

Tartu Ülikool

Loodus- ja täppisteaduste valdkond

Arvutiteaduse instituut

Martin Toode

Virtuaalne põgenemistuba Deltas

Bakalaureusetöö (9 EAP)

Juhendaja: Madis Vasser PhD

Tartu 2021

Virtuaalne põgenemistuba Deltas

Lühikokkuvõte:

Bakalaureusetööna valmis õpetlik veebirakendus, mis tutvustab Noored Koodi pakutavaid kursuseid. Mängu raames omandatakse algteadmised küberturvalisusest, veebilehe tööriistadest, Linuxist ja algoritmikast. Valminud töö õpetab kasutajale algteadmised eelnevalt mainitud teemadest. Bakalaureusetöös kirjeldatakse kasutatud tehnoloogiaid, vaadeldakse võimalikke lahendusi väljamõeldud ülesannetele ja analüüsitakse rakenduse testijaid.

Võtmesõnad: Veebirakendus, küberturvalisus, Linux, algoritmika, õppemäng

CERCS: P175 Informaatika ja süsteemiteooria

Virtual escape room at Delta

Abstract:

The aim of this Bachelor's thesis was to create an educational web application that introduces the program Noored Koodi courses. The application is exposed to various topics, such as cybersecurity, web development tools, Linux and algorithms. The focus of the formed application is to introduce basic knowledge of the previously mentioned topics. The first part of the thesis talks about the technologies used to create the application. The second part discusses possible solutions to the created challenges. The last part analyses the testers of the application.

Keywords: Web application, cybersecurity, Linux, algorithms, educational game

CERCS: P175 Informatics, systems theory

Sisukord

Sissejuhatus	5
1. Tarkvara, riistvara ja teekide valikud	6
1.1 Tahvelarvuti iPad Pro 2020	6
1.2 LiDAR skänner	6
1.3 Three.js	7
1.3.1 WebGL	7
1.3.2 OpenGL ES	8
1.3.3 Three.js kasutus	8
1.4 Blender	8
1.4.1 Modelleerimine ja skulptimine	8
2. 3D objektide käsitus Three.js'is	11
2.1 Scene	11
2.2 Objektid	11
2.3 Valgus	12
2.4 Liikumine ja suund	13
2.5 Octree	14
2.6 Ruumiliste objektide sisse laadimine	14
2.6.1 OBJLoader	14
2.6.2 GLTFLoader	14
2.6.3 DRACOLoader	15
3. Implementatsioon	16
3.1 Mängu ülevaade	16
3.2 Linux'i süsteemihaldus	16

3.2.1 Pwn Adventure 3: Pwnie island	17
3.3 Hanoi torn	18
3.3.1 Hanoi torni läbimine itereerides	18
3.3.2 Hanoi torni läbimine rekursiivselt	18
3.4 Muud ülesanded	20
4. Testimine	21
4.1 Tulemused	21
Kokkuvõte	25
Viidatud kirjandus	26
Lisad	28
I. Valminud veebirakendus	28
II. Mänguga seonduv video materjal	29
III. Koosolekutel käsitletud materjal	30
V. Litsents	35

Sissejuhatus

Informaatika valdkonna populaarsus on olnud pidevas kasvus. Tartu Ülikooli programm Noored Koodi viib tudengid nii põhikooli- kui ka gümnaasiumiõpilastele töötubasid õpetama [1]. Projekti raames on välja töötatud kursused koolitundides kasutamiseks. Projekt täidab Haridus- ja Teadusministeeriumi programmi “Digipööre” eesmärki[2]. Noored Koodi kursuste seas on 6 erinevat teemat: virtuaalreaalsus, turvalisus, veebilehed, algoritmid, Linux ja Raspberry Pi ning Arduino [3]. Koolidele tutvustatakse eelnevalt mainitud programme, koolitades õpetajaid ja tehes näidistunde. Huvitegevust ei tekitata õpilastele, kes soovivad õppida eelpool nimetatud teemasid iseseisvalt. Õpilastele saab tekitada huvi reklaamides programmi läbi teemakohaste huvitavate tegevuste. Eestis kehtestatud piirangute tõttu toimub õppetöö digilahendusi kasutades, mis omakorda tõstab õpilaste digipädevust.

Antud bakalaureusetöö käigus loodi veebipõhine mäng, mida nii põhikooli kui ka gümnaasiumi õpilased saavad mängida, eesmärgiga õppida erinevate IKT teemade kohta ja tekitada mängijates huvi erinevate erialade vastu. Mäng on 3D graafiline ja esimese inimese vaatepunkt perspektiivis. Põgenemistoa tegevus toimub Tartu Ülikooli Delta õppehoones virtuaalselt. Mängu vaheülesanded on IT-teemalised ja käsitlevad erinevaid infotehnoloogia õpingute suundi. Loodud mängu katsetatakse nooremate laste peal, et määrata loodud lahenduse keerukust.

Esimene osa bakalaureusetööst käsitleb praktilise töö käigus kasutatud tarkvara ja riistvara algteadmisi. Samuti vaadeldakse ka muid võimalikke alternatiive. Teises osas analüüsitakse kasutatud primitiivseid elemente Three.js'is. Kolmas osa bakalaureusetööst käsitleb loomisprotsessi. Peatükkides on kirjeldatud töö ülevaadet ning süvitsi käsitletud loodud ülesandeid ning nende võimalikke lahenduskäike. Viimases peatükis analüüsitakse testgrupilt saadud tagasisidet.

Bakalaureusetöö eesmärk on virtuaalse põgenemistoa loomine. Põgenemistuba toimub läbi veebipõhise keskkonna. Ruumiliste objektide loomiseks võetakse üles LIDAR-i tehnoloogiat kasutades Tartu Ülikooli Delta õppehoone kõige huvitavamaid ruume. Iluved korrastatakse 3D mudelit modelleerides. Mäng koos loogikaga luuakse veebirakendusena ja tehakse kasutatavaks veebikeskkonnas¹. Ülesannete teemad luuakse põhinedes olemasolevatele Noored Koodi programmi materjalidele.

¹ <http://nooredkoodi.ee/>

1. Tarkvara, riistvara ja teekide valikud

Selles peatükis käsitletakse bakalaureusetööga seotud tarkvara, riistvara ja teeke, mida kasutati praktilise töö loomisel. Riistvara valikul eelistati olemasolevat varustust CGVR labori poolt.

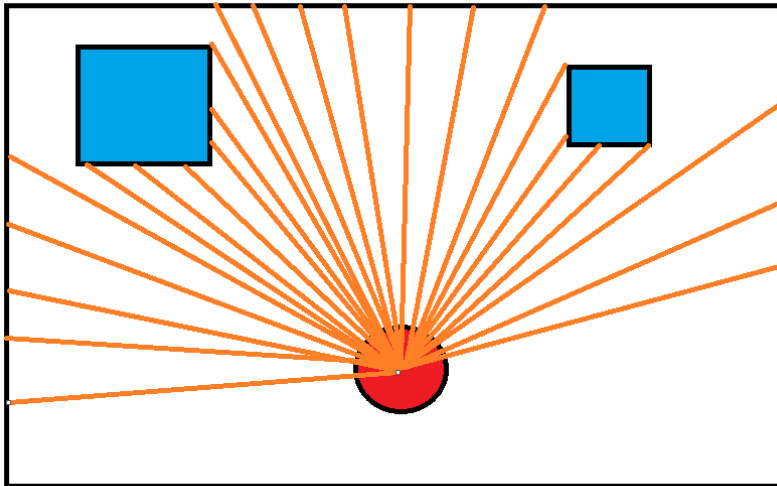
1.1 Tahvelarvuti iPad Pro 2020

iPad Pro on Apple poolt loodud tahvelarvuti. Aasta 2020 versioon sisaldab A12Z Bionic kiipi, mis võimaldab käsitleda keerukaid protsesse mõistliku aja raames. Tegemist on seadmega, millel on väga hea tarkvaraline tugi ning ülilainurk kaamera koos LiDAR skänneriga. Kõrge kvaliteetsed kaamerad, LiDAR skänner ja liikumissensorid teevad iPad Pro ideaalseks sissejuhatus liitreaalsusesse seadmeks [3].

Tahvelarvuti avaldamisel tehti saadavaks ka ARKit 4, mis võimaldab saada detailset informatsiooni ümbritsevast keskkonnast kasutades LiDAR tehnoloogiat [4]. Rakendusliides ARKit 4 põhjal on kirjutatud seadmele rakendusi, mille abil saab teha skanneeringuid ja luua 3D kujutus siseruumist reaalsete mõõtudega. Saadud kujutlust saame töödelda kasutatavaks formaati teegi Three.js jaoks.

