse tõttu kulus alguses palju aega programmi planeerimisele ja kasutatavate meetodite välja mõtlemisele.

Lisaks kasutajaliidese loomise õppimisele, sai autor kinnitada teadmisi rekursioonist. Kuna klotsid olid omavahel rekursiivselt seotud ning lõpuks taandus programm puu struktuurile, siis rekursiivseid funktsioone tekkis palju.

Kõige huvitavam oli õppida ja meelde tuletada abstraktset süntaksipuud (AST). Kuna Pythoni koodi muutmisel klotsideks läks seda väga põhjalikult vaja, siis oli see ASTi õppimiseks väga hea programm. Kui ainest ‘Automaadid, keeled ja translaatorid’ õpitud ASTi puu sai meelde tuletatud, läks see programmeerimise osa üsna kiirelt.

Kuna programm osutus oodatust keerulisemaks ning ajakulukamaks, siis jäi see praeguseks prototüübiks ning pole veel ühendatud Thonny süsteemi pistikprogrammina.

Pistikprogrammi loomiseks on vaja teha järgmist:

- luua juurde klotse, näiteks aritmeetika, for-tsükkel, listidega tegelemine ning funktsioonide loomine ja välja kutsumine;
- kood → klotsid teisendus peaks toimuma Thonnys lahti olevast Pythoni faili põhjal;
- klotsid → kood teisendus peaks toimuma Thonnys lahti olevast Pythoni faili põhjal ning kohanduma olemasoleva koodiga, näiteks reavahetustega ning kommentaaridega;
- luua võimalus kasutajal ise klotside komplekt salvestada, et neid hiljem kasutada;
- testida programmi rohkem ning lisada piiranguid, et ei tekiks võimalusi lisada klotse valedesse kohtadesse;

6. Kokkuvõte

Käsuklotsisüsteem teeb algajale programmeerijale programmeerimiskeele õppimise lihtsamaks. Kuna klotsid on ees, siis ei pea programmeerimiskäske pähe õppima. Lisaks ei teki süntaktilisi vigu, millest tingitud veateated võivad alguses segased olla. Thonny jaoks on eraldi pistikprogrammi vaja, kuna algajale programmeerijale on mugavam, kui kasutatavad programmid on koos.

Töös toodi välja erinevaid klotsisüsteeme. Need jagas autor kaheks: visuaalsed ning visuaal-tekstilised käsuklotsisüsteemid. Visuaal-tekstilised süsteemid arendati selleks, et üleminek klotsisüsteemilt tekstilisele programmeerimiskeelele oleks sujuvam. Visuaalsetest süsteemidest kirjeldas autor Scratchi, BeetleBlocksi, Blocklyt, Pocket Code'i ning Alice'it. Visuaal-tekstilisest süsteemidest tõi autor välja Tiled Grace'i ning Greenfoot'i. Eraldi kirjeldati füüsiliste klotsidega süsteemi Tern ning algselt Thonny jaoks mõeldud süsteemi Monty.

Käsuklotsisüsteemidel on nii eeliseid kui ka puuduseid. Eelistena tõi autor välja kõrge motivatsiooni terve kursuse jooksul ja ühe probleemiga korraga tegelemise. Hoolimata kõrgest motivatsioonist, õpilaste väljakukkumiste arv ei muutunud. Puudusena tuli välja, et mõnel juhul õpilaste motivatsioon langes ning oleks soovinud klotsisüsteemi vahele jätta.

Töö tulemusena valmis kasutatav Pythonil põhinev käsuklotsisüsteemi prototüüp, millest saab tulevikus edasi arendades Thonny pistikprogramm. Programmi valmimisel kasutatakse seda algajate programmeerijate õpetamisel Tartu Ülikooli ainetes „Programmeerimine" ja „Programmeerimisest maalähedaselt".

