

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Carl Eric Reinsberg 193983IAIB

**MUUSIKALE REAGEERIV 3D KESKKOND
VEEBIRAKENDUSELE WANDERFUL DAY**

Bakalaureusetöö

Juhendaja: Gert Kanter
Doktorikraad

Tallinn 2020

Autorideklaratsioon

Kinnitan, et olen koostanud antud lõputöö iseseisvalt ning seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on töös viidatud.

Autor: Carl Eric Reinsberg

31.03.2022

Annotatsioon

Lõputöö eesmärk on luua edasiarendatav eesliides avalikule veebilehe rakendusele, mille tagaliidese loob Rasmus Riismann. Autori ja Rasmuse tööd kokku moodustavad interaktiivse veebirakenduse prototüübi Wonderful Day.

Eesliideses on vaja luua keskkond, kus liigutakse sõidukiga lõpmatult edasi koos sõprade või teiste inimestega. Samal ajal kuulevad kasutajad muusikat ning näevad, kuidas keskkond nende ümber mööda liigub. Keskkond ei pea olema maapealne vaid võib olla midagi täiesti muud. Käesolevas lõputöös keskendutakse eelkõige elamuse aspektile ja teostatakse ainult üks võimalik stsenaarium (autosõit).

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 16 leheküljel, 8 peatükki, 12 joonist, 0 tabelit.

Abstract

Music Adaptive 3D Environment for Wonderful Day

The aim of this thesis is to create a frontend for a public web application for which the backend is created by Rasmus Riismann. The author's and Rasmus's work together form a prototype for an interactive web application called Wonderful Day.

In the frontend there must be an environment, where the user is on a vehicle with their friends or random people, which is driving on infinitely. At the same time it is possible to listen to music and see how the environment moves past you. The environment itself does not need to be of earthly context and it could be something absolutely different. In this thesis the application focuses primarily on the experience and only one environment is created (driving in a vehicle).

The thesis is in Estonian and contains 16 pages of text, 8 chapters, 12 figures, 0 tables.

Lühendite ja mõistete sõnastik

eesliides	Veebirakenduse osa, mis käivitatakse veebibrauseris
HTML	<i>HyperText Markup Language</i> , hüpertekst-märgistuskeel
Javascript, JS	Eesliidestest kasutatav programmeerimiskeel
punktmass	Kogum punktidest mida saab eraldiseisvalt või ühiselt mõjutada
tagaliides	Veebirakenduse osa, mis käivitatakse serveril
Three.js	Javascripti teek mis võimaldab 3d visuaale kuvada
URL	<i>Uniform Resource Locator</i> , internetiaadress
Wanderful.day	Lõputöö rakenduse nimi ja internetiaadress

Sisukord

Autorideklaratsioon	2
Annotatsioon.....	3
Abstract Music Adaptive 3D Environment for Wonderful Day	4
Lühendite ja mõistete sõnastik	5
Sisukord	6
Jooniste loetelu	8
1 Sissejuhatus	9
2 Analüüs.....	10
2.1 Olemasolevad lahendused	10
2.2 Tehnoloogiate valik	10
2.2.1 Veebiliidese 3D teek.....	10
3 Rakenduse nõuded.....	11
3.1 Mittefunktsionaalsed nõuded.....	11
3.2 Funktsionaalsed nõuded	11
4 Eesrakenduse komponendid	12
4.1 CreateScene().....	12
4.2 Render()	13
4.2.1 Maastiku liikumise efekti loomine itereerimise abil	13
4.2.2 Maastiku liikumise efekti loomine horisondi järgi.....	14
4.2.3 Punktmass.....	15
4.2.4 Üllatuselement	16
4.2.5 Sõiduvahend	16
4.3 Mängija.....	16
5 Arendus.....	17
5.1 Veebilehe baas	17
5.2 Maastiku realiseerimine.....	17
5.3 Sõiduvahend	18
5.4 Üllatuselement	18
5.5 Mängija tegelane.....	19
5.6 Muusika visualiseerimine	20
5.7 Liidestus tagaliideselega.....	22

5.8 Kasutajatelt saadud tagasiside	22
6 Tähelepanekud ja järeldused	24
7 Edasised arendusvõimalused ja sisu suurendamine.....	25
8 Kokkuvõte	25
Kasutatud kirjandus	26
Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks	27

Jooniste loetelu

Joonis 1. CreateScene() funktsiooni sisesed sammud 3D keskkonna initsialiseerimiseks	12
Joonis 2. Maastiku detailsus kahe lainefunktsiooniga.....	14
Joonis 3. Maastiku detailsus nelja lainefunktsiooniga.....	14
Joonis 4. Punktmass sinisena auto ümbruses.....	15
Joonis 5. Üllatuselement puu.....	19
Joonis 6. Erinevad pead ja kehad eri värvides.....	20
Joonis 7. Graaf mis näitab <i>audioAnalyserilt</i> saadud helisageduste (vasakult paremale tõusev) ärakasutamise jaotust <i>getLowMidHigh()</i> poolt.....	21
Joonis 8. Maastik ja punktmass juhul kui on vaikus	21
Joonis 9. Maastik ja punktmass juhul kui muusika on intensiivne, mäestik on teravam ja punktid on suuremad	22
Joonis 10. Tagasiside 3D liidese jõudluse kohta	23
Joonis 11. vastajate rahulolu nendel oleva jõudluse kohta.....	23
Joonis 12. Rakenduse 3D liides küsitluse esitamise ajal	24

1 Sissejuhatus

Teistega suhtlemine on inimestele oluline. Meie liigi arenemise suureks aitavaks faktoriks oli meie oskus omavahel suhelda. Teistega suhtlemine aitab kaasa aju arengus, mälu parendamises ja kaitseb aju haiguste eest. Lähedased suhtlusringkonnad teevad üleüldiselt meid rõõmsamaks ning tõstavad eluga rahulolu. Tänapäeval me tunneme heaolu olles sotsiaalsed ning kasutame internetti sotsialiseerimiseks üle kaugete distantside. Antud rakendus annab võimaluse suhelda teiste inimestega, olgu nad võõrad või lähedased, ning aitab seetõttu inimestel lõõgastuda ja areneda sotsiaalselt [1].

Käesoleva bakalaureusetöö idee tuli autori sõbralt ja kursusekaaslaselt Rasmus Riismannilt, kes pakkus välja autorile oma idee. Taheti luua rakendus mis lubab läbi interneti sõpradega või uute inimestega aega veeta, lobiseda ja kuulata muusikat, keskkonnas mis kujutab endast lõputut sõitu mille sihtpunkt ei ole oluline ning mille taust oli meeldiv. Lisaks pidi rakenduse osaliste vahel saama kuulata muusikat.