1.2 LiDAR skänner

LiDAR, lühend “*Light Detection and Ranging*”, on meetod, kus kasutatakse pulseerivat laserit, et koguda informatsiooni ruumi mõõtmetest. Saadud väärtusi kasutatakse 3D mudeli loomiseks ruumist. LiDAR süsteem arvutab ajakulu, mis kulus pulseerival laserkiirel, et jõuda tagasi sensorini ning selle info põhjal saab teada, kui kaugel on objekt ruumis [5]. Alustades ühest kohast ruumis ja skanneerides ümbritseva keskkonna saame kätte sellest punktist olevate objektide kaugused (Joonis 1) ja kuna skännerid tulevad enamasti kaamera lisandina, siis saab salvestada ka esemete detailse visuaali. Liikudes ringi ruumis, skanneerides erinevatest kohtadest punkte, saab kätte ruumiliste objektide kuju ja informatsiooni.



Joonis 1. LiDAR skanneerimine punktist ruumis

Skänneri kasutamiseks iPad Pro'l kasutati rakendust nimega 3d Scanner App. Rakendus võimaldab luua 3D kujutluse siseruumist. Kasutamisel on ekraanil tumedalt märgitud alad, mille kohta puudub informatsioon. Rakendus kasutab informatsiooni LiDAR tehnoloogialt, ruumi mõõtmete jaoks ning samal ajal salvestab pildi lõikudena objektide värvi ja üleüldise mustri. Hiljem pannakse kokku pildi lõigud ning ruumi mõõtmed, et luua ruumist värviline 3D objekt.

1.3 Three.js

Three.js on JavaScripti teek, mis lihtsustab WebGL kasutust veebis. WebGL kasutatakse 3D objektide animeerimiseks ja loomiseks veebikeskkonnas.

1.3.1 WebGL

WebGL on JavaScripti rakendusliides, mis võimaldab veebibrauserites kasutada 3D ja 2D graafikat, ilma pluginiteta brauserites, millel on selle jaoks olemas tugi. Rakendusliides on laialt toetatud, sest see tugineb HTML5 `<canvas>` elementidele, mis on kõikides HTML5 toetavates veebilehitsejates [6]. Kasutades ära olemasolevaid HTML5 elemente, viiakse üle renderdamine kasutaja masinale, mis võimaldab kiiret ja sujuvat kasutajakogemust. Enne WebGL loomist olid veebibrauserid ainult 2D kujul, erandiks olid brauserid, kus kasutati pluginaid nagu Adobe Flash.

1.3.2 OpenGL ES

OpenGL on sarnane WebGL'ile, sest mõlema eesmärk on tuua tänapäevane graafika tavakasutajate seadmetesse, kuid kasutavad erinevat platvormi. Nimelt WebGL on veebipõhine, aga OpenGL on laiemalt kasutuses. Rakendusliides on saadaval mitmel erineval platvormil ja erinevates programmeerimise keeltes.

1.3.3 Three.js kasutus

Peamine põhjus three.js kasutuseks on WebGL komplekssus. Ekraanile 3D kuubiku saamiseks kulub Three.js teeki kasutades 25 rida koodi. Selleks peab ainult panema paika kuubiku ja selle vastava materjali, ümbritseva keskkonna, kaamera, stseeni ja lõpuks kõik animeerima. WebGL'is sarnase asja saavutamiseks on vaja näha rohkem vaeva. Näiteks kuubiku loomiseks ei piisa üherealisest käsust, vaid on vaja defineerida kuubiku iga nähtava pinnase konstantsed mõõtmed. Three.js lihtsustab koodi ja teeb keerukamate probleemide käsitlemise kergemaks.

Kõik seadmed, mis toetavad WebGL'i toetavad ka Three.js'i. Täpsemalt kui veebibrauser toetab `<canvas>` elemente, siis Three.js on seal kasutatav. Tähtis on ka teada, et kõik brauserid ei toeta `<canvas>` elemente. Näiteks Opera Mini, vanemad Androidi brauseri versioonid või Internet Exploreri aegunud variandid [7].

1.4 Blender

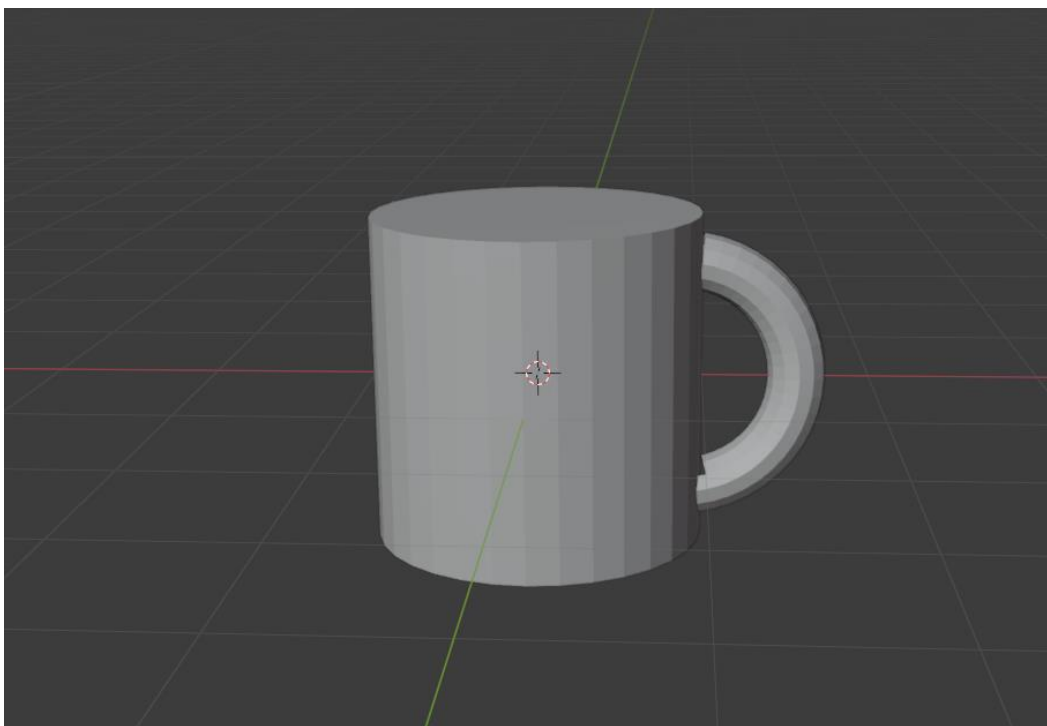
Blender on tasuta avatud lähtekoodiga 3D objektide käsituskomplekt. Programm toetab objektide mudeldamist, animatsiooni, renderdamist, kompositsiooni ja muud.

Blender on kasutatav kõikidel operatsioonisüsteemidel ja töötab väga hästi, sest kasutab OpenGL'i objektide renderdamisel. Avatud litsentsiga tarkvara võimaldab loodut kaubanduslikult kasutada ilma otsese loata.

1.4.1 Modelleerimine ja skulptimine

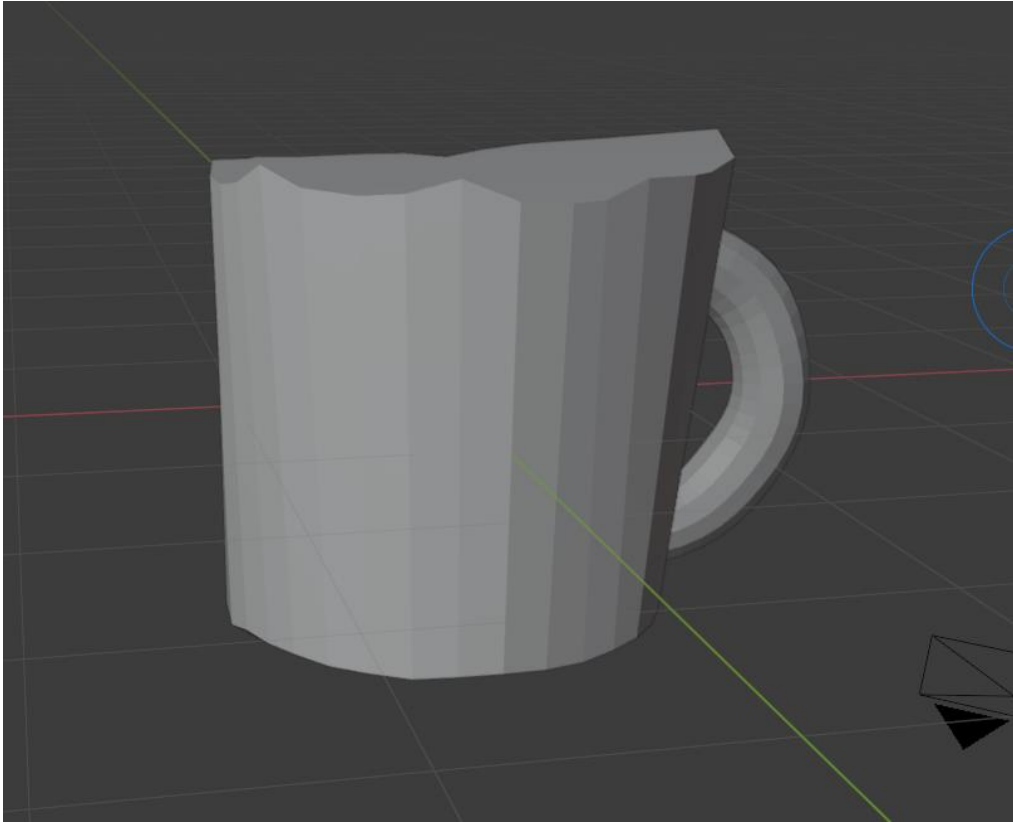
Blender võimaldab 3D objekte muuta, mõjutades pinnaste asukohta, struktuuri, sügavust ja nurkade arvu. Objektid sisaldavad nurki, pinnaseid ja külgi, mida on võimalik individuaalselt mõjutada. Igale objektile saab lisada juurde nurki ja külgi ning neid venitada vastavalt vajadusele [8].

