

LUMI- JA JÄÄRAKENTEIDEN SISUSTUSSUUNNITTELUOHJELMISTO

Elina Männikkö

Opinnäytetyö
Tietotekniikan koulutusohjelma
Insinööri (AMK)

2014

LAPIN AMMATTIKORKEAKOULU

Tekniikka ja liikenne

Tietotekniikan koulutusohjelma

**LUMI- JA JÄÄRAKENTEIDEN
SISUSTUSSUUNNITTELUOHJELMISTO**

2014

Toimeksiantaja Ohjelmistotekniikan laboratorio pLAB

Tekijä Elina Männikkö

Hyväksytty

Tekniikka ja liikenne
Tietotekniikan koulutusohjelma

Tekijä	Elina Männikkö	Vuosi	2014
Toimeksiantaja	Ohjelmistotekniikan laboratorio pLAB		
Työn nimi	Lumi- ja jäärakenteiden sisustussuunnitteluohjelmisto		
Sivu- ja liitemäärä	50		

Opinnäytetyön tavoitteena oli jatkokehittää Lapland Snow Design -hankkeessa luotua lumi- ja jäärakennusohjelmistoa. Ajatuksena oli tuottaa ohjelmistoon sisustussuunnittelutoiminnallisuus, jonka avulla voidaan sisustaa ohjelmistolla rakennettavia lumi- ja jäärakennuksia. Uusi toiminnallisuus oli tarkoitus tehdä osaksi vanhaa ohjelmistoa. Opinnäytetyön avulla aiottiin kehittää ohjelmiston toimivuutta ja käytettävyyttä. Työn toimeksiantajana toimi Lapin Ammattikorkeakoulun ohjelmistotekniikan laboratorio pLAB, joka osallistui lumi- ja jäärakennusohjelmiston toteuttamiseen.

Tavoitteena oli luoda käyttäjäystävällinen ohjelmisto. Ohjelmiston käyttöliittymällä on suuri rooli tämän saavuttamisessa, siksi opinnäytetyössä päätettiin tutkia käyttöliittymän toteuttamisessa huomioitavia asioita. Opinnäytetyössä tutkittiin myös muutamaa luotua sisustussuunnitteluohjelmistoa, joista saatiin ideoita opinnäytetyössä tuotettuun sisustustoimintoon. Ohjelmiston sisustussuunnittelu päätettiin toteuttaa Unity 3D -pelimoottorilla, jolla myös aikaisempi ohjelmiston versio oli toteutettu. Opinnäytetyössä tutustuttiin Unity 3D -pelimoottorin tekniikkoihin, joita hyödynnettiin sisustussuunnittelutoiminnallisuuden luomisessa.

Työn tuloksena saatiin toteutettua sisustustoiminnot lumi- ja jäärakennusohjelmistoon, jotka mahdollistavat luotujen rakennuksien sisätilojen sisustamisen ja sisustuselementtien muokkaamisen. Ohjelmistoon tuotettiin luotujen suunnitelmien pohjalta uusia malleja, joita käytettiin sisustamisessa.

Avainsanat Unity 3D, sisustussuunnitteluohjelmisto, lumirakentaminen

School of Technology, Communication and
Transport
Information Technology Programme

Author	Elina Männikkö	Year	2014
Commissioned by	Software Engineering Laboratory pLAB		
Subject of thesis	Snow and Ice Building Interior Designing Software		
Number of pages	50		

The subject of this thesis was to develop the Lapland Snow Design Snow and Ice design application. The aim was to create an interior design tool for the application, which enables to furnish the snow and ice structures that have been created using the application. The new add-on was intended to be part of the application by bringing new functions and developing the usability of the software. The challenge was to combine the interior design tool to the existing software. This Bachelor's thesis was commissioned by the Software Engineering Laboratory pLAB which was the developer of the snow and ice building application.

The goal of this thesis was to create a user-friendly software. The software interface played an important role in the achievement of this goal. Therefore it was decided to examine the literature on how to implement a large user interface. In addition a few interior designing applications were explored for this thesis. Thus a few developing ideas were created that could be useful for the interior designing tool.

The result of this thesis was an add-on to the Snow and Ice designing application which enables to furnish the created structures. The interior design tool was created by using the Unity 3D game engine which was the development tool for the Lapland Snow Design application. The Unity 3D techniques were studied and utilized while creating the interior design tool for the application.

Key words Unity 3D game engine, interior designing application

SISÄLTÖ

KUVIOLUETTELO	1
TERMI LUETTELO	2
1 JOHDANTO	3
2 LAPLAND SNOW DESIGN -HANKE	5
2.1 HANKKEEN TAUSTAA.....	5
2.2 LUMI JA JÄÄRAKENTAMISEEN TARKOITETTU OHJELMISTO.....	6
2.2.1 Ohjelmiston tarkoitus.....	6
2.2.2 Toteutus Unity 3D -pelimoottorilla.....	6
2.2.3 Ohjelmiston toimintojen kuvaus.....	7
2.3 OHJELMISTON JATKOKEHITYS.....	9
3 OHJELMISTON KÄYTTÖLIITTYMÄN SUUNNITTELU	11
3.1 KÄYTTÖLIITTYMÄSUUNNITTELUN MERKITYS.....	11
3.2 NIELSENIN MENETELMIÄ.....	11
3.2.1 Ohjelmiston ja käyttäjän välinen kommunikointi.....	11
3.2.2 Poistuminen ja peruuttaminen.....	12
3.2.3 Käyttöliittymän piirteitä.....	13
3.2.4 Virheiden ehkäisy.....	13
3.2.5 Avustaminen ja ilmoitukset.....	15
4 UNITY 3D -PELIMOOTTORI	16
4.1 YLEISTÄ TYÖKALUISTA.....	16
4.2 PELIOBJEKTIT JA KOMPONENTIT.....	19
4.2.1 Peliobjektit.....	19
4.2.2 Colliderit ja raycast-toiminto.....	21
4.2.3 Kamera.....	23
5 SISUSTUSSUUNNITTELUOHJELMISTON MÄÄRITTELY	25
5.1 SISUSTUSSUUNNITTELUOHJELMISTOT.....	25
5.2 SISUSTUSSUUNNITTELUN OMINAISUUKSIEN TUTKIMINEN.....	25
5.3 LUMI- JA JÄÄRAKENNUSOHJELMISTON SISUSTUSTOIMINTO.....	28
6 SISUSTUSSUUNNITTELUOHJELMISTON TOTEUTUKSEN KUVAUS	31
6.1 SISUSTUS TOIMINNON LIITTÄMINEN OHJELMISTOON.....	31
6.1.1. Sisustaminen 2D-tilassa.....	31
6.1.2 Toteuttaminen 3D-tilassa.....	33
6.2 SISUSTUSOBJEKTIEEN LISÄSOMINAISUUDEN KEHITTÄMINEN.....	35
6.3 SISUSTUSOBJEKTIEEN TOIMINNALLISUUDET.....	39
6.4 UUDET VALIKKO-OMINAISUUDET.....	41
6.4.1 Sisustusobjektin valintavalikko ja -painikkeet.....	41
6.4.2 Haamuobjekti.....	42
6.4.3 Aktiivisen sisustusobjektin toimintavalikko.....	43
6.5 SISUSTUSMALLIT.....	43
7 POHDINTA	46
LÄHTEET	48

KUVIOLUETTELO

Kuvio 1. Lumi- ja jäärakennusohjelmiston rakennusnäkyvä.....	8
Kuvio 2. Lumi- ja jäärakennusohjelmiston kävelytila lentokameranäkymästä	9
Kuvio 3. Lumi- ja jäärakennusohjelmiston rakennus- ja kävelytilojen päävalikot	14
Kuvio 4. Unity 3D Editor	17
Kuvio 5. Inspector-ikkuna	18
Kuvio 6. Konsoli-ikkuna ja saatuja viestejä	19
Kuvio 7. Sphere-peliobjekti ja sen komponentit	20
Kuvio 8. Kuvan kaappaus peliobjektien ryhmittelystä Unity 3D -projektissa	21
Kuvio 9. Erikokoisia ja -muotoisia collidereita	22
Kuvio 10. Esimerkkikoodia Raycast-toiminnosta	23
Kuvio 11. Lumi- ja jäärakennusohjelmiston kaksi eri kamerakulmaa	24
Kuvio 12. Autodesk Homestyler katselutila (ruudunkaappauskuva)	26
Kuvio 13. Homestyler- ja RoomSketcher -ohjelmistojen sisustusobjektien valintavalikot	27
Kuvio 14. Ensimmäinen suunnitteluversio sisustuselementtien lisäämisestä 2D-tilassa.....	32
Kuvio 15. Ensimmäinen suunnitteluversio sisustusobjektin valitsemisesta	33
Kuvio 16. Sisustamisnäkyvä seinän läpi.....	34
Kuvio 17. Paikkaobjektit on asetettu ja valikkopaneelit ovat näkyvillä.....	36
Kuvio 18. Rakennuksen paikkaobjektien ja niiden sisustusobjektien perintäkaavio	37
Kuvio 19. Kupolin paikkaobjektiin luotu lista	38
Kuvio 20. Listan elementit luodussa skriptissä	39
Kuvio 21. Valittu sisustusobjekti ja sen toimintavalikko.....	40
Kuvio 22. Suunnitelma kuva sisustuselementin valintavalikosta	41
Kuvio 23. Näyttämömallin haamuobjekti.....	42
Kuvio 24. Mallin UV-map ennen käsittelyä	45
Kuvio 25. UV-mapattu sisustusobjekti (vasen) ja alkuperäinen mallikappale	45