Töö eesmärk oli luua prototüüp Wonderful Day veebirakendusele, kus on loodud põhikomponendid rakenduse edasiarendamiseks ja sisu suurendamiseks lõputöö väliselt. Selles bakalaureuse töös keskendub autor rakenduse Wonderful Day eesliidese arendusele ja loodud komponentidele. Kogu arenduse käigus keskenduti rakenduse baasi loomisele ehk põhifunktsioonide arendamisele.

2 Analüüs

Eeltöök oli vaja autoril leida olemasolevaid sarnaseid lahendusi ning mis võimalused on rakenduse 3D veebiliidese loomiseks.

2.1 Olemasolevad lahendused

Otseselt sellise ideega veebirakendust autor ei leidnud. Autor leidis videomänge [2], mis annavad autosõidu rännaku kogemust aga need ei keskendu sotsiaalsele aspektile või muusika visualiseerimisele. Töö autor on küll mitu aastat kasutanud sotsiaalportaal rakendust VRChat [3], kuhu on loodud paar sarnast kogemust andvat alamkeskkonda.

2.2 Tehnoloogiate valik

2.2.1 Veebiliidese 3D teek

Autor leidis, et kõige populaarsemad 3D Javascript teegid on Babylon.js [4] ja Three.js [5]. Kuna Babylon.js oli React.js [6] poolne teek ning autoril oli idee hakata arendama Angular-iga [7] kuna see on talle rohkem tuntum siis valis ta Three.js-i. Autor leidis ka, et Babylon.js on suhteliselt noor teek, mille tõttu autori valik kaldus eemale sellest, kartses, et peab vahetama teeki keset arendust.

3 Rakenduse nõuded

Selles alapeatükis on kirjeldatud soovitud nõuded eesliidesele. Need nõuded kujunesid eeltöö käigus teise arendaja Rasmusega arutades ning neid täiendati koosolekute ja arenduse ajal.

3.1 Mittefunktsionaalsed nõuded

Rakendusele rakendati alljärgnevatid mittefunktsionaalseid nõudeid.

1. Rakendus peab jooksma keskmisel lauaarvutil keskmiselt 60 kaadrit sekundis.
2. Mudelite ja keskkonna disain peab olema lihtne või madala detailsusega [8].
3. Rakenduses ei ole sisselogimist ega kasutaja loomist. Kõik info hoitakse mälus.

3.2 Funktsionaalsed nõuded

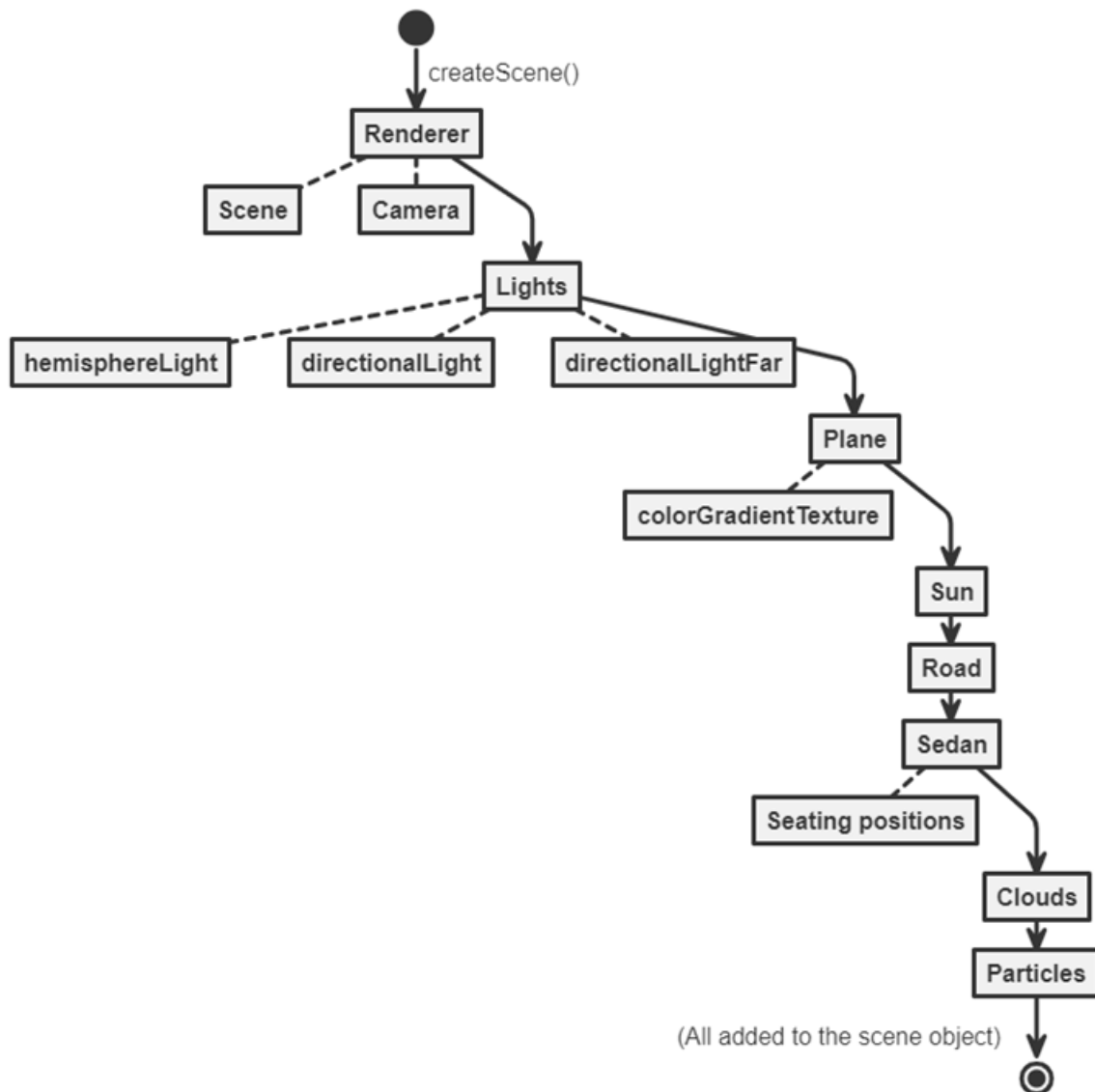
Rakendusele rakendati alljärgnevatid funktsionaalseid nõudeid.