Modelleerides alustatakse primitiivse objektiga, mis näeb sarnane välja lõpptootele. Blenderis on nendeks objektideks: kuup, silinder, pinnas, ring, koonus, rõngas, kera ja ahv [13]. Näiteks standartne kohvikruus koosneb seest tühjast ja pealt lahtisest silindrist ning poolikust rõngast (Joonis 2). Luues töötlemata objektidest sarnase kuju, saab ümber kujundada ning muuta primitiivseid elemente sobivaks vastavalt vajadusele, lisades detaile.



Joonis 2. Primitiivsetest objektidest koosnev kruus

Skulptides on saab muuta loodud objekti kuju, lisades detaile üldisele kujule. Oletame, et antud kruus on pealt ebamäärane, sest tegemist on käsitööga (Joonis 3). Vastavat detaili saame lisada kasutades paisutamise tööriista.



Joonis 3. Ebamäärase pinnasega kruus

2. 3D objektide käsitus Three.js'is

Lõputöö kontekstis on vaja mõista, kuidas objektide loomine ja nende käsitlemine töötab, sest võrreldes traditsiooniliste mängumootoritega pole Three.js selleks loodud.

2.1 Scene

Stseen on Three.js teegis konstant, mis hoiab endas objekte, valgust ja kaamerat. Hea külg on stseenide puhul see, et mängu kontekstis piisab lihtsalt stseeni konstandi vahetamisest, et olla teises ruumis või mängus.

2.2 Objektid

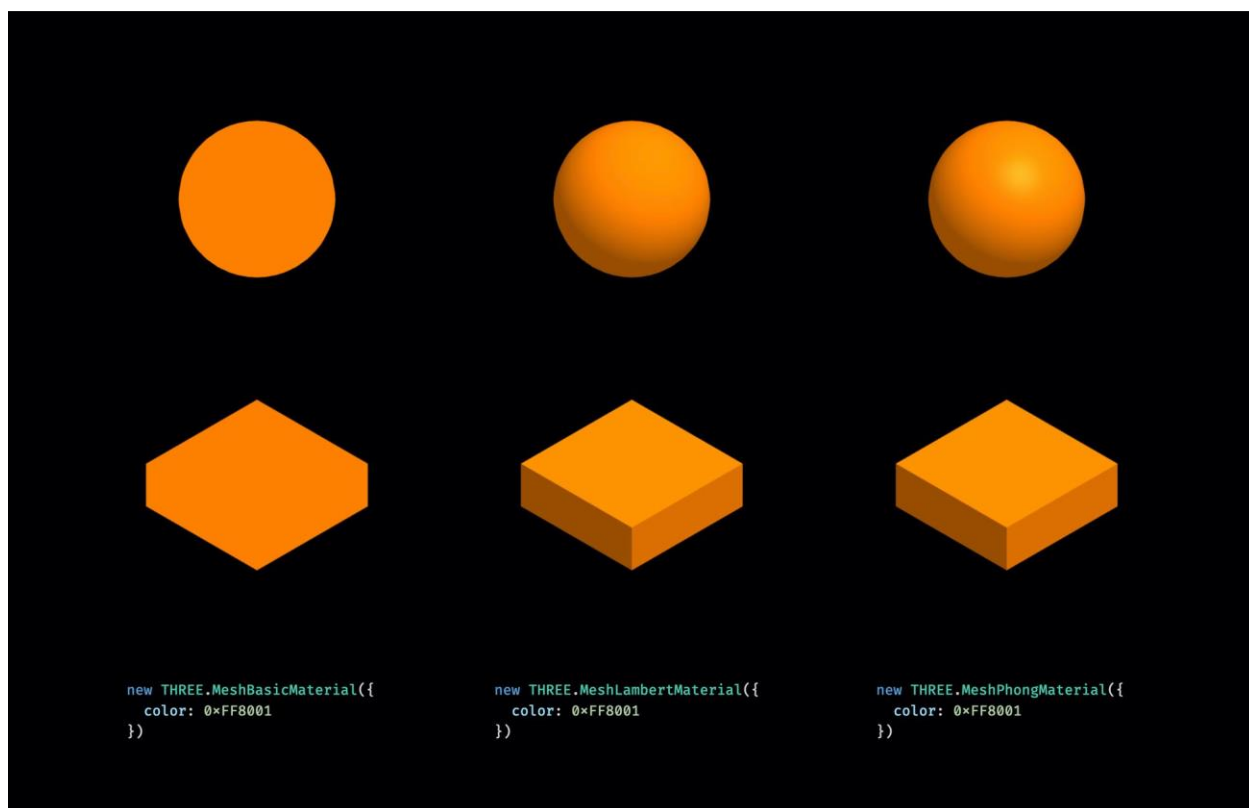
Stseen sisaldab objekte ja objektidel on samuti enda parameetrid. Igal esemel ekraanil on geomeetria (pikkus, laius, sügavus), materjal (värv või tekstuur või muu) ja asukoht (x -, y -, z -koordinaatteljestik).

Geomeetria on kujund ja igal kujundil on vastavad mõõtmed. Kujundeid on erinevaid. Three.js poolt on defineeritud ära kuus erinevat tüüpi: pinnas, kast, silinder, ikosaeeder, koonus ja kera. Kõikidest eelnevalt mainitud kujunditest saab luua keerukamaid objekte põimides neid teiste kujunditega või mõjutades nende mõõtmeid.

Materjalid kirjeldavad kujundi nähtavat pinnast. Igal materjalil on tekstuur, värv ja läbipaistvus. Materjalidega töötamine on vajalik, sest iga pinnas reageerib keskkonnale erinevalt. Three.js sisaldab kolme peamist materjalitüüpi. MeshBasicMaterial kasutades värvitakse kõik kujundi küljed ühte värvi, aga keskkond (näiteks valguse nurk) ei mõjuta selle välimust. MeshLambertMaterial on sarnane MeshBasicMaterialile, aga selle pinnaste värvus muutub vastavalt valgusele. MeshPhongMaterial käsitleb igat individuaalset pikslit pinnasel, et anda võimalikult reaalne reageerimine valgusele (Joonis 4). Ehk kui MeshLambertMaterial valguse puhul värvib terve pinnase ühte värvi, siis MeshPhongMaterial puhul on näha valguse tugevusest mõjutatud värvuse muutust.

Asukoht näitab, kus objekt asub meil ruumis kasutades x -, y -, z -koordinaatteljestikku. Animeerides liikumist muudetakse ainult asukoha parameetreid. Lisaks asukohale saab määrata ka objektile kaldenurka. Nurkade lisamine toimub radiaanides ja põhineb samuti x -, y -, z -

koordinaatteljestikule. Kõik keeramised toimuvad ümber telje, millele on antud parameetriks radiaanides nurk.