TERMILUETTELO

Blender	3D-grafiikan mallinnusohjelma.
Editor	Unity 3D -pelimoottorin näkymä, joka sisältää pelimoottorin toimintaikkunat (Unity 3D 2014f).
FBX	3D-mallien luomiseen käytettävä tiedostomuoto. (Autodesk Inc. 2014).
Mesh	Objektin 3D-malli (Unity 3D 2013d).
pLAB	Lapin Ammattikorkeakoulun ohjelmistotekniikan laboratorio.
Scene	Näkymäikkuna Unity 3D -pelimoottorissa, joka sisältää kaikki luodut peliobjektit (Unity 3D 2013b).
Script	Ohjelma eli ohjelmointikielellä kirjoitettu tekstitiedosto (Lennes , M. 2004).
Shader	3D-mallin materiaalin pintaan käytettävä lisäominaisuus, joka muuttaa muun muassa pinnan valaistusta (Unity 3D 2013c).
Skaalaus	Peliobjektin koon muuttaminen.
Tag	Peliobjektin tunnistamiseen käytetty ominaisuus (Unity 3D 2013e).
UV-Mapping	2D-tekstuurin asettelu 3D-objektiin (Blender 2014).

1 JOHDANTO

Lumi ja jää ovat tärkeät materiaalit Lapin arktisessa muotoilussa ja matkailussa. Koko ajan kehitellään uusia ideoita, miten näitä materiaaleja voitaisiin käyttää hyödyksi muun muassa markkinoinnissa. Opinnäytetyön tarkoituksena on kehittää Lapland Snow Design -hankkeessa tuotettua lumi- ja jäärakennusohjelmistoa. Ohjelmiston avulla käyttäjällä on mahdollisuus suunnitella ja luoda jää- ja lumirakennelmia. Opinnäytetyön ideana on toteuttaa ohjelmistoon lisätoiminnallisuus, jonka avulla käyttäjä voi sisustaa luomiaan rakennuksia. Opinnäytetyön tilaajana on Lapin Ammattikorkeakoulun ohjelmistotekniikan laboratorio pLAB, joka oli kehittämässä lumi- ja jäärakennusohjelmistoa. Opinnäytetyön ensimmäisessä osiossa kerrotaan hankkeen ja ohjelmiston taustaa, kuten mistä kaikki lähti liikkeelle ja minkälainen lumi- ja jäärakennusohjelmisto on.

Opinnäytetyön ideana oli tuottaa lumi- ja jäärakennusohjelmistoon uusia toiminnallisuuksia, jotka lisäisivät ohjelmiston käytettävyyttä ja parantaisivat sen visuaalisuutta. Ohjelmiston tärkeimpiin osioihin kuuluu sen käyttöliittymä. Hyvä ja toimiva käyttöliittymä mahdollistaa miellyttävän käyttökokemuksen. Tämän vuoksi päätettiin tutkia käyttöliittymän toteutuksessa huomioitavia asioita. Opinnäytetyöprosessiin kuului myös valmiiden sisustussuunnitteluohjelmistojen tarkastelu. Selvitettiin, millaisia toiminnallisuuksia tällaiset ohjelmistot sisältävät. Tutkimisen aikana syntyneitä ideoita hyödynnettiin sisustussuunnittelutoiminnon suunnittelussa ja toteuttamisessa. Myös käyttöliittymäratkaisuihin löytyi ideoita näistä ohjelmistoista.

Sisustussuunnittelutoiminnon luominen päätettiin tehdä Unity 3D -pelimoottorilla, jolla lumi- ja jäärakennusohjelmisto on toteutettu. Pelimoottorin tarjoamat monipuoliset ominaisuudet mahdollistavat monenlaisen ohjelmiston luomisen. Opinnäytetyöprosessin aikana tutustuttiin muutamaa Unity 3D:n ominaisuuteen, joita käytettiin hyödyksi sisustustoiminnon kehittämisessä. Teoria- ja suunnitteluosuuden jälkeen kerrotaan kuinka sisustamisohjelmiston toiminnot toteutettiin.

Lopuksi työn pohdintaosuudessa arvioidaan opinnäytetyön tuloksia ja miettään, mitä mahdollisesti olisi voitu tehdä toisin. Pohdintaosuudessa kerrotaan myös opinnäytetyöprosessin aikana syntyneistä jatkokehitysideoista.

2 LAPLAND SNOW DESIGN -HANKE

2.1 Hankkeen taustaa

Pohjois-Suomen matkailun vetonauloihin kuuluvat luonto ja arktiset olosuhteet. Arktisiin olosuhteisiin mielletään lumi ja jää, joita hyödynnetään muun muassa graafisessa tuotannossa, matkailussa sekä markkinoinnissa. Jää- ja lumiosaamista kehitetään koko ajan, ja pyritään luomaan uusia, omaperäisiä markkinointitapoja, joiden avulla on tarkoitus lisätä kiinnostusta muun muassa lumi- sekä jäämateriaaleihin. Pohjoisessa on monia lahjakkaita ammattilaisia, joiden taidot halutaan tuoda esille. Tavoitteena on myös kasvattaa myyntiä sekä avata uusia yhteyksiä ulkomaille muun muassa omaperäisten vienti-ideoiden avulla. Näiden ideoiden pohjalta perustettiin Lapland Snow Design -hanke, jonka aikana kehitettäisiin ja toteutettaisiin näitä ideoita. Hankkeen avulla haluttiin luoda uusia yhteyksiä, ja siten lisätä kysyntää sekä vientiä. Hankkeen teemoihin kuuluivat Pohjois-Suomen muotoiluosaaminen sekä lumi- ja jäärakennukset. (Lapland Snow Design -hanke 2013.)

Hankkeen avulla pyrittiin lisäämään myös yhteistyötä eri yritysten ja tahojen kanssa. Heidän palvelunsa haluttiin yhdistää, jotta saataisiin enemmän mahdollisuuksia uusien väylien luomiseen. Yhtenä kehittämisajatuksena oli toteuttaa Lapin yliopiston ja Rovaniemen ammattikorkeakoulun yhteistyönä ohjelmisto, jolla kuka tahansa pystyy suunnittelemaan ja toteuttamaan lumirakennuksia. Ohjelmiston prosessointiin osallistuivat myös eri yritykset. Heiltä saatiin ideoita lumirakennemalleihin, millaisia kappaleita voitaisiin luoda. Ohjelmiston avulla halutaan myös markkinoida lumiosaamista ja viestittää käyttäjälle, että mikä tahansa on mahdollista (pLAB 2014).

Hankkeen toteuttajiin kuuluivat Rovaniemen korkeakoulut (Lapin yliopisto ja Rovaniemen ammattikorkeakoulu) ja Rovaniemen kehitys. Useat yritykset osallistuivat hankkeeseen muun muassa kehittämällä ideoita ja niiden toteuttamiseen. Hankkeen rahoittajana toimivat korkeakoulut, useat yritykset sekä Lapinliitto. (Lapland Snow Design -hanke 2013.)

2.2 Lumi ja jäärakentamiseen tarkoitettu ohjelmisto

2.2.1 Ohjelmiston tarkoitus

Olin kehittämässä Lapland Snow Design -hankkeeseen kuuluvaa suunniteluohjelmistoa Lapin ammattikorkeakoulun ohjelmistotekniikan laboratoriossa pLAB:ssa. Tarkoituksena oli tuottaa ohjelmisto, jolla käyttäjä pystyy suunnittelemaan lumi- ja jäärakennuksia annettujen mallien avulla. Ohjelmisto olisi tarkoitus julkaista hankkeelle tehdyllä sivustolla. Käyttäjryhmää ei ole rajattu, eli kuka tahansa yksittäinen henkilö tai yritys pääsee käyttämään ohjelmistoa. Käyttäjä pystyy lähettämään tarjouspyynnön tekemästään rakennelmasta ohjelmiston avulla.