1. Rakendus genereerib fookuses oleva sõiduvahendi ümber maapealse lähedase välimusega maastiku. Maastik liigub alati sõiduvahendi suunale vastu.
2. Rakenduses olev maastik genereeritakse reaajas.
3. Rakenduses olev keskkond reageerib samast rakendusest mängitud muusikaga.
4. Rakenduses esineb vahetevahel objekt (üllatuselement), mis toob muutust maastikul.
5. Rakenduses on põhifookus sõiduvahendil, kus sees on mängijate tegelased.
6. Mängijate tegelastel saavad olla erinevad välimused kuju ja värvi poolest.

4 Eesrakenduse komponendid

4.1 CreateScene()

CreateScene() käivitub enne renderdamise alustamist. See funktsioon initsialiseerib Three.js põhiobjektid nagu stseen, renderdaja ja kaamera. Edasi asetatakse stseeni kõik objektid mida luuakse ning soovitakse kuvada. Pärast põhiobjektide loomist loob *CreateScene()* valgustuse, maastiku, autotee, sõiduvahendi, üllatusobjekti ja punktmassi objektid. *CreateScene()* sammud on näha Joonis 1, lk 12.



Joonis 1. *CreateScene()* funktsiooni sisesed sammud 3D keskkonna initsialiseerimiseks

4.2 Render()

Render() funktsioonis küsitakse audio-player elemendilt muusika sageduse andmed mida kasutab ära *getLowMidHigh()* funktsioon mis jagab muusika sagedused kolme gruppi: madal, keskmine ja kõrge, ning ka veel madala ja keskmise positiivne muut. Neid viite muutujat saab ära kasutada muusika reageerivuse visualiseerimiseks. Järgmisena arvutatakse maastiku kõrgused.

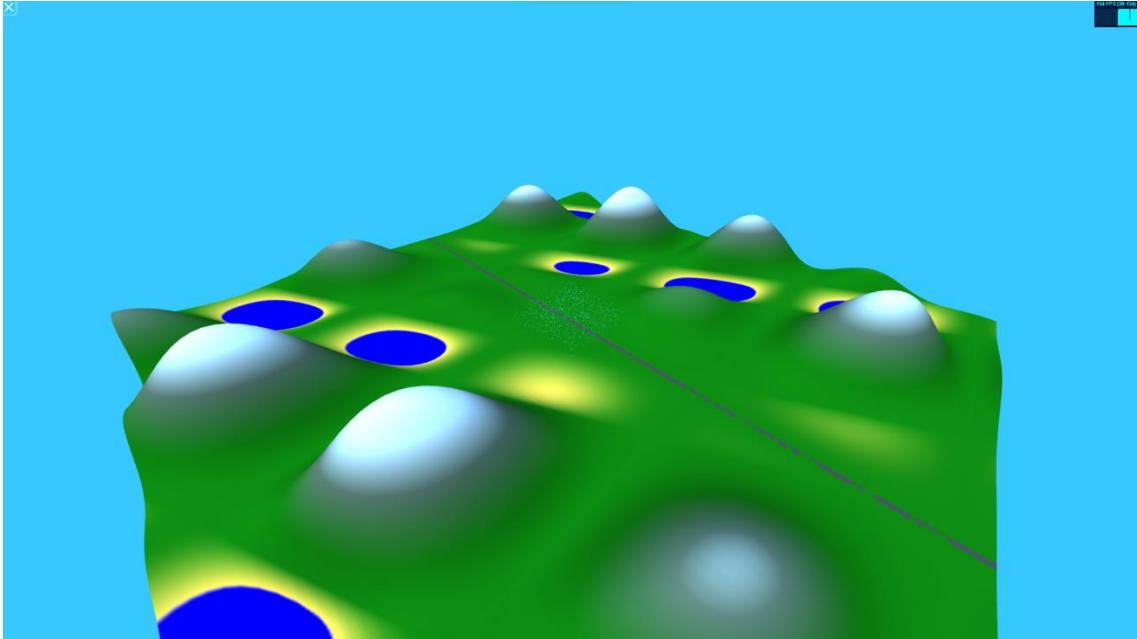
4.2.1 Maastiku liikumise efekti loomine itereerimise abil

Maastiku genereerimise põhimeetod on läbi maastiku objekti iga punkti itereerimine ning selle kõrguse arvutamine. Kõrgus arvutatakse nelja erineva laine funktsiooni abil, et luua võimalikult mürane maastik. Kaks esimest lainefunktsiooni mõjutavad maastikku rohkem ning kaks viimast toovad kujus juurde väiksemaid muutusi. Esimese kahe funktsiooni mõju on näha Joonis 2, lk 14. Kolmanda ja neljanda funktsiooni toodud väiksemaid muudatusi on näha Joonis 3, lk 14.

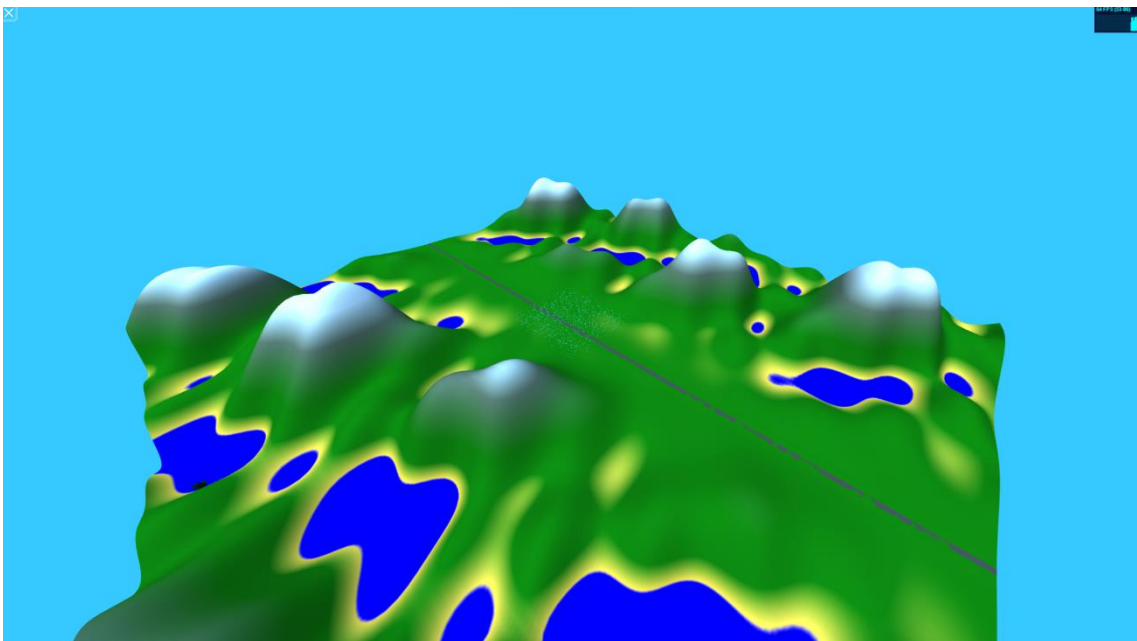
Igal lainel on funktsiooni y vähemalt mõjutatud ajaga positiivselt, seega maastik liigub ajas y telge mööda positiivselt. Funktsioonid on ka autotee alal määratud nulliks ning korrutatud kaugusest teega ruudus, ehk kõrgused kasvavad astmeliselt teest kaugemale minnes, et tekitada sujuvat üleminekut autotee juurest genereeritud maastikule.