Joonis 4. Erinevad Three.js objektide materjalid[14]

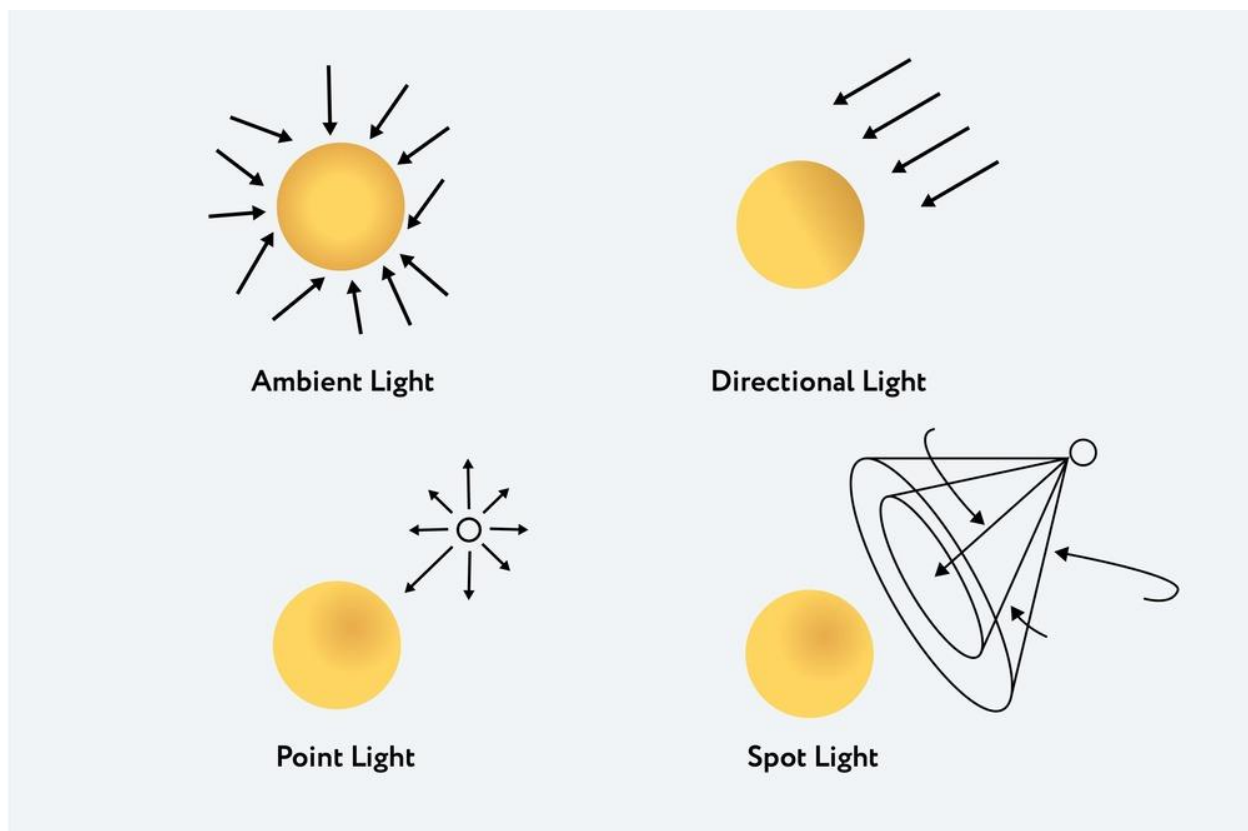
2.3 Valgus

Valguse kasutus oleneb, mis tüüpi materjali on kasutatud luues objekte. Ruumis ei ole vaja kasutada valgust, kui tegemist on MeshBasicMaterial materjaliga, sest valgus ei mõjuta selle välimust. Teised kaks materjali tüüpi vajavad valgust, et olla nähtav. Valgust on kaks peamist tüüpi: ümbritsev ehk *ambient* ja suunatud ehk *directional* (Joonis 5). Neid kahte valgustüüpi kombineerides saab saavutada väga realistliku keskkonna. Looduses on ainult suunatud valgused, mis peegelduvad erinevatelt pinnastelt ja loovad nähtava ruumi. Valguse protsessi kulu vähendamiseks on meil ümbritsev valgus, mis vähendab suunatud valguste arvu ruumis.

Three.js puhul on valgusel kaks kuni kolm parameetrit. Esimene parameeter on valguse intensiivsus. Teine parameeter määrab valguse värvi. Kolmas parameeter, mis on ainult suunatud valgusel, on suund. Three.js käsitleb seda kui mitmeid paralleelseid valguskiiri, mis tulevad ühe

kindla nurga alt. Nurga olemasolu annab objekti pinnastele erinevad toonid ningmõnda külge mõjutab valgus rohkem kui teist.

Teegis on veel kaks valgustüüpi: punktvalgus, mis suunab valgust kindlas punktis kõikides suundades ja prožektorvalgus, mis sarnaneb punktvalgusele, aga tal on kindel koonusekujuline ala, mida valgustatakse (Joonis 5).



Joonis 5. Erinevad valguse tüübid Three.js teegis [16]

2.4 Liikumine ja suund

Javascripti teek Three.js on eelkõige loodud liikuvate 3D objektide näitamiseks ekraanil, kasutaja sisestus tuleb kõik ise luua. WASD nuppe vajutades saab ruumis ringi liikuda. See toimub nii, et mängija objektiks on kaamera, mis on konstantselt ühel kindlal kõrgusel maapinnast. Mängu siseselt on ka gravitatsioon, mis koguaeg kukutab kaamerat maapinnale. Hiirega saab muuta kaamera suunda ja klaviatuuriga asukohta.

2.5 Octree

Mainimata ei saa jätta ühte peamist objektide vahel eristamiseks kasutatud Three.js osa, milleks on octreed. Need on puud, mille igal lehel on kaheksa last, see võimaldab hoida erinevate objektide andmeid, sest igas ruumis on need erinevad. Iga objekt ruumis on oma tingimustega, sellepärast kasutatakse puid, mis sisaldavad objekti struktuurilisi andmeid. Uuemates Three.js versioonides puudub objekt octree, mistõttu selle töö tegemisel on tähtis mõista selle sisu. Kõik varasemad materjalid sellega seoses on ametlikust Three.js dokumentatsioonist eemaldatud.

Octree kirjeldamiseks piisab 3D objekti lahti mõtestamisest. Ruudul on nelu nurka ja see on 2D objekt, kuubil on kaheksa nurka ja see on 3D objekt. Octree on puu, mis sisaldab informatsiooni iga nurga kohta ehk üks kuup koosneb kaheksast nurgast. Kuna octree on puu, millele saame lisada lehtedele veel elemente, siis on võimalik luua näiteks kuubik, mis koosneb kaheksast kuubikust, mille iga element koosneb kaheksast nurgast. Selliste puuga on salvestatav kõik vajalik informatsioon mingisuguse ekraanil oleva 3D elemendi kohta.

2.6 Ruumiliste objektide sisse laadimine

Töö raames tuleb laadida erinevad Delta õppehoone ruumid mängudesse sisse. Failitüüpe, mida kasutatakse initsialiseerides on palju, sellepärast on tähtis teada, millist eelistada ja kuidas neid käsitleda.

2.6.1 OBJLoader

Kõige laialdasemalt töötav lahendus on OBJLoader. See laeb sisse *.obj* faililaiendiga ruumilisi objekte [15]. Toetatud ES5 ja ES6 veebilehitsejate poolt, aga suure failisuuruse tõttu ei ole see kõige optimaalsem ruumikasutuse puhul.

Antud töö puhul on tähtis, et veebipõhiselt saaks rakendust kasutada igäüks. Veebilehitsejas suur 3D fail laetakse arvutimällu ja arvutid, millel on 2GB mälu, ei pruugi mahutada suurt ruumilist objekti koos muude mänguga seotud andmetega.

2.6.2 GLTFLoader

GLTFLoader toetab kahte erinevat failiformaati *.gltf* ja *.glb*. GLTF failiformaat sisaldab JSON-i kujul andmeid 3D objekti kohta ja GLB hoiab binaarkujul informatsiooni. GLB on natukene

väiksem kui GLTF, kuid peamine ruumi kokkuhoid tuleb pigem tekstuuride muutmisest. Nimelt GLTF fail ei sisalda pindade värve, see laetakse eraldi sisse.

2.6.3 DRACOLoader

Dracoloader on GLTFLoaderil põhinev laadija, mis laeb *.drc* faile. Selle failiformaadi eesmärk on saavutada võimalikult väikese failisuurusega 3D objekt läbi kokkupakkimise. Draco on C++ põhine, aga selle dekodeerija eksisteerib ka javascripti jaoks. Seda objekti kasutades saab võimalikult kiire laadimisaja ja mälukasutus on mitmekordselt väiksem.[9]

3. Implementatsioon

Mäng koosneb mitmest informaatikat õppides käsitletavatest ülesannetest. Antud ülesanded tuleb täita, et liikuda järgmisse ruumi ning seal sooritada viimane ülesanne. Ülesanded käsitlevad andmeturvet, operatsioonisüsteeme, veebiarendust, algoritme ja arvuti arhitektuure.