Hankkeeseen kuuluva ohjelmisto toteutettiin myynnin edistämiseksi. Tarkoitus oli luoda käyttäjäystävällinen ohjelmisto, jossa käyttäjä toimii tuotteensa suunnittelijana. Käyttäjälle ei kaupata valmista tuotetta, vaan hän luo sen itse ohjelmiston avulla. (Yliharju 2014.) Tarkoituksena on myös luoda kuva, mitä kaikkea on mahdollista tehdä lumen ja jään avulla. Rakentamisen aikana käyttäjälle syntyy jonkinlainen hahmotelma siitä, millaisia lumi- sekä jäärakennukset voisivat olla. Ohjelmistossa mahdollistetaan rakennusmallien tutkiminen. Tämä myös lisää käyttäjän tietoisuutta ja auttaa paremmin hahmotamaan lumi- ja jäärakenteita.

Käyttäjälle annetaan mahdollisuus käyttää omaa luovuuttaan lumi- ja jäärakennusten suunnittelussa. Hän saa luoda itselleen mieluisat rakennelmat, jotka hänen olisi mahdollista tilata. Ohjelmistoon on asetettu rajoitukset rakentamiseen, jotta käyttäjä ei pysty luomaan toteuttamiskelvottomia rakennelmia. Rajoitukset perustuvat oikean elämän rakentamismääräyksiin, jotka on saatu tietoon ammattilaisilta. Näin käyttäjän ei tarvitse miettiä rakentaessaan, onko hänen rakennelmansa mahdoton toteuttaa.

2.2.2 Toteutus Unity 3D -pelimoottorilla

Ohjelmistoa suunniteltaessa päätettiin ohjelmiston toimintoihin käyttää pelillisiä ominaisuuksia. PLAB:ssa on kehitetty monenlaisia visualisointisovelluksia, ja yksi käytetty työkalu on pelimoottori nimeltä Unity 3D. Lumi- ja jäärakennusohjelmisto päätettiin toteuttaa pelimoottorin avulla, sillä sen katsottiin olevan oiva työkalu visuaalisen ohjelmiston toteuttamiseen. Ohjelmiston käyt-

tö perustuu muun muassa pelillisten toimintojen hyödyntämiseen, kuten liikkeeseen sekä 3D-maailman ominaisuuksiin. Tämän tyylisten asioiden toteuttamiseen pelimoottorit ovat sopivia työkaluja, siksi työkaluksi valittiin Unity 3D. (Laakko 2014.)

Valmis ohjelmisto on tarkoitus sijoittaa hankkeelle tehdylle sivustolle, jotta kuka tahansa pystyisi sitä käyttämään ja tekemään tarjouspyyntöjä. Tämän vuoksi ohjelmisto päätettiin luoda ja julkaista Web-alustalle, joka mahdollistaisi ohjelmiston käytön selaimessa. Julkaisun tekemiseen ei tarvitse hankkia erikseen lisenssiä. Tämä on mahdollista tehdä Unity 3D -pelimoottorilla, ja se vaikutti myös ohjelmiston työkalun valintapäätökseen.

2.2.3 Ohjelmiston toimintojen kuvaus

Lumi- ja jäärakennusohjelmisto koostuu kahdesta osiosta: niin kutsutusta kävelytilasta ja rakennustilasta. Kummatkin osiot sisältävät omat ominaisuudet ja toiminnot lumirakennuksien luomisen ja tutkimisen saavuttamiseksi. Tilan vaihdon voi tehdä päävalikon kautta, joka on käyttäjälle koko ajan näkyvillä.

Oletusnäkymänä käyttäjä siirtyy suoraan rakennustilaan. Rakennustila on 2D-näkymä, joka on kuvattu ylhäältäpäin. Käyttäjä aloittaa rakentamisen aina tyhjältä rakennuspohjalta. Hänen on mahdollista valita erimuotoisia ja kokoisia rakennusobjekteja objektivalikosta, johon eri objektit on ryhmitelty. Objektivalikko näkyy kuviosta 1 vasemmassa laidassa. Rakennusobjekteja pystytään liikuttamaan ja liittämään toisiinsa. Liittäminen on tehty automaattiseksi, eli käyttäjän tarvitsee vain viedä valittu objekti lähelle toista objektia ja liittäminen tapahtuu itsestään. Näin käyttäjän ei tarvitse itse arvioida ovatko rakennusobjektit yhdistyneet, ja se nopeuttaa käyttämistä. Kuviosta 1 näkyy ohjelmiston rakennusosio, sen toimintoja ja rakennettu rakennelma.

Käyttäjän klikatessa luotua rakennusobjektia se muuttuu aktiiviseksi. Objektille ilmestyy valikko, josta voidaan tehdä muutoksia objektiin. Objektia on mahdollista kääntää halutessa eri suuntaan (myötä- ja vastapäivään). Joidenkin rakennusobjektien materiaali on mahdollista vaihtaa joko jääksi tai lumeksi. Valikko sisältää myös objektin poisto-ominaisuuden. Halutessaan käyttäjä voi poistaa kaikki rakennukset kerralla ja aloittaa alusta rakentami-

sen. Osa objekteista sisältää ovia, jotka voi aukaista ja sulkea. Kuviossa 1 valikko näkyy ylhäällä, kupoliobjektin alapuolella.



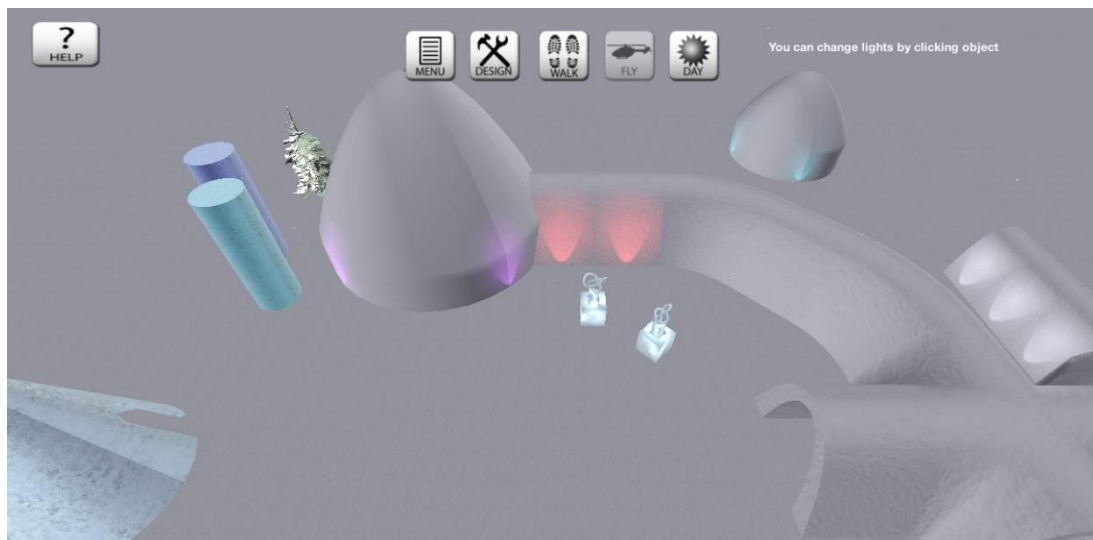
Kuvio 1. Lumi- ja jäärakennusohjelmiston rakennusnäkyvä

Rakennusalue on rajattu selkeästi, jotta käyttäjä näkisi mihin on mahdollista rakentaa. Rajauksen avulla halutaan myös estää loputun rakentaminen. Käyttäjä voi asettaa koristeobjekteja, kuten puita, rakennusalueen ehostamiseksi. Hänellä on mahdollista lisätä myös talokappaleita luomaan kuvan siitä, millaiselle alueelle rakennelma oikeasti suunnitellaan. Talokappaleet auttavat käyttäjää hahmottamaan, kuinka lumi- ja jäärakennukset tulisi asettaa, jos lähetyvillä olisi taloja.

Rakentamisen jälkeen käyttäjällä on mahdollisuus tutkia rakentamaansa rakennelmaa. Hänellä on käytössään niin kutsuttu kävelytila, jossa on mahdollisuus kulkea rakennusalueella. Myös rakennettujen rakennusten sisälle pystyy menemään ja tutkimaan niitä sisältäpäin. Tutkiminen tapahtuu 3D-näkyvässä, jotta kokemuksesta saisi mahdollisen aidon tuntuksen. Rakennelmaa ja aluetta voi halutessa tutkia ylhäältäpäin. Tämä mahdollistetaan lenkokameratilan avulla, jossa käyttäjä näkee yläilmoista koko alueen. Kuviossa 2 näkee luodun rakennelman lentokameran kuvakulmasta.

Jokainen rakennusobjekti sisältää valot, jotka voi kytkeä päälle kävelytilassa. Haluttu rakennusobjekti klikataan aktiiviseksi. Sen jälkeen käyttäjälle ilmestyy valon värin vaihto -paneeli, josta käyttäjä pystyy valitsemaan mieluisan värin. Kuviossa 2 näkyy lumi- ja jäärakennuksia, joissa osassa on valot päällä. Kä-

velytilassa on kaksi näkymää: päivä ja yö. Päivänäkymä on oletuksena ja sen pystyy vaihtamaan yötilaan, jolloin valot näkyvät paljon paremmin.