Iga punkti kõrguse arvutamisel arvestatakse ka *getLowMidHigh()* funktsioonist saadud valitud muutujate väärtustega.

Tõstes lainete kogust neljalt kuuele saaks maastikule lisada veel rohkem detaile aga autor otsustas selle vastu kuna see aeglustas rakendust liiga palju.



Joonis 2. Maastiku detailsus kahe lainefunktsiooniga



Joonis 3. Maastiku detailsus nelja lainefunktsiooniga

4.2.2 Maastiku liikumise efekti loomine horisondi järgi

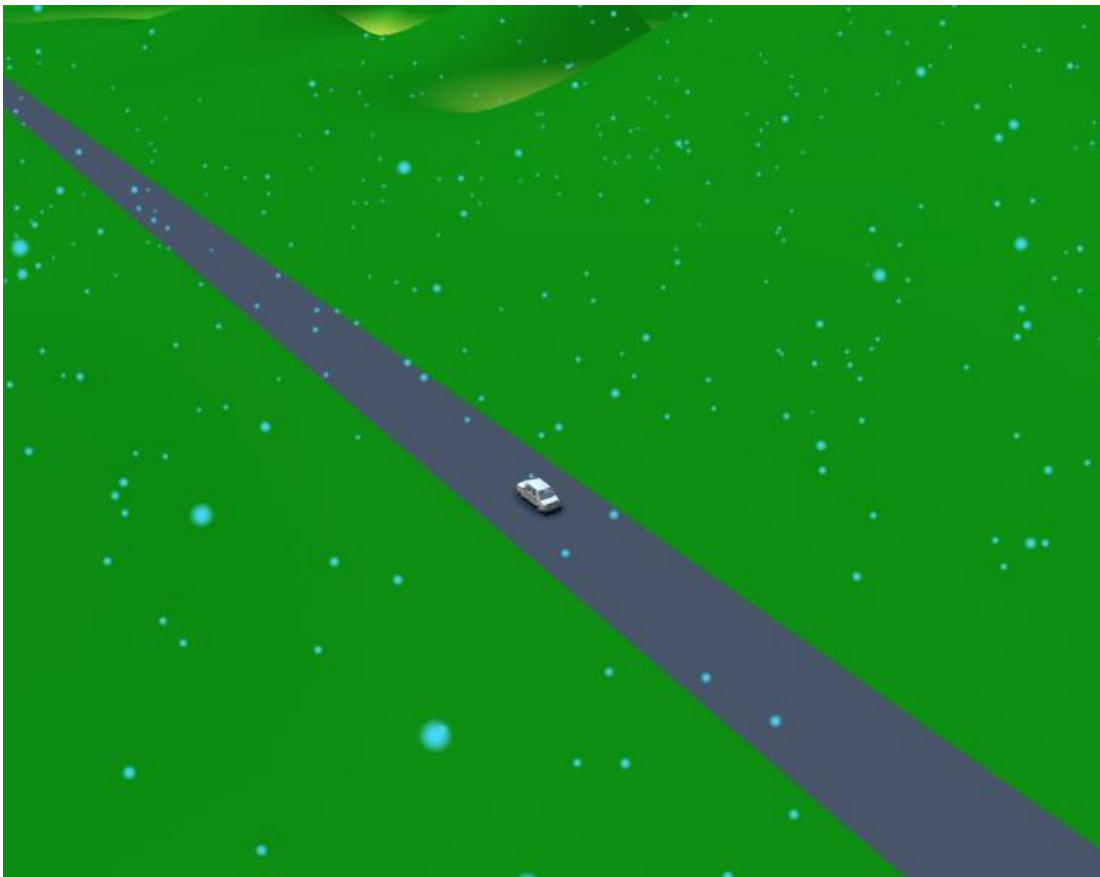
Autor proovis ka luua maastiku genereerimist horisondi järgi. See meetod on palju kiirem kuna see arvutab maastiku punktide kõrgusi ainult horisondi real, ülejäänud kõrgused hoitakse mälus. Igal kaadril liigutatakse kõrgus rea võrra edasi. Liikumine toimub sama moodi mööda y-telge aga selle meetodi üks puudus on see, et liikumine ei

ole mitte seotud ajaga vaid rakenduse kaadrite kiirusega. Seetõttu oleks seda meetodit kasutades kahe kasutaja kogemused erinevad kui nende arvuti võimsus erineb.

Teine puudus selle meetodiga on see, et kui rakendus alustab on maastik lapik ning maastik peab horisondilt liikuma sinuni, kuid itereerimismeetodiga on maastik alati olemas.

4.2.3 Punktmass

Järgmisena *Render()* funktsioonis muudetakse punktmassi punktide asukohad ja suurust. Nende asukohti mõjutab väikene baaskiirus aga põhiliselt *getLowMidHigh()* funktsioonist saadud muusika sagedused suurendavad punktide kiirust. See annab paremat rütmi tajuvust muusika kuulamisele. Kontrollitakse ka, et kui punkt liiga kaugemale liigub asetatakse ta teisele poole, muidu punktid liiguksid sõiduvahendi ümbrusest minema. Punktmassi on näidatud Joonis 4, lk 15.



Joonis 4. Punktmass sinisena auto ümbruses

4.2.4 Üllatuselement

Kuna üllatuselement peab nägema välja nii, et ta oleks maastiku peal paigal. On vaja seda liigutada ajas samas suunas mis maastik liigub. Ning peab veel arvutama ka kõrguse, sest maastik muutub kõrgemaks muusikaga. Kõrguse arvutamiseks kasutatakse ära autori loodud funktsiooni *calculateEggHeight_Iterative(x, y, now)* mis kasutab samu arvutusi, mis maastik kõrguse leidmiseks, aga ainult üllatuselemendi asukohas.

Seoses üllatuselemendiga tehakse ka kontroll, et ta liiga kaugemale pole liikunud, ning kui jah, siis luuakse ta horisondile uuesti teisse asukohta.

4.2.5 Sõiduvahend

Iga kaader keerutatakse sõiduvahendi rattaid baaskiirusega ja *getLowMidHigh()* funktsioonilt saadud kõrge muutujaga.