3.1 Mängu ülevaade

Mäng algab videoklipiga, mis tutvustab üleüldist lugu. Tumedas riietuses mees annab peategelasele USB mälupulga ja väikse krüpteeritud kirja. Krüpteeritud kirjas on öeldud, et tumedalt riides mehel ja ta kaaslastel on vaja eksamivastuseid. Selleks tuleb viia mälupulk arvutimuuseumi öösel, siis kui Delta on tühi. Öösel hiilib peategelane Delta õppehoonesse ning näeb kaugemas koridoris eredat valgust. Sisenedes erksalt valgustatud ruumi, sulgub seljataga olev uks. Olles suletud kinnises ruumis astub peategelane virtuaalreaalsusesse.

Eesmärk esimeses ruumis on leida ruumis olevad kolm ülesannet ja need lahendada. Ülesannete lahendamise järjekord pole oluline kuid kõik ülesanded peavad olema edukalt lõpetatud, et liikuda edasi muuseumisse. Ruumis käsitletakse veebiturvalisuse, algoritmide, Linuxi ja arvuti arhitektuuri ülesandeid. Kõik ülesanded tuginevad Noored Koodi õppematerjalidel.

Kolme ülesande lahendamisel transporditakse mängija arvutimuuseumisse, kuid peab ta arvama 4-numbrilise uksekoodi. Parooliks on '0000'. Jõudes arvutimuuseumi on eesmärgiks leida õppejõu arvuti ja sinna viirus üles laadida. Viiruse üleslaadimine toimub Linux terminalis, kus käivitatakse viiruseprogramm.

Korrektse üleslaadimise korral näeb kasutaja viimast videoklippi, kus mängija põgeneb Delta õppehoonest. Videoklipi lõppedes avaneb vorm, kus küsitakse informatsiooni õpilase kohta, kes on ülesande lahendamisega hakkama saanud.

3.2 Linuxi süsteemihaldus

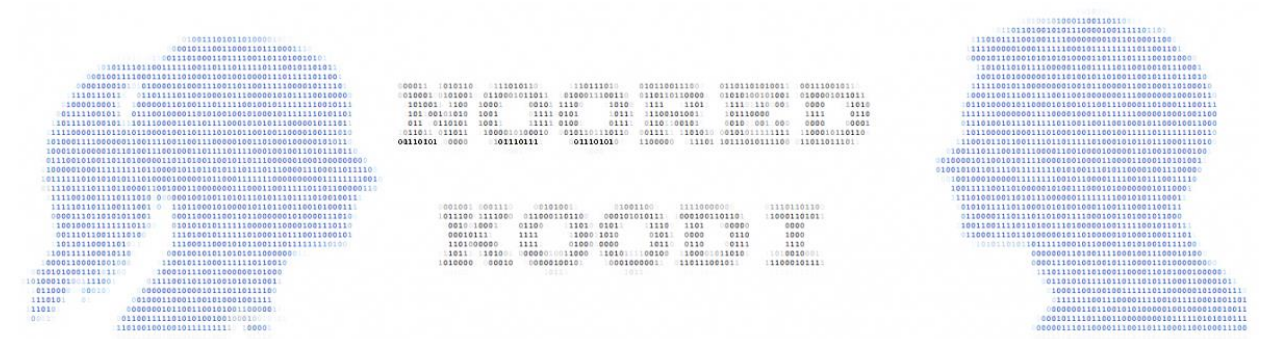
Kõige mahukam ja keerulisem ülesanne on Linuxi süsteemihaldus, kus on põimitud mitu teemat üheks keeruliseks ülesandeks.

Ülesande esimene osa põhineb Magnar Varese loodud eestikeelse õppematerjali "Linux ja Raspberry Pi algkursus" põhjal. Kasutaja liigub Linux kaustade vahel, otsides faili, mis sisaldab

informatsiooni järgmise ülesande kohta. Loodud terminali kloonis on võimalik liikuda ainult kasutades relatiivset teed. Algselt ülesanne koosnes absoluutse tee kasutusest, kuid testides oli antud element liiga keeruline. Avades otsitava faili, kasutades cat käsku, tuleb ekraanile tekst, mis kirjeldab, kuidas edasi antud ülesandega tegeleda.

Failist saadud kirjeldus suunab kasutaja *nooredkoodi.ut.ee* leheküljele², kus tuleb inspekteerida bännerit (Joonis 6). Inspekteerides on näha teksti, mis viitab pildile. Noored Koodi logo on kirjutatud kasutades binaarkoodi, mõned numbrid on tumedamad kui teised. Need numbrid tuleb binaarkoodi tõlkijasse sisestada ja tulemuseks saab parooli, millega on võimalik lõpetada ülesanne.

Lõpetamisel on viimaseks ülesandeks jällegi Linuxi süsteemi käsitlemine. Nimelt tuleb õpetuse järgi navigeerida kausta, kus asub sisestatud USB mälu pulk ning aktiveerida viirus.



Joonis 6. Peidetud sõnumiga Noored Koodi bänner [1]

3.2.1 Pwn Adventure 3: Pwnie island

Ülesanne on inspiratsiooni saanud mängust Pwn Adventure. Mäng koosneb mitmest sarjast, aga üleüldse on selle eesmärk uurida ning muuta lähtekoodi. Selle eesmärk on mänguarendajatele õpetada hea koodi tava ja kuidas hoiduda potentsiaalsetest vea ära kasutamisest. Sarnaselt sellele mängule on ka selle töö raames loodud väike näidis sellest, mis võib leida uurides mõne lehekülje lähtekoodi. [10]

Mäng avalikustati CTF (capture the flag) üritusel, kus mitmed kogenud küberturvet spetsialistid kogunevad, et võistelda häkkimises. Ghost in the Shellcode võistluse, mis toimub häkkerite

² <http://nooredkoodi.ut.ee/>

konverentsi ShmooCon ajal, avaldati mäng Pwn Adventure. Võistlejad said osa võtta veebipõhiselt kui ka kohapeal. Osalejate edusamme käsitleti otseülekande teel ürituselt. [11]

3.3 Hanoi torn

Hanoi torn on matemaatiline puzzle, mis koosneb kolmest erinevast platvormist ja silindrid, mis on üksteise peal järjestuses vastavalt diameetrile suuremast väiksemani (Joonis 7). Puzzle eesmärk on liigutada kõik kettad viimasele platvormile jälgides kindlaid reegleid:

1. kettaid tohib liigutada ühekaupa;
2. liigutatav ketas tohib ainult olla torni pealmine ketas;
3. suuremat ketast ei tohi kunagi väiksema ketta peale panna.

Ülesanne on lahendatav algoritmiliselt kui ka katse eksitus meetodiga. Hanoi torni on võimalik lahendada mitme erineva algoritmiga.

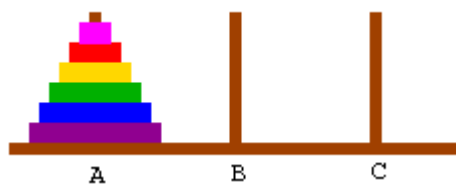
3.3.1 Hanoi torni läbimine itereerides

Iteratsiooni kasutades on Hanoi torni läbimine väga lihtne. Algoritm tuleb Peter Buenemanilt ja Leon Levylt, kes leidsid, et seni kuni meil on kettaid vasakul või keskel, on neid võimalik liigutada ühe võrra paremale. Olukorras, kus meil ei ole enam võimalik ketast paremale liigutada, liigutame paremal oleva ketta vasakule ja jätkame algset tsüklit. [12]

3.3.2 Hanoi torni läbimine rekursiivselt

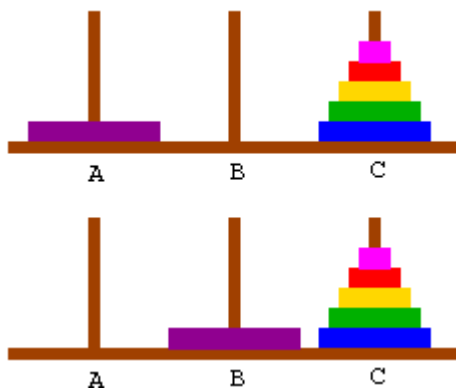
Rekursiivne lähenemine probleemile on keeruline mõista, kuid lihtne kirjutada. Rekursiivset lahendust saab mõelda ka puuna. Lahendades ülesannet peab olema kindel, millisele platvormile me tahame antud kettaid liigutada. Edaspidiselt toimub algoritm järgnevalt:

1. Liigutame $n-1$, kus n on ketaste arv hanoi tornis, ketast platvormilt a platvormile c kasutades platvormi b ajutise hoidisena. Liikumine toimub rekursiivset funktsiooni kasutades.