Kuvio 2. Lumi- ja jäärakennusohjelmiston kävelytila lentokameranäkymästä

2.3 Ohjelmiston jatkokehitys

Ohjelmiston toteutus päättyi vuoden 2013 loppupuolella. PLAB:n ja Lapin yliopiston projektivastaavat ehdottivat, että ohjelmistoa voisi jatkokehittää. Kehittämiskohteena oli ohjelmiston käytettävyyden ja visuaalisuuden lisääminen. Eräänä ideana oli toteuttaa ”lisäosa”, jolla on mahdollista sisustaa soveluksella tehtyjä rakennuksia. Idea kuulosti mielenkiintoiselta, ja se vaikutti tämän opinnäytetyön aiheen valintaan.

Tarkoituksena olisi saada ohjelmistoon lisää ilmettä, sillä muun muassa ohjelmiston rakennukset ja maisema ovat hyvin karun näköisiä. Opinnäytetyön tavoitteena on suunnitella ohjelmistoon lisäominaisuus, jonka avulla voidaan tehdä lisäyksiä rakennuksen sisäosiin. Ideana on toteuttaa sisätilojen sisustamismahdollisuus erilaisten sisustusobjektien avulla. Käyttäjälle annetaan valikoima sisustusobjekteja, joiden avulla hänen on mahdollista luoda erilaisia tiloja rakennuksiin. Rakennusobjekteihin saa luotua erinäköisiä huoneita, sillä jokaiseen rakennukseen on määritetty ennakolta sisustusvaihtoehdot. Mitä isompi ja laajempi rakennus on, sitä enemmän sitä voi sisustaa erityylikseksi.

Sisustusominaisuus parantaa ohjelmiston käytettävyyttä, ja sen avulla saadaan luotua lisää toiminnallisuutta koko ohjelmistoon. Kehittämisessä täytyy

ottaa huomioon vanha ohjelmisto ja sen lähdekoodi. Lisäosan täytyy liittyä kokonaisuuteen sulavasti ja niin, että koko ohjelmisto on sen jälkeen yhdenmukainen. Myös visuaalisuuden suunnittelussa täytyy ottaa huomioon vanhan osion ulkomuoto.

3 OHJELMISTON KÄYTTÖLIITTYMÄN SUUNNITTELU

3.1 Käyttöliittymäsuunnittelun merkitys

Käyttöliittymä on ohjelmiston moduuli, joka kommunikoi ohjelmiston muiden osien kanssa. Ohjelmiston käyttöliittymä toimii käyttäjän ja ohjelmiston välisenä viestintävälineenä. Se on käyttäjälle näkyvässä oleva komponentti, jonka ansiosta käyttäjän ei tarvitse koskea ohjelmiston lähdekoodiin voidakseen tehdä ohjelmiston toimintoja. Käyttötavasta riippumatta käyttöliittymä tulee suunnitella huolella, jotta ohjelmiston osat toimivat toisistaan riippuen. Suunniteltaessa täytyy ottaa myös mahdolliset käyttäjät huomioon. Käyttöliittymän käytettävyydellä on suuri vaikutus ohjelmiston miellyttävään käyttökokemukseen. (Kuutti 2003, 13.)

Käyttöliittymän suunnittelua ja toteuttamista varten on kehitetty erilaisia sääntöjä, joita noudattamalla saadaan luotua hyvä käyttöliittymä. Sääntöjä noudattamalla suunnittelija osaa kiinnittää huomiota eri tilanteisiin, joita voi ohjelmistossa sattua. Esimerkiksi virhetilanteiden sattuessa tiedetään informoida tapahtuneesta käyttäjälle, ja mahdollisesti estää virheiden tapahtumista etukäteen. Käyttäjän käyttötapauksia täytyy liittää yhteen, jotta saataisiin yhtenevä ohjelmisto. Tämä asia huomioidaan myös käyttöliittymän implementoinnissa. Säännöt, eli niin kutsutut heurestiikat, koskevat käyttöliittymän useita ominaisuuksia. (Auer 2006.) Eräs tunnettu heurestiikkaoppi on Jacob Nielsenin luoma Nielsenin lista, joka sisältää käyttöliittymän vaatimuksia. Tarkastelemalla eri ominaisuuksia voidaan luoda tehokas käyttöliittymä. (Kuutti 2003, 47–49.)

3.2 Nielsenin menetelmiä

3.2.1 Ohjelmiston ja käyttäjän välinen kommunikointi

Ohjelmistoa ei suunnitella yksittäistä käyttäjää ajatellen, vaan sen suunnittelu keskitetään koko kohderyhmää ajatellen (Kuutti 2003, 140). Tavoitteena olisi, että ohjelmiston käyttö olisi käyttäjille joustavaa ja vaivatonta. Käyttökokemus ei saisi aiheuttaa käyttäjälle tarpeetonta päänvaivaa. Ohjelmisto ei saisi sisältää vaikeusasteita, vaan sen käyttäminen täytyisi olla käyttäjälle yhtä helppoa tai haastavaa. Näin ollen ohjelmiston käyttäjäryhmä olisi hyvä olla tiedossa etukäteen, jotta käyttöliittymä voitaisiin suunnitella ja toteuttaa sellaiseksi,

joka vastaisi käyttäjäryhmän käyttökokemuksia. (Nielsen 1998.) Jotta käyttäjä osaisi käyttää joustavasti ohjelmistoa, täytyy hänen ymmärtää ohjelmiston konsepti. Ihanteellista olisi, että käyttäjä huomaisi ohjelmiston eri toiminnot ilman erillistä apua ja osaisi käyttää niitä luontevasti. (Wiio 2004, 29–32.)

Ohjelmiston täytyisi kielellisesti vastata käyttäjien omaa kieltä eli sanaston täytyisi olla käyttäjäryhmälle sopiva. Täten käyttäjä ei tarvitse erillistä sanakirjaa ymmärtääkseen ohjelmiston kieltä, ja hän osaa käyttää ohjelmistoa luontevammin. (Wiio 2004, 74–76.) Epämääräisten ja harhaanjohtavien toimintojen välttämiseksi täytyy myös ohjelmiston symboliikan olla käyttäjälle helposti tunnistettavissa ja vastata hänen tietämystään. Esimerkiksi jos painikkeen kuvana on roskakori, useat käyttäjät osaavat jo valmiiksi mieltää painikkeen poistotoiminnoksi. Kuvallisen ilmaisun esittämisvaihtoehtona on ilmaista asia pelkkänä tekstinä tai yhdistää sekä teksti että symboliikka. (Auer 2006.)

Suunniteltaessa käyttöliittymää täytyy olla tiedossa käyttäjien käyttötapaukset ja tieto mitä ohjelmistolla pystyy tekemään. Mietitään, miten käyttäjä voi valita toiminnot ohjelmistossa, ja mistä ne löytyvät. Käyttöliittymän toiminnallisuuden, grafiikan ja käyttäjän pitää osata kommunikoida ja toimia keskenään. (Wiio 2004, 212–213.)

3.2.2 Poistuminen ja peruuttaminen

Käyttäjän kaikkia toimintoja ei saisi rajoittaa niin, ettei hän pysty niitä perumaan. Hänelle täytyy antaa joskus lisäaikaa ja mahdollisuus estää tekemisensä. (Kuutti 2003, 58–59.) Käyttäjälle voi sattua tilanne, jolloin hän ei halua viedä toimintoa loppuun asti. Tällöin hänellä täytyisi olla käytössä tilan peruutus- tai poistumistoiminnot. (Nielsen 1995.) Jos esimerkiksi käyttäjä aikoo aloittaa ohjelman alusta, sovellus voi varmentaa käyttäjältä haluaako hän varmasti aloittaa ohjelman uudelleen vai perua toiminnot. Näin varmistetaan ymmärtääkö käyttäjä toimintonsa seurauksen, ja käyttäjä ehtii vielä puuttua tapahtuman estämiseen. (Kuutti 2003, 58–59.)

Ohjelmiston peruutus- ja poistumistoiminnot täytyy olla selkeästi näkyvissä, jotta käyttäjä osaa hyödyntää näitä toimintoja muun muassa virhetilanteen sattuessa. Jos esimerkiksi käyttäjä jäisi kiinni seinään simulaatio-ohjelmistossa, hänellä pitäisi olla näkyvissä ja käytössä toiminnot, joilla hän

voi peruuttaa takaisin edelliseen toimintoon tai kohtaan. Tällöin keskeytetään virhetilanne, ja käyttäjälle annetaan uusi mahdollisuus toiminnon suorittamiseen. (Kuutti 2003, 58–59.)