4.3 Mängija

Mängijate tegelased asetatakse sõiduvahendisse juhul kui ühinetakse uue ruumiga, sest siis saadakse kätte kõik ruumis olijad ning nendele genereeritakse välimus ja asetatakse järjest istmetele, kuni sõiduvahend saab täis. Kui ühinetakse uue ruumiga siis tühjendatakse istmed ning täidetakse istmed uute ruumisolejatega.

5 Arendus

Esimesed arendusetapid keskendusid rohkelt Three.js põhifunktsioonide ja objektide loomisele. Idee on luua nullpunktis seisev sõiduvahend ning liigutada maastikku selle ümber, et luua edasiliikumise efekt.

5.1 Veebilehe baas

Rakenduse algseks etapiks loodi Angular-iga projekti üldstruktuur autori poolt leitud malli [9] järgi. See mall andis hea alguspunkti Angulari ja Three.js-iga projekti ülesseadmiseks, kuna see tuli kaasa kasutajaliidese komponentidega ning Bootstrapiga [10]. Hiljem avastas autor, et ei ole võimalik tagarakendusega ühendust luua Angulari kasutades ning autor otsustas Angularist eemalduda ja viis kõik Three.js meetodid üle ühte Javascript faili.

On olemas nüüd Three.js süsteem mis pärast initsialiseerimist ja *createScene()* jooksutamist alustab *render()* funktsiooni jooksutama igal kaadril.

Autor leidis ka hea baasi keskkonna valgustamiseks kasutades kolme valgusallikat kus kaks keskenduvad auto lähedal olevale ümbrusele ning kolmas annab kaugemale liikuvale üllatuselemendile varju. Kõik elemendid loovad varje v.a maastik ise kuna meetod, mis annab maastikule oma värvi, olenevalt maastiku punkti kõrgusest, läheb vastuollu varju loomisega ja tekitab visuaalseid tõrkeid maastiku pinnal.

5.2 Maastiku realiseerimine

Autor lõi maastiku objekti. See on *THREE.Plane()* ehk tasapinnaline ristküliku kujuline objekt, millele määratakse suurus ja mitmest segmendist see koosneb. Segmentide arv on pikkupidi ja laiupidi erinevad, täpsemalt on laiupidi segmente poole vähem, kuna maastiku laiupidine resolutsioon ei ole nii tähtis visuaali poolest ning see on sellevõrra rakendusele vähem jõudlust vajav. Autor valis segmentide arvud kõrgusele ja laiupele erinevad, et hoida kokku igas kaadris tehtud arvutuste kogust, visuaalset kvaliteeti oluliselt mõjutamata.

Maastiku pidi keerama ümber 90 kraadi, et see horisontaalselt oleks, kuna see on vaikimisi vertikaalses asendis. Hiljem renderdamise ajal muudetakse maastiku geomeetriat, et visualiseerida mööduvaid mäestikke.

Autor valis sõiduvahendi liikumissuuna y-teljes positiivse liikumissuuna ning lõi veel ühe *THREE.Plane()* objekti autotee jaoks ja värvis selle tumedaks halliks.

5.3 Sõiduvahend

Järgmiseks etapiks oli vaja luua sõiduvahendi objekt. Arendusaja optimeerimise huvides, leidis autor internetist vabade kasutusõigustega sedaani mudeli. Selle mudeli laadis autor *GLTFLoader*-iga rakendusse sisse ning asetas sõiduvahendi grupp objektile. Grupp objekti kasutades on hiljem sõiduvahendit lihtsam kätte saada ja mõjutada. Sõiduvahendi grupp objektile lisas autor ka istumispunktid mida hiljem saab mängijate tegelaste asetamiseks istmetele ära kasutada.

5.4 Üllatuselement

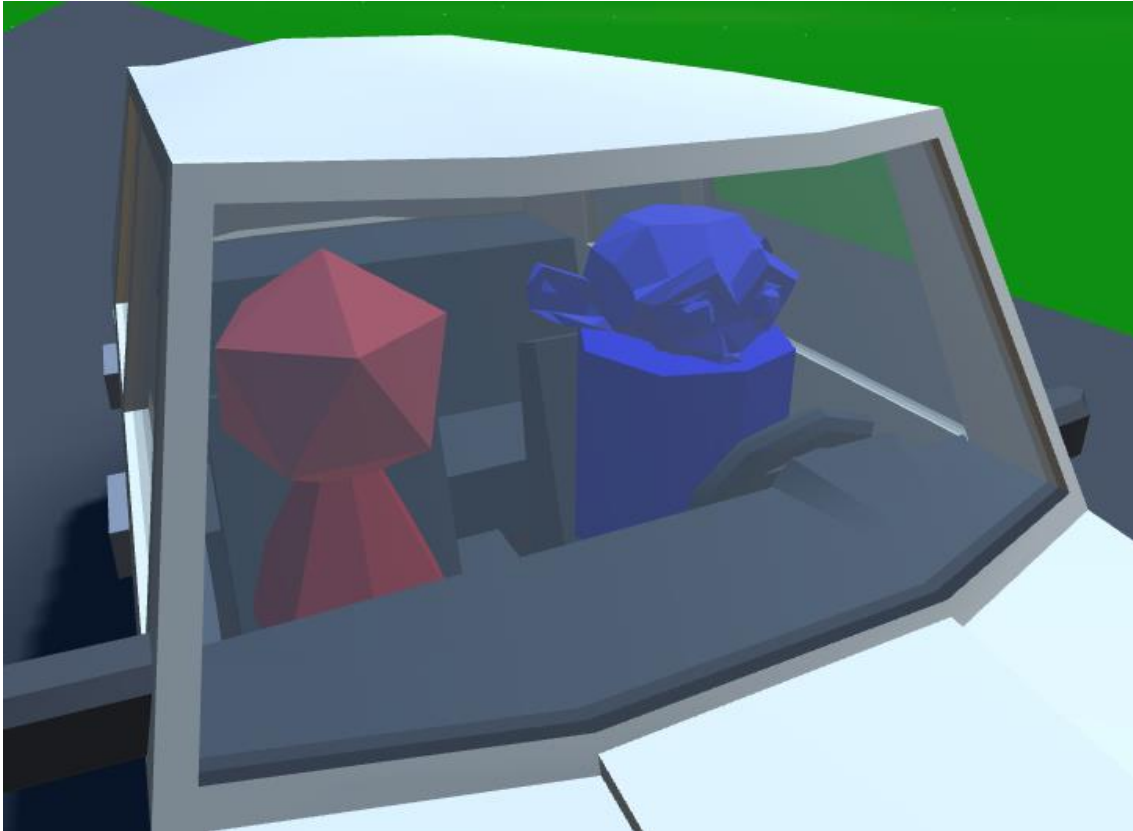
Üllatuselemendiks valis autor puu. Näidatud Joonisel 5 lk 19. Sarnaselt sõiduvahendi mudeliga otsustas autor arendusaja kokkuhoiuks leida internetist vabade kasutusõigustega puu mudeli. Puu loomiseks loodi funktsioon *spawnTree()*, mille abil laetakse puu mudel rakendusse. Funktsiooni uuesti jooksumisel kopeeritakse olemasolev puu uue genereerimiseks. Kõik puud pannakse *normalEggPivot* gruppi mis on alati seotud ajaga sama moodi kui maastik. Seetõttu ei ole vaja igat puud eraldi liigutada vaid ainult *normalEggPivot* gruppi.