Joonis 7. Hanoi torni algne seis

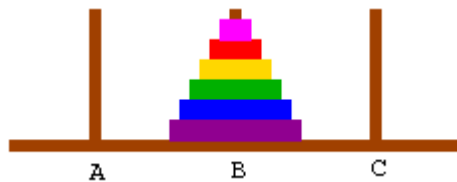
2. Kui meil on kõik pealmised kettad liigutatud ajutiselt platvormile c, siis saame liigutada kõige alumise ketta Hanoi tornist eelistatud platvormile, kus on meie eesmärk saada Hanoi torn kokku (Joonis 8)



Joonis 8. Hanoi torn peale esimest rekursiivset läbimist

3. Kordame samasugust protseduuri, mis defineeriti esimese punkti all, aga liigutades $n-2$ kettast platvormile A ning lõpuks kõige alumise ketta platvormile b.

Sellist tsüklit kordame seni kuni meil on kõik kettad platvormil b (Joonis 9). [12]



Joonis 9. Hanoi torn lõpetades.

3.4 Muud ülesanded

Eelnevalt sai käsitletud keerukamaid ülesandeid mängu siseselt. Mängu on ka lisatud lihtsamaid ülesandeid.

Pannkoogiresepti järjekorra sorteerimine on ülesanne, kus on suvalises järjekorras nimekiri pannkookide küpsetamise etappidest ja paika tuleb panna selle õige järjekord. Ülesandel on kaks võimalikku lahendust: tundes pannkoogiresepti või avades veebikonsooli ja leides sealt olevatest logidest õige järjekorra, mida arendaja unustas enne avalikustamist kustutada. Turvaauk on lisatud mõttega, et tutvustada õpilastele potentsiaalseid turvariske. Kuigi ülesanne on lahendatav algoritmiliselt, ei ole seda kasutajatelt eeldatud.

Uksekoodi ülesanne tuleneb tüüpilisest lihtsa parooliga seonduvast turvariskist. Lihtsad paroolid on väga kergesti ära arvatavad, kasutades erinevaid rünnakuid. Ülesandes on mõeldud kasutajal kasutada *brute force* meetodit, kus uksekoodiga olevasse klahvistikku toksitakse lihtsamaid paroole nagu näiteks '1234', '0000', '9876' jne. Kui iga kombinatsioon läbi proovida on võimalikke lahendeid 1000 ja see võtab 200 korda rohkem aega kui viie kõige populaarsema lahendi proovimine.

4. Testimine

Antud mängu kasutatakse Noored Koodi reklaami jaoks koolides. Kuulutatakse välja võistlus, kus lahendajate vahel loositakse välja Noored Koodi teemalisi meeneid. Tulevase võistluse tõttu on tähtis kasutajate peal mängu keerukust ja üleüldist korralikkust testida.

Katse viidi läbi 5.-6. klassi õpilaste peal. Osalejaid oli kümme ning õpilaste arvutioskused olid erineva tasemega. Õpilased alustasid ühel ajal ja jagasid enda ekraani ning akadeemilise tunni (45 minutit) lõpus paluti neil täita küsimustik mängu kohta.

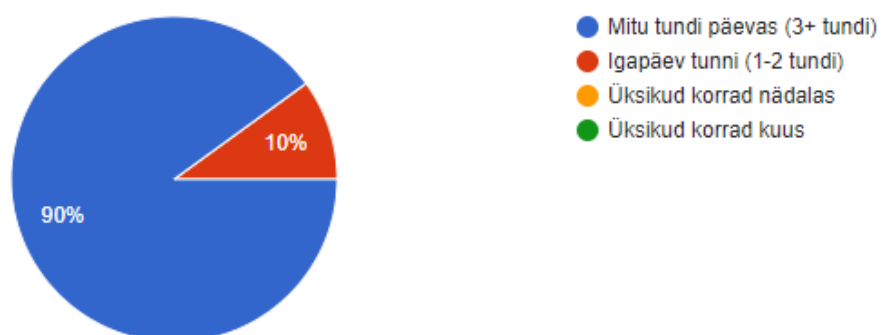
Küsimustik uuris ülesannete keerukust, kokkupuudet arvutitega, arvutimängude mängimist, üleüldist mängulist kogemust ja võimalikke soovitusi.

4.1 Tulemused

Küsimusele “Kui tihti sa arvutit kasutad?” vastati valdavas enamuses üle 3 tunni päevas (Joonis 10). Selle põhjal võib eeldada, et õpilased on päris tuttavad tehnoloogia kasutusega. Selle küsimuse tulemus on oodatav, sest küsimustiku toimumise hetkel oli õpilastel e-õpe.

Kui tihti sa arvutit kasutad?

10 vastust

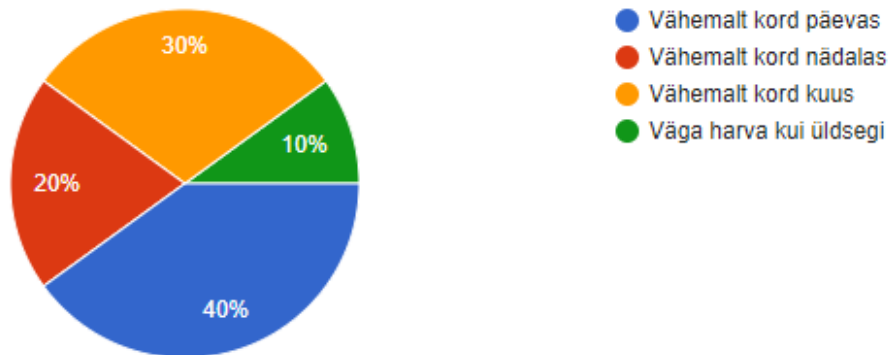


Joonis 10. Õpilaste arvutikasutuse tulemused

Küsimuse “Kui tihti sa arvutimänge mängid?” oli väga varieeruv tulemus (Joonis 11). Õpilastel on väga erinev kogemus arvutimängudega ja see vastas eeldusele, et kõik ei pruugi olla samal tasemel. Selle informatsiooni põhjal on võimalik vaadata korrelatsiooni mängu keerukuse ja kogemuse vahel.

Kui tihti sa arvutimänge mängid?

10 vastust



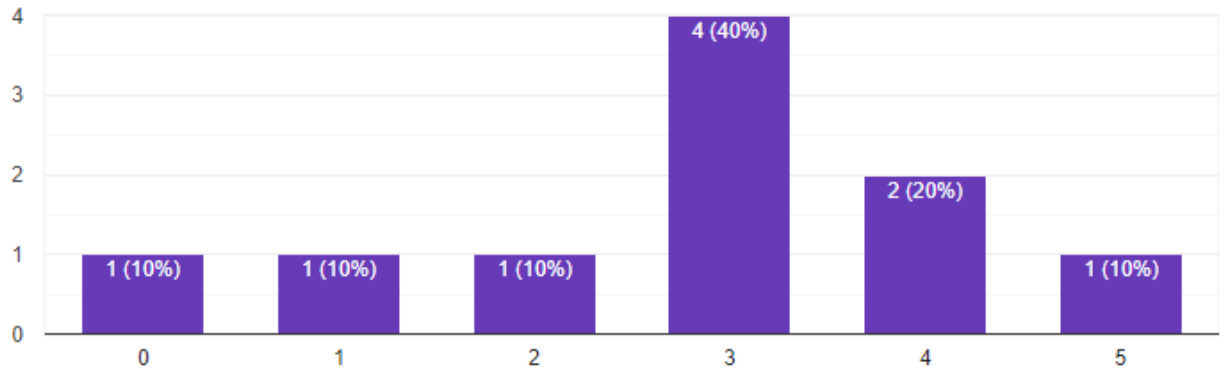
Joonis 11. Õpilaste kogemus arvutimängudega

Küsimuse “Hinda skaalal kui keerulised olid ülesanded?” tulemused olid väga mitmekülgsed (Joonis 12). Kõiki vastusevariante oli vastatud, kuid 70% õpilastest leidis, et ülesanded on pigem keerulised. Õpilastel, kes vastasid skaalal 1 või 2, on korrelatsioon arvutikasutuse ja ülesannete keerukuse vahel. Õpilased, kellel on suur arvutikasutus ja tihe arvutimängude kogemus, on tõenäolisemad ülesannetega hakkama saamisel. Erandiks on õpilane, kes märkis 0, kuna tal oli probleeme mängu laadimisega veebilehitsejas. Samuti õpilased, kellel on minimaalne kogemus

arvutimängudega, olid tõenäolisemad leidma, et ülesanded on keerulised.

Hinda skaalal kui keerulised olid ülesanded?

10 vastust



Joonis 12. Õpilaste arvamus ülesannete keerukusest

Küsimuse “Hinda skaalal enda kogemust?” tuli välja, et kasutuskogemus oli positiivne (Joonis 13).