3.2.3 Käyttöliittymän piirteitä

Koko ohjelmiston tulisi olla yhtenäinen näkymältään ja toiminnoiltaan. Toimintojen tulisi vastata ohjelmiston eri osissa samaa käyttötarkoitusta. Jos esimerkiksi peliohjelman kääntämistoiminnon kuvakkeena olisi myötöpäivään osoittava nuoli, täytyisi sen vastata tätä toimintoa kaikissa ohjelmiston näkymissä, eikä vaihtua liikuttamistoimintoon jossain ohjelmiston osiossa. (Nielsen 1995.) Täten käyttäjän ei tarvitse asettaa kyseenalaiseksi järjestelmän toimintaa.

Lopuksi ohjelmiston kaikki osiot täytyy liittää yhteen yhdeksi kokonaisuudeksi, eli ohjelmiston eri tilat ja osiot pitäisi saada loogisesti yhdistettyä. Tilanvaihto näkymästä toiseen pitäisi tapahtua sujuvasti. Suositeltavaa on, että kaikki osiot olisivat myös graafiselta näkyvyydeltään samankaltaiset, jotta käyttöliittymä olisi mahdollisimman yhtenäinen (Kuoppala, Parkkinen, Sinkkonen, Vastamäki 2006, 155–157). Ohjelmiston oppimisprosessi nopeutuu, jos koko ohjelmisto on tyyliltään samankaltainen ja toiminnallisuus ei muutu hurjasti eri osioissa. (Kuoppala ym. 2006, 178.)

Ohjelmiston käyttö tulisi olla käyttäjälle mutkatonta. Tavoitteena olisi, ettei käyttäjän tarvitsisi opetella muistamaan, kuinka eri toiminnot tapahtuvat. Käyttäjän tulisi käyttää muistiaan minimaalisesti, eikä kohdistaa muistiaan toimintojen suorittamiseen tai etsimiseen. Ohjelmistossa suoritettavat tehtävät ovat helposti löydettävissä, ja tarvittaessa käyttäjää ohjeistetaan ruudulla näkyvän informaation tai esikatselun avulla. (Kuutti 2003, 50–51.) Esimerkiksi ohjelmiston työkalupaneelissa on monenlaisia työkaluja kuvapainikkeina. Käyttäjä saa tarkempaa tietoa työkaluista laittamalla kursorin kunkin painikkeen päälle, jolloin näkyville ilmestyy vihjeteksti. Näin käyttäjän ei tarvitse muistaa tai kokeilla kaikkia työkaluja tietääkseen mitä niillä voi tehdä.

3.2.4 Virheiden ehkäisy

Ilhailtava ohjelmiston käyttökokemus olisi sellainen, jossa käyttäjä välttäisi kaikki virhetilanteet kokonaan. Käyttöliittymän suunnittelun avulla pyritään

minimoimaan ohjelmiston ja sen käyttöliittymän riskejä, ja pyritään korjaamaan huomattuja ongelmakohtia. (Nielsen 1995.) Mahdollisia virheitä havainnoidaan tekemällä ohjelmistoon testauksia, esimerkiksi pyytämällä useaa eri ihmistä testaamaan keskeneräistä ohjelmistoa useassa vaiheessa. Näin voidaan luoda erityyppisille virhetilanteille omat virheenkäsittelynsä sekä informaatiotekstit, jotka näytetään käyttäjälle. Eri tilanteet pitäisi ottaa huomioon ja suunnitella virheilmoitukset joka tilanteen mukaiseksi. (Kuutti 2003, 62–64.)

Käyttäjän toimintoja voidaan rajoittaa ohjelmallisesti, jotta hän välttyisi tekemästä sopimattomia tekoja. Osa rajoituksista on mahdollista olla käyttöliittymässä näkyvissä. Tällöin käyttäjän täytyisi osata päätellä, ettei toimintoa ole mahdollista suorittaa. Esimerkiksi toiminnon ollessa passiivisena, ohjelmiston painike voi muuttua harmaansävyiseksi. Kuvio 3 näkyy kuinka painikkeet on asetettu pois päältä muuntamalla ne hieman läpinäkyviksi. Tekemistä voidaan myös fyysisesti rajoittaa eli ohjelmiston toimintoja voidaan asettaa pois päältä tai piilottaa väliaikaisesti. Kuviossa 3 esitellään lumi- ja jäärakennusohjelmiston eri tilojen päävalikot. Kuvio 3 nähdään, kuinka toisen tilan toimintopainikkeet on asetettu pois päältä tai piilotettu näkymässä. Näin käyttäjä ei voi suorittaa toimintoja väärässä tilassa. (Kuoppala ym. 2006, 134–135.)



Kuvio 3. Lumi- ja jäärakennusohjelmiston rakennus- ja kävelytilojen päävalikot

Käyttöliittymän teossa voidaan hyödyntää aikaisemmin tuotettuja ja tuttuja toimintoja, jotka käyttäjä on oppinut käytettyään toista samantyylistä ohjelmistoa. Vanhan toiminnon hyödyntäminen voi nopeuttaa uuden toiminnon oppimisprosessia. Tällöin puhutaan konvertoinnista, eli sovelletaan vanhoja asioita ja toimintoja uudessa, samantyyllisessä ohjelmistossa. Uudistuneen

tehtävän ei tarvitse vastata täysin vanhaa toimintoa, vaan se voidaan ottaa pohjaksi uuden kehittämisessä. (Kuoppala ym. 2006, 136.) Esimerkiksi sisustusohjelmistoja käyttänyt voi helposti päätellä, kuinka liikutella huonekaluja huoneen sisällä.

3.2.5 Avustaminen ja ilmoitukset

Jokaisesta käyttöliittymän toiminnosta olisi suositeltavaa tehdä dokumentaatio niiden käyttötavasta, vaikka toiminnot saisi helposti suoritettua. (Nielsen 1998). Ohjeistukseen ei tarvitse kirjoittaa kaikista toiminnoista, vaan kertoa niistä toiminnoista, jotka on haastavampaa suorittaa ja on vaikeaa päätellä niiden käyttötarkoitus. Ohjeistus voi olla selkeä, tiivis, tarpeeksi apua antava dokumentti. Kuvia voidaan käyttää apuna tulkitsemisessä, esimerkiksi näyttämällä käyttäjälle kuinka toiminnot tapahtuvat. (Kuutti 2003, 64–66.)

Ohjeiden olisi hyvä olla helposti löydettävissä, jotta käyttäjä osaisi hyödyntää niitä. Tärkeää olisi, että hätätapauksen sattuessa ne olisivat heti saatavilla. Näin käyttäjä ehtii reagoimaan ja puuttumaan ongelmaan nopeasti ja tehokkaasti.

4 UNITY 3D -PELIMOOTTORI

4.1 Yleistä työkaluista

Unity 3D -pelimoottori on pelien luontiin kehitetty työkalu. Työkalussa hyödynnetään 2D- ja 3D-suunnittelua ja niiden ominaisuuksia ohjelmistojen toteuttamiseen. Unity 3D -pelimoottorin käyttöä ei ole rajattu vain pelkästään pelien tuottamiseen, vaan sillä voidaan kehittää erilaisia ohjelmistoja visualisointiin ja simuloimiseen. Unity 3D:n useanlaiset työkalut mahdollistavat monenlaisien ohjelmistojen kehittämisen, vain mielikuvitus on rajana. Ohjelmistosta löytyy ilmainen versio Unity 3D -pelimoottorin virallisilta Internet-sivuilta, jonka saa kuka tahansa ladata ja asentaa käytettäväksi. Vaihtoehtona on myös maksullinen pro-lisenssi, joka sisältää enemmän kehitystyökaluja, laajennuksia ja tukia lisäosiin sekä kirjastoihin. (Menard 2011, 42–43.)