Joonis 5. Üllatuselement puu

5.5 Mängija tegelane

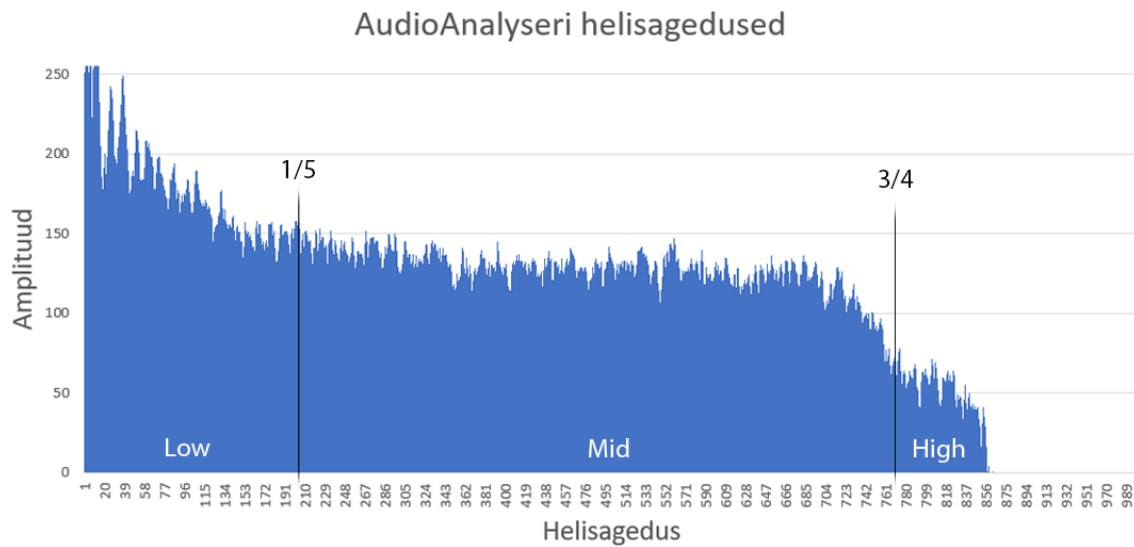
Lihtsus ennekõike otsustas autor luua mängija tegelase kahest osast: pea ja keha. Kuna peade ja kehade mudelid oli raske leida sellisel kujul nagu autor soovis, lõi ta nende mudelid ise Blenderis[11]. Loodi kaks kehakuju ja kaks pead. Nende vahel valitakse suvaliselt kui rakendus genereerib mängijate tegelased. Kuna nendega tuleb kaasa lihtne hall materjal siis genereeritakse ka uus materjal suvalise värviga. Tegelast genereerides laetakse pea objekt sisse istmepunktile ning keha istmepunktile nihutusega allapoole. Kokkuvõttes tekib kehaga tegelane kes sümboliseerib mängijat istumas sõiduki istmel. Tegelased on kujutatud Joonis 6, lk 20.



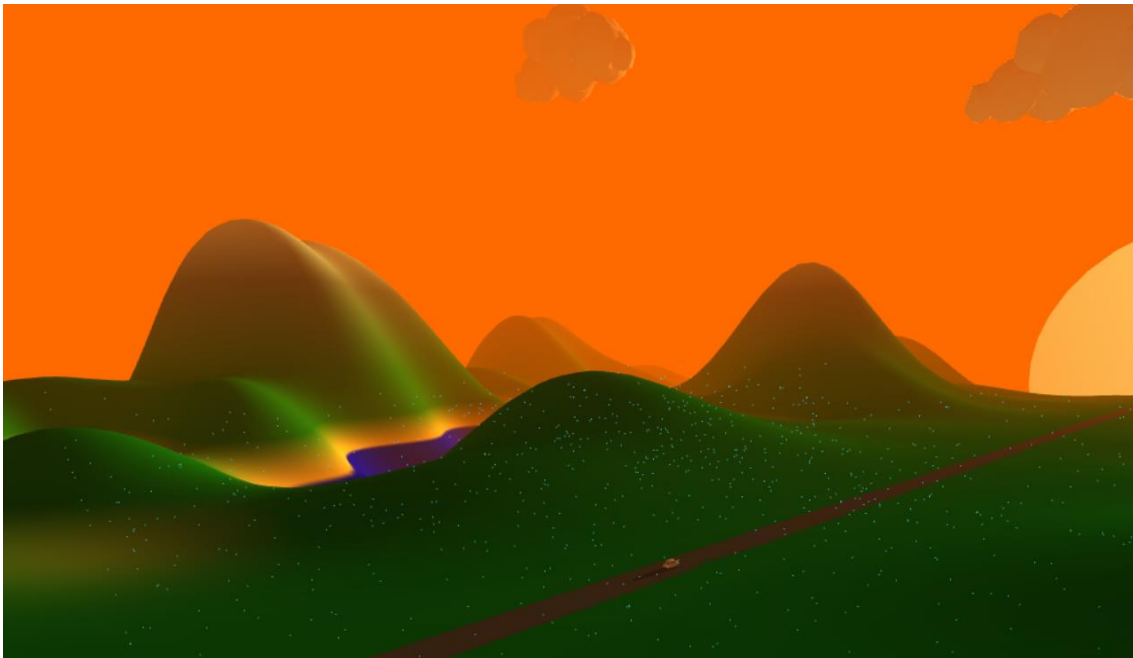
Joonis 6. Erinevad pead ja kehad eri värvides