7. Kasutatud materjalid

[1] Stencyl'i kodulehekülg: http://www.stencyl.com/ (05.01.2017)

[2] Kelleher, Caitlin; Pausch, Randy. Mai 2003. Lowering the Barriers to Programming: a survey of programming environments and languages for novice programmers.

[3] Scratchi statistika: https://scratch.mit.edu/statistics/ (03.01.2017)

[4] Illustratiivne projekt Scratchis: https://scratch.mit.edu/projects/103776594/ (05.01.2017)

[5] Info Scratchist: https://scratch.mit.edu/about/ (03.01.2017)

[6] Into BeetleBlocksist: http://wiki.scratch.mit.edu/wiki/BeetleBlocks (03.01.2017)

[7] Pilt BeetleBlocksi projektist: http://www.xleroy.net/ByobTuto/Beetleblocks/dicelis-code.html (05.01.2017)

[8] Blockly kodulehekülg: https://developers.google.com/blockly/ (03.01.2017)

[9] Blockly näidisprojekt: https://blockly-games.appspot.com/turtle (05.01.2017)

[10] Info Blocklyst: https://developers.google.com/blockly/about/showcase (05.01.2017)

[11] BlockPy kodulehekülg: http://think.cs.vt.edu/blockpy/ (05.01.2017)

[12] Pocet Code'i tõmbamine ja info: https://play.google.com/store/apps/details?id=org.catrobat.catroid (05.01.2017)

[13] Alice'i kodulehekülg: http://www.alice.org/index.php (03.01.2017)

[14] Alice 3 projekti pilt: http://www.cs.duke.edu/csed/alice/aliceInSchools/workshop11/alice3/SceneXEditor.png (05.01.2017)

[15] Info Alice 3 kohta: https://www.cmu.edu/homepage/computing/2009/summer/alice-3-released.shtml (03.01.2017)

[16] Terni kodulehekülg: http://hci.cs.tufts.edu/tern/ (03.01.2017)

[17] Pilt Terni klotsidest: http://hci.cs.tufts.edu/tern/images/tangk.jpg (05.01.2017)

[18] Tiled Grace: http://homepages.ecs.vuw.ac.nz/~mwh/minigrace/tiled/ (05.01.2017)

[19] Greenfooti kodulehekülg: http://www.greenfoot.org/home (05.01.2017)

[20] Info Montyst: https://github.com/taivop/Monty/wiki/Motivation (05.01.2017)

[21] Pilt Montyst: https://camo.githubusercontent.com/7b9a6cc84d7c01c2b573abb58755279bc4100ff3/687474703a2f2f692e696d6775722e636f6d2f6a3247705849502e706e67 (05.01.2017)

[22] Robins, Anthony; Rountree, Janet; Rountree, Nathan. 2003. Learning and teaching programming: a review and discussion. Computer Science Education, Vol 13, No 2, 137-172.

[23] Friss de Kereki, Inés. Oktoober 2008. Scratch: Applications in Computer Science 1.

[24] Malan, David J., Leitner, Henry H. Märts 2007. Scratch for Budding Computer Scientists.

[25] Meerbaum-Salant, Orni; Armoni, Michal; Ben-Ari, Mordechai. Juuni 2011. Habits of Programming in Scratch

[26] Bau, David. Juuni 2005. DROPLET, A BLOCKS-BASED EDITOR FOR TEXT CODE. 138-144.

[27] Thonny kodulehekülg: http://thonny.org/ (05.01.2017)

[28] Blockly disain: https://developers.google.com/blockly/guides/app-integration/best-practices#2_nesting_sub-stacks (05.01.2017)

Lisad

I. Litsents

Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks

Mina, **Ingrid Kaasik**,
(*autori nimi*)

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) enda loodud teose
Thonny arenduskeskkonna käsuklotsisüsteemi prototüüp,
(*lõputöö pealkiri*)

mille juhendaja on **Aivar Annamaa**,
(*juhendaja nimi*)

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

Tartus, **05.01.2017**