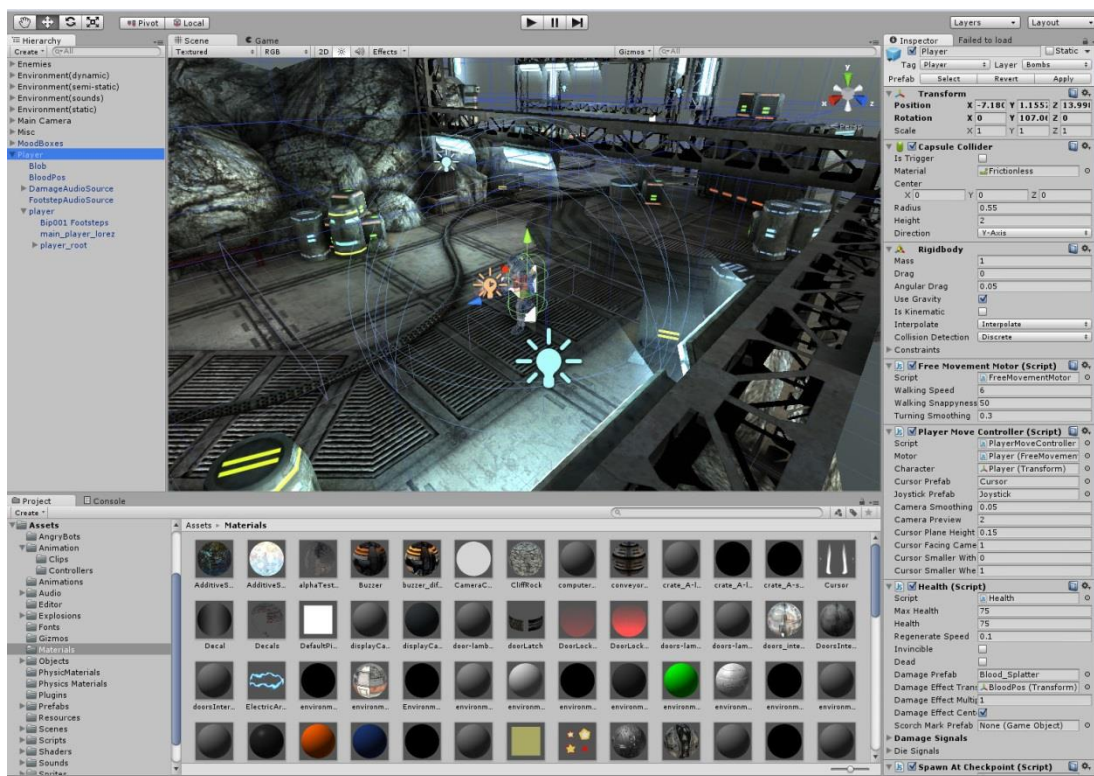
Unity 3D -pelimoottori tukee useita eri tiedostoformaatteja. Pelimoottoriin voidaan tuoda tiedostoja muista ohjelmista, kuten muun muassa 3D-malleja, ääniä ja tekstuureita. (Unity 3D 2014a.) 3D-malleihin kelpaa tiedostomuodoksi .FBX tai .OBJ. Useilla 3D-grafiikkaohjelmistoilla saadaan luotua 3D-malleja näillä formateilla. (Unity 3D 2013a.) Taulukon 1 vasemmassa laidassa on lueteltu eri 3D-grafiikkaohjelmistoja ja niiden tiedostoformaatteja. Taulukon toisella rivillä on 3D-mallien ominaisuuksia, joita voidaan tuottaa näillä ohjelmistoilla. Taulukosta nähdään, mitä 3D-mallien ominaisuuksia Unity 3D tukee. (Unity 3D 2014a.) Unity 3D tukee myös eri ohjelmointikieliä. Luodun Unity 3D -projektin skriptit, eli ohjelmointikiielellä kirjoitetut tekstitiedostot, voidaan kirjoittaa Javascript-, C#- tai Boo-kielillä. (Unity 3D 2014g.)

3D package support						
	Formats	Exporter	Meshes	Textures	Anims	Bones
Maya	.mb & .ma	FBX	X	X	X	X
3D Studio Max	.max	FBX	X	X	X	X
Cheetah 3D	.jas	FBX	X	X	X	X
Blender	.blend	FBX	X	X	X	X
Autodesk FBX	.fbx	FBX	X	X	X	X
SketchUp Pro		FBX	X	X		
3D Studio	.3ds		X			

Taulukko 1. 3D-grafiikkaohjelmistojen tiedostot ja niiden tuetut tiedostomuodot Unity 3D -pelimoottorissa (Unity 3D 2014a)

Unity 3D:n hienoimpiin ominaisuuksiin kuuluu sen laaja julkaisuvalikoima. Pelimoottorilla on mahdollista julkaista ohjelmistoja usealle eri alustalle: Web-alustalle, Windows- ja Mac-käyttäjärjestelmiin sekä erilaisiin pelikonsoleihin, kuten Playstation 3, Wii U, ja Xbox 360. Julkaisumahdollisuus on laajentunut myös mobiilialustoille, kuten Applen iOS ja Googlen Android. (Unity 3D 2014c.) Julkaisualusta olisi hyvä päättää ennen ohjelmiston toteuttamisen aloittamista. Eri alustat eivät hyväksy kaikkia Unity 3D:n tarjoamia ominaisuuksia. Lisäksi alustojen tehot eivät aina riitä suorittamaan näitä. Jokainen alusta sisältää omat määrytykset ja rajoitukset, jotka tulee huomioida ohjelmiston toteutuksessa.

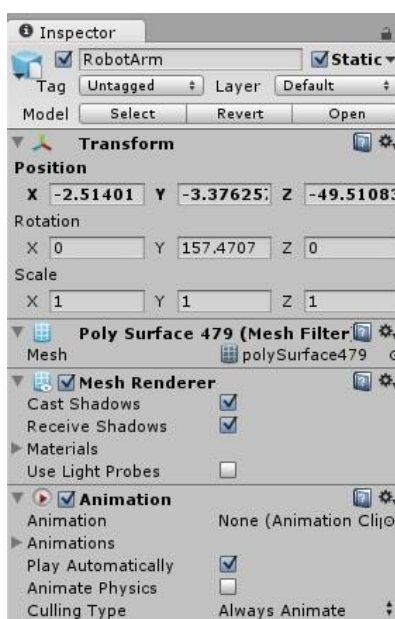
Unity Editor on Unity 3D -pelimoottorin näkymä, jonka ikkunoista seurataan ohjelmiston tapahtumia, esimerkiksi pelin kulkua. Ohjelmisto rakennetaan eri ikkunoita, niiden näkymiä ja ominaisuuksia hyödyntäen. Unity Editor sisältää työkalut, joiden avulla ohjelmistot rakennetaan. Esimerkiksi kaikki Unity 3D -projektissa käytettävät peliobjektit ja -kentät luodaan Unity Editorin avulla. (Unity 3D 2014f.) Kuvioista 4 nähdään Unity Editor ja sen sisältämät ikkunat.



Kuvio 4. Unity 3D Editor

Unity Editor koostuu viidestä eri ikkunasta. Projekti-ikkunassa (Project Browser) näkyy projektin sisältämät tiedostot. Kaikki projektiin tuodut mallit, asetukset, kirjoitetut skriptit ja graafiikat näkyvät hierarkkisessa järjestyksessä projekti-ikkunassa. Jokainen tiedosto on niin kutsutun Assets-kansion alapuolella. Klikattaessa hierarkiassa olevaa kansiota ilmestyy näkyviin ikkuna, josta voidaan katsoa kuvina kansion sisältämät tiedostot. (Unity 3D 2014f.) Muissa ohjelmistoissa luotuja tiedostoja voidaan lisätä Unity 3D-projektiin vain viemällä ne haluttuun kansioon. Tiedostot tulevat automaattisesti osaksi projektia, eikä niiden käyttöä varten tarvitse asentaa lisäohjelmia. (Unity 3D 2014a.) Kuvio 4 näkyy vasemmassa alalaidassa projekti- ja valikkoikkuna.

Unity Editorin sisältämästä Inspector-ikkunasta voidaan katsoa peliobjektin tiedot. Valitessa jonkin peliobjektin Inspector-ikkunaan ilmestyy peliobjektin sisältämät komponentit ja asetukset. Inspector-ikkuna mahdollistaa peliobjektien komponenttien lisäämisen, poistamisen sekä niiden muokkaamisen. (Unity 3D 2014f.) Kuviossa 5 näkyy valitun peliobjektin tiedot ja sisältämät komponentit Inspector-ikkunassa.

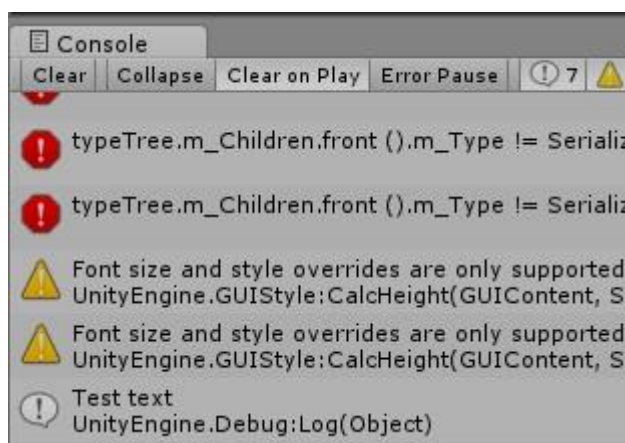


Kuvio 5. Inspector-ikkuna

Ohjelmisto rakennetaan pelinäkömään eli skeneen (scene). Skene-ikkunaan luodaan ohjelmiston oma näköm, joka sisältää muun muassa valaistuksen, luodut peliobjektit ja eri kamerat. Peliobjekteja voidaan muokata skenessä viemällä ne skeneen ja skenestä takaisin hierarkiaan. Kaikki skeneen lisätyt peliobjektit ovat lueteltuna hierarkia-ikkunassa (Hierarchy). Hierarkia-

ikkunassa olevat peliobjektit ovat osa luotua ohjelmistoa. Luotu ohjelmisto voidaan testata ajamalla projekti, jonka jälkeen skene-näkymä muuttuu pelinäkymäksi (Game). Peli-ikkunan saa suurennettua koko ruudun kokoiseksi ajon ajaksi. Kuviossa 4 keskellä näkyy skene/peli-ikkuna ja sen yläpuolella painikkeet, joilla ohjelmisto ajetaan ja pysäytetään. (Unity 3D 2014f.)