5.6 Muusika visualiseerimine

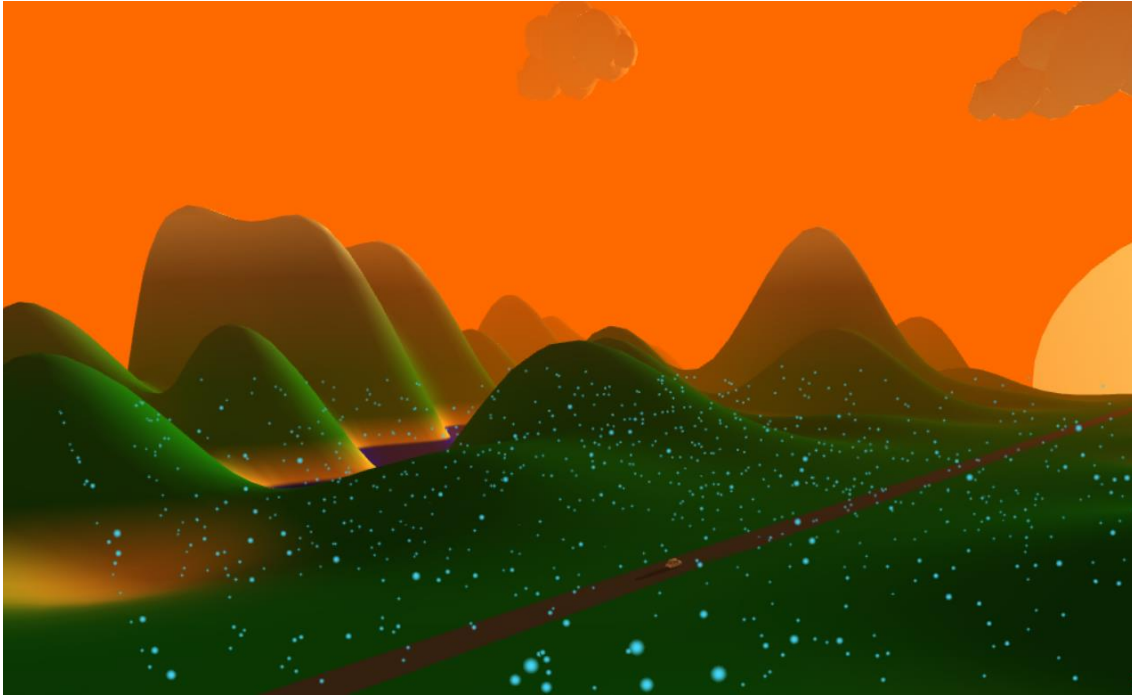
Muusika visualiseerimise loomisega pidi autor arvestama tagaliidese funktsioonidega, mis värskendasid audioelemendi allikat. Pärast seda värskendust oli vaja luua *audioAnalyser* klass, mille abil saab analüüsida heli ning mis tagastab massiivi helisageduste amplituudidest. Autor lõi funktsiooni *getLowMidHigh()* mis grupeerib selle massiivi helisageduste klasside (*low*, *mid* ja *high*) järgi. Kujutatud Joonis 7 lk 21. *Mid* ja *low* leidmiseks leiti nende vasta piirkonna keskmine väärtus, aga *low* jaoks leiti kõrgeim väärtus, kuna bass noodid on tavaliselt üksikud ja vastavad väga väikese osa eest *low* piirkonnas siis võttes selle piirkonna keskmise väärtuse summutaks selle bass noodi välja tulemusest. Neid väärtusi on lihtsam kasutada ülejäänud objektide mõjutamiseks. Eraldi arvutab see funktsioon ka välja positiivse muudu *mid* ja *high* korral, millega saab välja tuua järske trummilööke, mida kasutatakse põhiliselt punktmassi korral. Nende väljunditega mõjutatakse rakenduse komponentide atribuute mis visualiseerivadki muusikat. Joonise 8, lk 21 ja 9, lk 22, vahel on võimalik näha erinevusi maastikus ja punktmassis muusika intensiivsuse tõttu.



Joonis 7. Graaf mis näitab *audioAnalyserilt* saadud helisageduste (vasakult paremale tõusev) ärakasutamise jaotust *getLowMidHigh()* poolt



Joonis 8. Maastik ja punktmass juhul kui on vaikus



Joonis 9. Maastik ja punktmass juhul kui muusika on intensiivne, mäestik on teravam ja punktid on suuremad

5.7 Liidestus tagaliidesega

Tagaliidesest kasutas autor ära kasutajate arvu uuendamise funktsiooni *refreshUserList()*. Selles funktsioonis kutsutakse välja sündmus, mis uuendab autos istuvate tegelaste kogust vastavalt kasutajate kogusest.

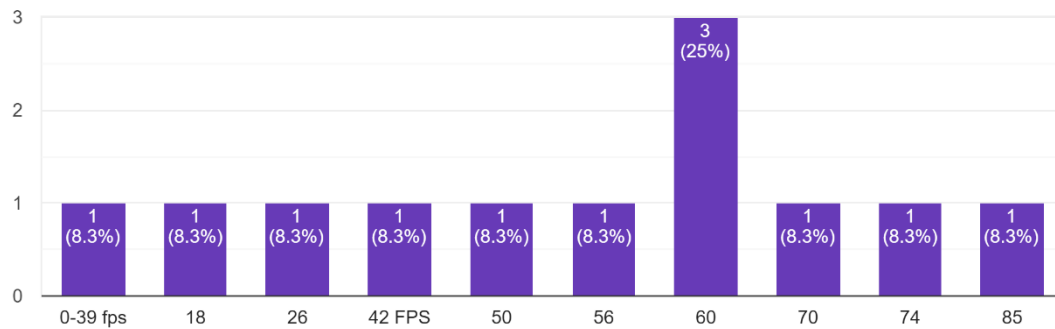
5.8 Kasutajatelt saadud tagasiside

Pärast algset 3D keskkonna loomist ja ülesseadmist, tegi autor koos Rasmus Riismanniga Wanderful Day kohta anonüümse küsitluse. Küsitlusele vastas 12 inimest. Küsitluses tahtis autor põhiliselt teada kui hästi või halvasti jookseb veebirakendus vastajate arvutis (Joonis 10, lk 23). Väheste vastajate tõttu ei saa kindlat vastust, aga esitatud vastustest on võimalik näha, et kasutajatel jookseb 3D keskkond 60 kaadri sekundi ümbruses. Järgmiselt küsimuselt, et kas olenemata eelmisest vastusest on kasutaja rahul saadud kaadrid sekunditega, tulid vastused 100% jah (Joonis 11, lk 23). Selline vastus viitab sellele, et kasutajate kogemus ei ole tugevasti mõjutatud sellest kui hästi rakendus jookseb nende arvutil. Ülejäänud küsimused 3D liidese kohta keskendusid sellele mida kasutajad sooviksid veel näha stseenis (Joonis 12, lk 23). Kasutajad vastasid erinevalt seehulgas rohkemate üllatuselementide variatsiooni lisada või teepealset detaili lisada. Nendehulgas aga kõige tihedaim vastus oli see, et taevas oli

liiga tühi või sooviti pilvi. Vastajate tagasisidest tingitud, lisas autor taevasse lihtsad pilved, mis tõesti seovad kokku kogu stseeni paremini. Neid pilvi on näha Joonis 8, lk 21 ja 9 lk 22.