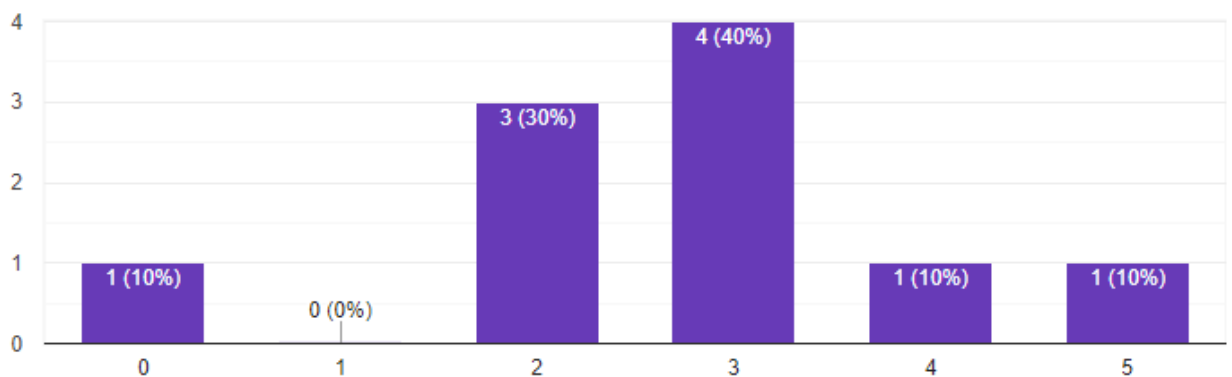
Õpilased, kes ütlesid, et ülesanded on mõõdukalt keerulised, on leidnud, et kogemus oli positiivne.

Õpilased, kes leidsid, et ülesanded olid väga keerulised, hindasid kogemust pigem negatiivselt.

Erandiks on õpilane, kes ei saanud mängu enda veebilehitsejas kasutada.

Hinda skaalal enda kogemust?

10 vastust



Joonis 13. Õpilaste kogemuse meeleolu

Küsimustiku viimane osa koosnes kolmest vabas vormis lisaküsimusest. Vastuste põhjal saab järeldada, et kuigi mängu testimise ajal oli probleeme, siis üldine kogemus oli positiivne. Õpilased leidsid ka probleeme, mis autori enda testimise käigus välja ei tulnud. Peamine probleem oli ruumilise keskkonna ümbristes. Leiti, et kui liikuda seina nurkadesse ja tulistada palle, siis on võimalik joosta seintest välja. Põhjuseks oli liiga kiire liikumiskiirus, mis võimaldas liikuda läbi seinte. Üks õpilane avastas, et kui ta on väikese ekraaniga arvutis, siis ta kukub läbi põranda enne alustamist.

Tagasisidet anti ka eraldi mängueelsetele tutvustavatele videotele. Esimese videolõigu pikkus oli ülemäära pikk ja ei seletatanud täielikult mängu lugu. Üleüldse kirjeldused ei olnud piisavalt selged, mis tekitas arusaamatusi.

Kokkuvõte

Bakalaureusetöö eesmärk oli luua virtuaalse põgenemistoa laadne veebipõhine mäng, et populariseerida Noored Koodi eesmärki tutvustada õpilastele kõrgkooli IKT erialasid. Tööd kasutatakse edaspidiselt tulevasel võistlusel.

Veebirakendus valmis 2021 aasta kevadel. Töö raames kasutati mitmeid erinevaid tehnoloogiaid, mida süvitsi analüüsiti. Loodi ülesanded, mis toetavad olemasolevatele kättesaadavatele materjalidele Noored Koodi programmist ning käsitleti võimalikke lahendusi. Programmi testis kümnest õpilasest koosnev grupp, kelle tagasiside põhjal tehti rakendusele lisaarendusi. Tagasisidet analüüsiti ning rakendusele tehti lisaarendused, vastavalt soovitudele.

Tagasisidest saab järeldada, et valminud rakendus tekitab huvi kõrgkoolides IKT erialade õppimise vastu ning seega sai töö eesmärk täidetud. Rakendust saab kasutada nii iseseisvalt kui ka arvutitundides. Veebipõhise mängu läbinud õpilane omab teadmisi turvalisusest, veebilehtedest, algoritmidest ja Linuxist. Valminud rakendus on kasutatav iseseisvalt kui ka arvutitunni teemadega kooskõlas oleva kõrvalpõikena. Samuti võib eeldada, et veebipõhise mängu lõpuni mänginud õpilane omab lõpetamisest kogemust turvalisusest, veebilehtedest, algoritmidest ja linuxist.

Valminud töö saab tulevikus võistlusena kasutusse võtta ning töö käigus valminud skanneeringud arvuti muuseumist ja virtuaalreaalsuse laborist saab võtta aluseks muude sarnaste projektide loomiseks. Autori tööd on võimalik kasutada ka alusena muude sarnaste 3D ruumide esitlemisena virtuaalselt. Töö käigus valminud ülesanded on kasutatavad ka vastava kursuse tutvustamiseks koolidele.

Viidatud kirjandus

- [1] Noored Koodi programmi koduleht <http://nooredkoodi.ut.ee/> (08.03.2021)
- [2] Rille K. Digipöördeprogramm https://www.hm.ee/sites/default/files/1_digiprogr_2020_23.pdf (08.03.2021)
- [3] Apple pressiteade Apple unveils new iPad Pro with breakthrough LiDAR Scanner and brings trackpad support to iPadOS <https://www.apple.com/newsroom/2020/03/apple-unveils-new-ipad-pro-with-lidar-scanner-and-trackpad-support-in-ipados/> (16.04.2021)
- [5] Blogipositus Why Apple's LiDAR Scanner Opens Up a Brave New World of Indoor Mapping <https://blog.magicplan.app/why-apples-lidar-scanner-opens-up-a-brave-new-world-of-indoor-mapping> (18.04.2021)
- [6] WebGL: 2D and 3D graphics for the web https://developer.mozilla.org/en-US/docs/Web/API/WebGL_API (18.04.2021)
- [7] Web 3D graphics using three js <https://humaan.com/blog/web-3d-graphics-using-three-js/> (02.08.2021)
- [8] Blender simply explained <https://all3dp.com/2/blender-simply-explained/> (02.08.2021)
- [9] Draco 3D andmete pakkimine <https://github.com/google/draco> (02.08.2021)
- [10] Pwn Adventure 3: Pwnie Island <https://charonv.com/pwn-adventure/> (07.08.2021)
- [11] Ghost in the Shellcode <http://ghostintheshellcode.com/> (07.08.2021)
- [12] Bueneman, P., & Levy, L. (1980). The Towers of Hanoi problem. *Information processing Letters*, 10(4-5), pp 243-244 (07.08.2021)
- [13] Primitiivsed blenderi objektid <https://docs.blender.org/manual/en/latest/modeling/meshes/primitives.html> (08.08.2021)
- [14] Erinevaid materjale kirjeldav pilt <https://www.freecodecamp.org/news/content/images/2021/02/freecodecamp-4.002.jpeg> (08.08.2021)

[15] OBJLoader dokumentatsioon <https://threejs.org/docs/#examples/en/loaders/OBJLoader>
(08.08.2021)

[16] Pilt valgustüüpidest Three.js'is <https://www.intexsoft.com/blog/post/three-js.html>
(08.08.2021)

Lisad

I. Valminud veebirakendus

Valminud veebirakendus on saadaval aadressil:

<http://nooredkoodi.ee/>

II. Mänguga seonduv video materjal

Koostöös juhendaja Madis Vasseri ja bakalaureuse tudengi Carmen Akkiga valminud videomaterjal, mida on kasutatud rakenduses:

1. Sissejuhatav videoklipp mängu - https://tartuulikool-my.sharepoint.com/:v:/g/personal/carmenak_ut_ee/ESqwdwdMNyxBqFMvYa7-8acBUKaz3FGT7oH3aHj2afQklg?e=M8Qozp
2. Mängu lõpetav videoklipp - https://tartuulikool-my.sharepoint.com/:v:/g/personal/carmenak_ut_ee/ESG3QrCx9qNEqPEj7EBUspMBoCqag1oA2h0xca5giIabYw?e=xM6gbq

III. Koosolekutel käsitletud materjal

Story

Läksid Delta peal uitama ja sattusid ühte põnevasse ruumi. Äkitselt käib su selja taga pauk ja kui ennast ümber pöörad, siis näed, et uks su selja taga on lukus. Proovid linki, taod natuke uksele, aga keegi ei tule, midagi ei aita. Oled lõksus. Ega midagi pole teha, pead oma nutikuse proovile panema ja hakkama otsima toast vihjeid, et ruumist välja pääseda.