Unity Editoriin voidaan asettaa näkyviin myös konsoli-ikkuna (Console), johon ilmestyy kaikki ohjelmiston virheilmoitukset, testausviestit ja varoitusilmoitukset. Kuviossa 6 näkyy erityyppiset viestit konsoli-ikkunassa. Ikkuna ei ole automaattisesti näkyvillä Editorissa, mutta sen voi halutessa lisätä asetusten kautta.



Kuvio 6. Konsoli-ikkuna ja saatuja viestejä

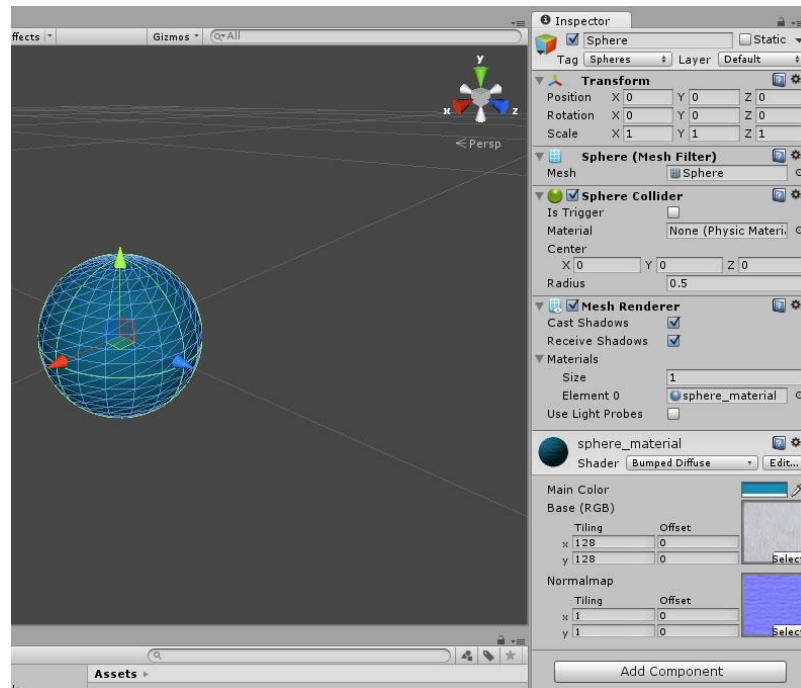
4.2 Peliobjektit ja komponentit

4.2.1 Peliobjektit

Unity 3D:n perusrakennuspalikoihin kuuluvat niin kutsutut peliobjektit (GameObjects). Kaikki projektin objektit kuuluvat peliobjekteihin. Peliobjekteja ovat muun muassa kamerat, erilaiset mallit (kuutiot, sylinterit) ja pelialustat (planes). (Menard 2011,42.) Peliobjektit ovat aluksi niin kutsuttuja tyhjiä peliobjekteja (empty GameObject). Ne eivät sisällä mallia, eivätkä täten näytä mitään. Toisin sanoen tyhjä peliobjektit ovat näkymättömiä peliobjekteja, jotka yksinään eivät tee mitään. (Unity 3D 2010.) Peliobjektiin voidaan lisätä niin kutsuttuja komponentteja, joiden avulla muokataan objektia ja lisätään tälle toiminnallisuutta. Komponenttien avulla pystytään muokkaamaan peliobjekteja erinäköisiksi. Kuviossa 7 näkyy pallomallin sisältävä peliobjekti. Pallolle on asetettu materiaali, joka muuttaa pallon kuviointia tekstuurin materiaalin

sisältävän kuvioinnin mukaisesti. Materiaalikomponentti mahdollistaa myös peliobjektin värin muuttamisen.

Peliobjekteille voidaan lisätä tageja. Tagi on objektin ominaisuus, jonka avulla voidaan ryhmitellä tai erotella peliobjekteja toisistaan. Tageja saa luotua itse ja niitä voidaan lisätä halutuille peliobjekteille. Kuviossa peliobjektille on lisätty Sphere-tagia, joka näkyy Inspektor-ikkunan ylälaudassa.



Kuvio 7. Sphere-peliobjekti ja sen komponentit

Peliobjektille voidaan asettaa toiminnallisuutta lisäämällä sille skriptejä (scripts). Skriptit ovat kooditiedostoja, jotka on kirjoitettu jollain ohjelmointikielellä. Koodissa voidaan määrittellä peliobjektille toimintoja, esimerkiksi peliobjekti voidaan käskä hyppimään hiirtä painettaessa. Halutut skriptit lisätään peliobjektien komponenteiksi. (Menard 2011, 42.) Skriptissä voidaan esitellä pelkkiä muuttujia, joita voidaan hyödyntää peliobjektissa esimerkiksi luomalla objektiin erilaisia listoja. Listoihin voidaan kirjoittaa erityyppisiä jäsenmuuttujia, jotka alustetaan Unity Editorin kautta. Tällöin listan arvojen alustusta ei tarvitse tehdä lähdekoodiin. Esimerkiksi pelimaailmassa listoja hyödynnetään tavara-luetteloinnissa varastoimaan käyttäjän keräämiä esineitä. (Menard 2011, 258.)

Peliobjekteja voidaan lisätä skeneen rajattomasti. Pelimoottori tarjoaa mahdollisuuden ryhmitellä peliobjekteja, jolloin objekteja voidaan asettaa sisäkkäin. Ryhmittely tapahtuu siirtämällä valittu objekti toisen objektin sisälle.



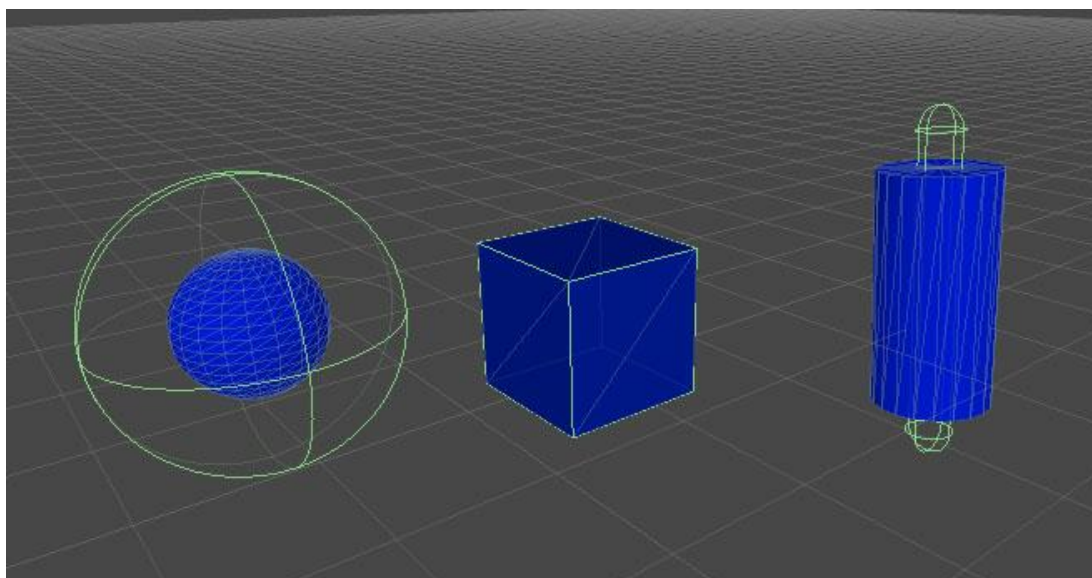
Kuvio 8. Kuvan kaappaus peliobjektien ryhmittelystä Unity 3D -projektissa

Saman peliobjektin alle voidaan asettaa useita objekteja, ja lisättyjen objektien alle voidaan sijoittaa uusia. Kuvio 8 näkee peliobjektien ryhmittelyn hierarkia-näkymässä. Hierarkiassa ylimpänä oleva objekti on kaikkien muiden objektien isäntäobjekti (parent), joka omistaa kaikki sen alapuolella olevat objektit. Alla olevat ovat lapsiobjekteja (child), joihin vaikuttaa isäntäobjektiin koskevat tapahtumat. Jos esimerkiksi isäntäobjektin sijaintia muutetaan, muuttuu myös lapsiobjektien sijainti samassa suhteessa. (Menard 2011,19–20.)