How many frames per second (fps) do you have on average?

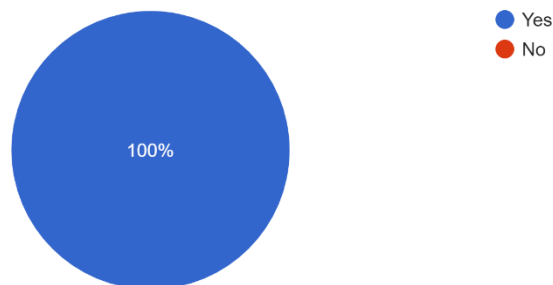
12 responses



Joonis 10. Tagasiside 3D liidese jõudluse kohta

Are you pleased with the amount of fps you have?

12 responses



Joonis 11. vastajate rahulolu nendel oleva jõudluse kohta



Joonis 12. Rakenduse 3D liides küsitluse esitamise ajal

6 Tähelepanekud ja järeldused

Suur osa lõputöö eesliidesest arendas autor Angulari platvormil, aga arenduskäigu lõpupool leidis autor, et kasutades Angulari ei saa ta korrektselt ühendada eesliidest tagaliidesega, ning seetõttu pidi konverteerima Angulari platvormilt tavaliseks Javascriptiks, mis kahjuks kulutas palju arendusaega. Lõputöö autor pani tähele ka, et kuigi audio visualisatsioonid töötavad, ei ole need nii täpsed kui ta sooviks. Kui soovitakse projekti edasi arendades täpsemat muusika reageerivust saavutada on vaja *getLowMidHigh()* funktsiooni täiustada.

7 Edasised arendusvõimalused ja sisu suurendamine

Kuna lõputöö plaaniks oligi luua algne rakendus Wonderful Day edasiarendamiseks siis on sisu suurendamisvõimalusi rohkelt. Sisu suurendamiseks ja arenduseks saab lisada rohkem üllatuselemente, teistmoodi esinevad üllatuselemente (nt linnud ja väiksed külad), teisi maastikke või keskkonna genereerimismeetodeid ja rohkem mängija tegelaste tükke. Samamoodi ka teisi sõiduvahendeid saab lisada. Peatükis 4.2.2 kirjutatud maastiku genereerimise meetodi edasiarendamiseks peaks veel aega pühendama, sest selle kasutuselevõtt mingisugusel paremal kujul oleks palju kiirem kui praegune meetod ning seetõttu annaks ruumi rohkemate detailide lisamiseks.

8 Kokkuvõte

Selle lõputöö eesmärk oli luua prototüüp Wonderful Day veebirakendusele, kus on loodud põhikomponendid rakenduse edasiarendamiseks ja sisu suurendamiseks lõputöö väliselt. See eesmärk sai täidetud, kuna töö arenduskäigus loodigi mitmeid vajalikke rakenduse põhikomponente nagu maastik, üllatuselement, sõiduvahend ja mängija tegelane. Kokku arendati 3D eesliides mille alusel saab Wonderful Day rakendust edasi arendada.

Lisaks saadi praeguselt tehtud küsitluselt vihjeid selle kohta, mida kliendid sooviksid näha rakenduse tulevikus ning ka huvitava teadmise, et jõudlus ei mõjuta suurelt kasutaja rahulolu.

Kasutatud kirjandus

- [1] “What are the health benefits of being social?” [Võrgumaterjal]. Available: <https://www.medicalnewstoday.com/articles/321019> [kasutatud 08.05.2022]
- [2] “Jalopy” [Võrgumaterjal]. Available: <https://excalibur-games.com/products/jalopy> [kasutatud 08.05.2022]
- [3] “VRChat” [Võrgumaterjal]. Available: <https://hello.vrchat.com> [kasutatud 08.05.2022]
- [4] “Babylon.js” [Võrgumaterjal]. Available: <https://www.babylonjs.com> [kasutatud 08.05.2022]
- [5] “Three.js” [Võrgumaterjal]. Available: <https://threejs.org> [kasutatud 08.05.2022]
- [6] “React.js” [Võrgumaterjal]. Available: <https://reactjs.org> [kasutatud 08.05.2022]
- [7] “Angular” [Võrgumaterjal]. Available: <https://angular.io> [kasutatud 08.05.2022]
- [8] “Low poly” [Võrgumaterjal]. Available: https://en.wikipedia.org/wiki/Low_poly [kasutatud 08.05.2022]
- [9] “Angular-Three Template” [Võrgumaterjal]. Available: <https://github.com/JohnnyDevNull/ng-three-template> [kasutatud 08.05.2022]
- [10] “Bootstrap” [Võrgumaterjal]. Available: <https://getbootstrap.com> [kasutatud 09.05.2022]
- [11] “Blender” [Võrgumaterjal]. Available: <https://www.blender.org> [kasutatud 08.05.2022]

Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks¹

Mina, Carl Eric Reinsberg

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose „MUUSIKALE REAGEERIVA 3D KESKKONNA LOOMINE VEEBIRAKENDUSELE WANDERFUL DAY“, mille juhendaja on Gert Kanter
 - 1.1. reprodutseerimiseks lõputöö säilitamise ja elektroonse avaldamise eesmärgil, sh Tallinna Tehnikaülikooli raamatukogu digikogusse lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;
 - 1.2. üldsusele kättesaadavaks tegemiseks Tallinna Tehnikaülikooli veebikeskkonna kaudu, sealhulgas Tallinna Tehnikaülikooli raamatukogu digikogu kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.
2. Olen teadlik, et käesoleva lihtlitsentsi punktis 1 nimetatud õigused jäävad alles ka autorile.
3. Kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest ning muudest õigusaktidest tulenevaid õigusi.

31.03.2022

¹ Lihtlitsents ei kehti juurdepääsupiirangu kehtivuse ajal vastavalt üliõpilase taotlusele lõputööle juurdepääsupiirangu kehtestamiseks, mis on allkirjastatud teaduskonna dekaani poolt, välja arvatud ülikooli õigus lõputööd reprodutseerida üksnes säilitamise eesmärgil. Kui lõputöö on loonud kaks või enam isikut oma ühise loomingu tegevusega ning lõputöö kaas- või ühisautor(id) ei ole andnud lõputööd kaitsvale üliõpilasele kindlaksmääratud tähtajaks nõusolekut lõputöö reprodutseerimiseks ja avalikustamiseks vastavalt lihtlitsentsi punktile 1.1. ja 1.2, siis lihtlitsents nimetatud tähtaja jooksul ei kehti.