Ruum: Delta arvutimuuseum ja CGVR labor

1. Algoritmika

- a. hanoi torn
- b. plokkiskeem - lihtsa pannkoogi retsept tuleb õigesse järjekorda panna, siis plokkide tagant saad mingi numbrikombinatsiooni, mis avab midagi uut

2. Linuxis mingi käsk

- a. (Martin arvas, et mõtleb ise midagi)

3. Turvalisus

- a. parooliräsist parool kätte saada - Carmen mõtleb, kuidas täpsemalt

4. ~~Arduino~~

- a. ~~elektroonikaskeem, kas Alo pidi ise tegema?~~

5. Veebilehed

- a. näeb html-i mingit väljundit, saab css-i, ja peab otsima, mis on määratud "hide" vms, ja selle nähtavaks tegema, saab sealt siis mingi vihje/vastuse
 - i. või on peidetud nt taustaga sama värvi vms ja peab värvi muutma
- b. inspectiga muuta kusagil teksti, a la "ma ei tea, mis on parool" → "ma tean, mis on parool"

6. ~~VR~~

- a. ~~????????? põgenemistuba ongi juba ise midagi sinna kanti~~

Koosolek 19.02.2021

1. Hiir või kursor võiks olla olemas. Küsimus kunstilises lahendamises, mida teeb üks ja teine.
2. Vaja teha video DELTA peauksest demoruumi (thriller) stiilis. Video pikkus 1,5-2 minutit ning annab sisendi, miks kasutaja demoruumi sattus ja miks uks lukku läks - Carmen ja Maarika
3. Storyline vaja täpselt üles kirjutada - Carmen ja Maarika
<http://www.lizengland.com/blog/2014/04/the-door-problem/>
4. Martin teeb demoruumi ülesvõtte ning teeb esimese ülesande sisse

Plot

Miks sattus deltasse?

Kasutaja on salaluureagent. Tal on ülesandeks toimetada Deltasse mälu-pulk, kus on peal spyware, et saada teada eksamitulemused. See on tal taskus. Et aga päevasel ajal on Delta täis tudengeid ja töötajaid, saadeti ta missioonile öösel. Kuna tegemist on top secret projektiga, siis saadeti kasutaja majja öösel.

Millist teed pidi liigub demoruumi?

Kasutaja siseneb Deltasse Vene bussipeatuse poolsest uksest, vaatab ettevaatlikult vasakule, paremale, siis järsult vasakule, sest märkab seal pooleldi lahtist demoruumi ust, kust kumab sooja valgust. Kuna kasutajale oli öeldud, et maja on tühi, tekitab see kahtlust ja ta läheb vaikselt, aga kindlal sammul demoruumi poole, aeg-ajalt enda ümber vaadates.

Jõudnud ruumini, vaatab ta ukse taha, kuna seal kedagi pole, astub sammu uksest sisse. Teeb poolkikivarvul paar sammu, kui kuuleb selja tagant kiireid samme, jõuab korraks üle öla kiigates märgata tumedat inimfiguuri, kui uks tema selja taga pauguga kinni lüüakse. Ta teeb kiire sammu ukse poole, katsub närviliselt linki, aga uks on lukus. Peksab käega ust, aga tulutult.

Vaatab enda ümber ringi. Ruum on muidu tühi (??), aga laua peal on VR-prillid. Liigub aeglaselt nende poole, paneb pähe ja mäng algab. Ruumis on äkitselt näha palju asju (erinevad sülearvutid, nii lahtise kui kinnise kaanega, emaplaadid jne) ja kasutaja peab hakkama õigeid kohti otsima.

Mis toimub demoruumis?

Kokku on demoruumis 5 ülesannet, iga ülesanne annab ühe tähe sõnasse DELTA. Kasutaja näeb kogu aeg ekraani üleval paremal nurgas _ _ _ _ _ ja iga kord, kui ta mingi ülesande ära lahendab, saab ta ühe tähe õigele kohale. Iga õige lahenduse korral ilmub (= põhimõtteliselt suumib sisse) trükitäht, mis sellele ülesandele vastas (“Mõmmi ja aabitsa” täheõppimise sarnaselt), ja see täht liigub vastava _ asemele.

Ülesanne 1 - Hanoi torn - A

Demoruumi nurgas on alus, kus on kolm pulka kõrvuti, ühe pulga otsas on kuus ketast. Pulkade kõrgus on umbes inimesele põlvini, laius umbes sama palju. Kasutaja peab liigutama kõik kettad teise pulga otsa, tehes vaid [lubatud liigutusi](#).

Ülesanne 2 - Raspberry Piga midagi - L

Peab välja mõtlema, kuhu porti mis kaabel tuleb panna. RasPi 4

Ülesanne 3 - pannkoogid - T

Kusagil laua peal on kauss. Kauss on praktiliselt poolkerakujuline, musta värvi ja poolmati pinnaga. Kausis on hulk paberilipikuid. Kui kasutaja vajutab kausile või selle sisule, siis ilmub tema ette õhku üksteise alla nimekiri, korraga on näha kõik lipikud:

Munad suhkruga vahtu kloppida.

Juurde segada jahu, piim ja näpuotsaga soola.

Segada ühtlaseks.

Lisada sulavõi.

Segada tainas läbi.

Kuumutada suur pann.

Lisada pannile sobiv kogus tainast.

Küpsetada paar minutit keskmisel kuumusel, kuni pannkook on altpoolt kuldne.

Keerata pannkook ringi ja küpsetada teiselt poolt.

Nimekiri on juhuslikus järjekorras, ülevalpool on esitatud õige järjestus.

Kasutaja saab lipikuid liigutada drag-and-drop põhimõttel, nagu [siin](#) on drag&drop sorting

Ülesanne 4 - Arduino - D

Riistvaras salvestatakse tähti ascii koodis. Leia binaarkoodile vastav täht ascii tabelisse.

Ülesanne 5 - veebilehed: peidetud sõnum - E

Ühe laua peal on monitori, musta pildiga. Kui kasutaja sellele läheneb, läheb ekraan valgeks. Kui sellele klõpsatakse, ilmub kasutajale <http://nooredkoodi.ut.ee/> leht, kus on inspect tool'i kasutades vaja leida üles koht, kus on pildi jaoks kirjutatud `style="display: none"` Praeguse avalehe teksti sees on enne lauset "Noored Koodi!" vihje "Inspekteeri seda lehte, leia peidetud pilt ning tühistada peitmine."

```
▼<div class="content" ng-non-bindable>
  ▼<div class="field field-name-body field-type-text-with-summary field-label-hidden">
    ▼<div class="field-items">
      ▼<div class="field-item even">
        ▼<p>
           == $0
          </p>
          ▶<p>...</p>
          <p></p>
          ▶<div class="media media-element-container media-default" style="display: block; margin-left: auto; margin-right: auto;">...</div>
          ...
        </div>
      </div>
    </div>
  </div>
```

Kui kõik tähed on käes

Kui kõik tähed on kokku saadud, hakkavad ruumis tuled vilkuma (või ekraan normaalne - valge - normaalne - valge jne) järjest kiiremini (== alguses vahe pool sekundit, viie sekundi jooksul vilgub nii kiiresti, et on põhimõtteliselt ainult valge), kuni kõik on valge. Siis läheb ekraan kaheks sekundiks mustaks ja kasutaja satub maagiliselt arvutimuuseumisse, selg vastu ust, VR-prillid endiselt ees.

Mis toimub arvutimuuseumis?

Ülesanne 6 - turvalisus: parooli äraarvamine

Proovi arvutisse sisenemiseks levinud lühikesi paroole (12345)

Ülesanne 7 - Linux

USB mountimine (vihjetega)

Kuhu lõpuks välja jõuab?

Leiab [kõige vanema arvuti, mis seal on, mis võtab USB-pulka]. Vajutab USB pordi piirkonda ja tekib hologrammilaadne pilt kirjaga “Missioon sooritatud!” (== poolläbipaistev must kast, mis asub kasutaja silmadest natuke kõrgemal, roheline tekst peal). Kasutaja võtab prillid peast, vaatab enda ümber ringi, näeb, et uks on umbes rusika jagu lahti tulnud, kõnnib rahulikult uksest välja ja tagasi vaatamata jalutab stoilise rahuga peaukse kaudu majast välja.

FORM:

eesnimi, perenimi, vanus, kool, email

V. Litsents

Lihtlitsents lõputöö reprodutseerimiseks ja üldsusele kättesaadavaks tegemiseks

Mina, Martin Toode,

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) minu loodud teose

Virtuaalne põgenemistuba Deltas

mille juhendaja on **Madis Vasser**,

reprodutseerimiseks eesmärgiga seda säilitada, sealhulgas lisada digitaalarhiivi DSpace kuni autoriõiguse kehtivuse lõppemiseni.

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

Martin Toode

04.08.2021