4.2.2 Colliderit ja raycast-toiminto

Collider kuuluu peliobjektin peruskomponentteihin, ja sen tehtävänä on mahdollistaa peliobjektiin koskeminen. Esimerkiksi peliobjekti huomaa hiiren klikkauksen, jos klikatulla objektilla on collider. Collider mahdollistaa myös muun muassa tiettyjen peliobjektien haun ja löytämisen pelikentässä. Collideriin voidaan lisätä materiaali, määrittellä sen sijainti peliobjektissa ja antaa sille koko. Collidereita on monenmuotoisia: pallo (sphere), laatikko (box), sylinteri (cylinder) ja objektin pintaa muotoileva verkko (mesh). (Unity 3D 2014b.) Kuvio 9 havaitaan pallon-, laatikon- ja sylinterinmuotoiset colliderit. Peliobjektin collider kannattaa valita objektin omaavan mallin mukaan, eli mahdolli-

simman mallin pintaan sopiva collider-muoto. Peliobjektiin osutaan tarkemmin, kun collider on oikean kokoinen. Jos peliobjektin collider on sille liian pieni, peliobjektiin ei voida tarttua. Päinvastaisesti liian suuri collider saattaa yltää toisiin peliobjekteihin ja siten hankaloittaa muihin objekteihin osumista. Kuviossa 9 vasemmanpuoleisessa objektissa on liian suuri collider, keskellä collider mukailee objektin pinta ja oikealla collider on osittain objektin sisällä.



Kuvio 9. Erikokoisia ja -muotoisia collidereita

Kameran avulla voidaan ampua säteitä pelimaailmaan. Säteiden avulla voidaan etsiä haluttua peliobjektia. Peliobjektit huomataan säteiden osuessa objektien collidereihin. Tällöin peliobjekti havaitaan ja osutun objektin tiedot jäävät muistiin. Tämän jälkeen objektille voidaan tehdä haluttu toiminto. Säteiden ammuntaominaisuus on nimeltään Raycasting, ja se mahdollistaa peliobjektien tunnistamisen. Havaitsemiseen voidaan hyödyntää hiiren toimintoja, esimerkiksi tarkistamalla hiirtä klikattaessa osuiko klikkaus johonkin peliobjektiin. (Unity 3D 2011.) Esimerkkikoodissa 10 tutkitaan onko klikattu peliobjekti nimeltään Sphere, ja jos on, pyydetään tulostamaan konsoliin teksti. Toimintoa varten täytyy määritellä mistä säde ammutaan, milloin se ammutaan sekä säteen kohde, joka on etsinnässä. Säteelle voidaan määritellä myös etäisyys kuinka pitkälle se yltää. Esimerkissä säde ammutaan pääkamerasta (main camera). (Unity 3D 2011.)

```

// Update is called once per frame
void Update () {

    if (Input.GetMouseButtonDown(0))
    {
        ray = Camera.main.ScreenPointToRay (Input.mousePosition);

        if (Physics.Raycast (ray, out hitObject))
        {
            if (hitObject.collider.gameObject.name == "Sphere")
            {
                DoSomething();
            }
        }
    }

}

public void DoSomething()
{
    Debug.Log("You clicked sphere object!");
}

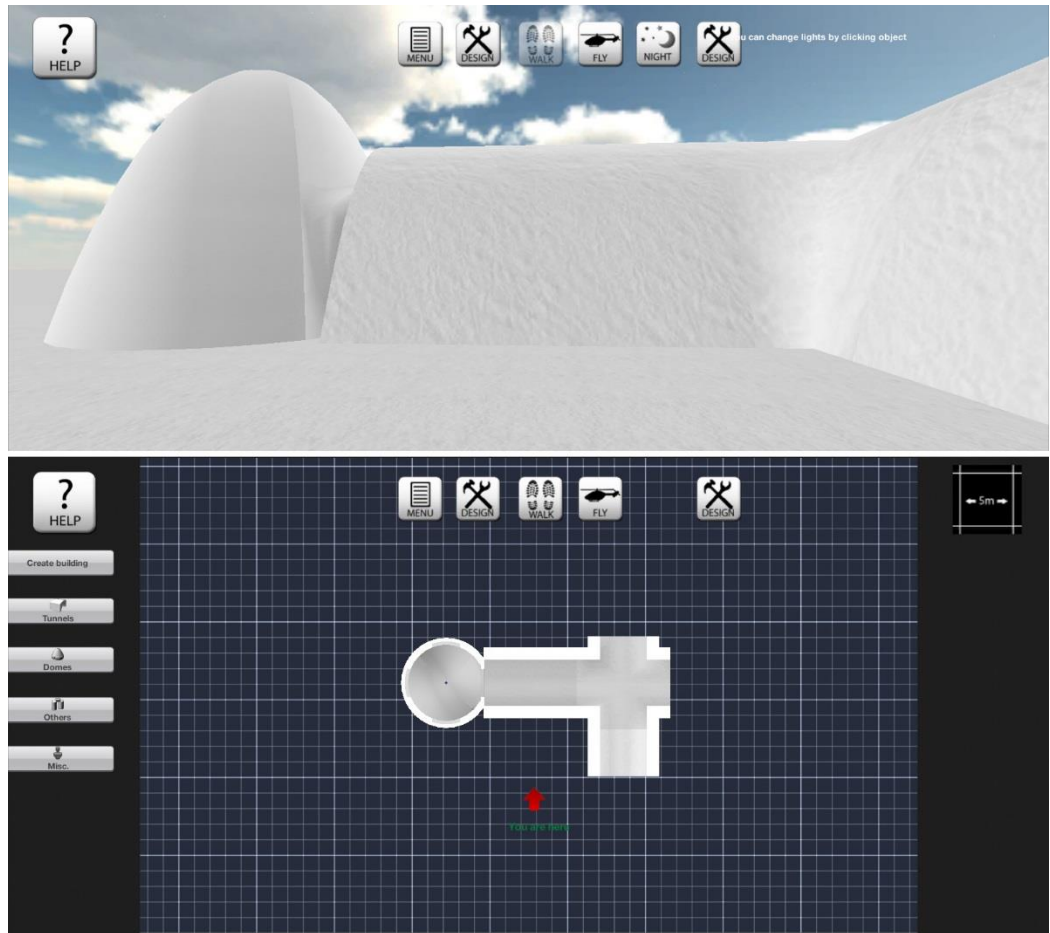
```

Kuvio 10. Esimerkkikoodia Raycast-toiminnosta

4.2.3 Kamera

Kamera kuuluu Unity 3D:n peruspeliobjekteihin, jonka avulla näytetään ohjelmistoon luotu pelimaailma tietokoneen näytöllä. Luotaessa uuden Unity 3D -projektin oletuksena tulee aina niin kutsuttu pääkamera (main camera), jonka avulla nähdään ohjelmiston päätapahtumat. Kameroita voidaan luoda pelimaailmaan useita. Monen kameran avulla voidaan hajauttaa ohjelmiston näkyvyyttä ja toimintaa hyödyntäen kameroiden kuvakulmia ja etäisyyksiä pelialustalla. Näin voidaan selkeyttää käyttöliittymän näkyvyyttä, kun kaikki toiminnot eivät ole samassa näkymässä. Samalla nopeutetaan ja helpotetaan sovelluksen käytön oppimista, kun toiminnot voidaan opetella jaksoissa. Esimerkiksi ylhäältäpäin kuvatussa näkymässä voidaan lisätä uusia peliobjekteja hiiren avulla. Sivukameralla voidaan liikuttaa peliobjektia näppäimistön avulla. Kameran avulla voidaan tutkia peliobjekteja eri perspektiiveissä. Kuviossa 11 nähdään lumi- ja jäärakennusohjelmiston hajautetut kameranäkymät. Kuvan tilanne on otettu samasta kohtaan, mutta eri näkymistä.

Hajauttamisen ja kuvakulman lisäksi kameran etäisyyttä voidaan säädellä. Etäisyyden muuttamiseen voidaan luoda zoomaustoiminto, jonka avulla kamerasta nähty näkymä lähenee tai pienenee. Tätä ominaisuutta voisi käyttää hyödyksi yksityiskohtiin kohdistuvien toimintojen suorittamisessa, jolloin käyttäjän on päästävä lähemmäs nähdäkseen toiminnon kohde tarkemmin. Jos halutaan käyttäjän huomaavan jotain pelimaailmassa, voidaan kamera asettaa automaattisesti osoittamaan tarkasteltavaa kohtaa.



Kuvio 11. Lumi- ja jäärakennusohjelmiston kaksi eri kamerakulmaa

5 SISUSTUSSUUNNITTELUOHJELMISTON MÄÄRITTELY

5.1 Sisustussuunnitteluohjelmistot

Opinnäytetyön aiheen valitsemisen jälkeen päätettiin tutkia muutamaa Internetistä löytynyttä sisustussuunnitteluohjelmistoa, joista saataisiin ideoita suunnitteluun ja toteutukseen. Yritettiin myös hakea kirjallisuutta sisustussuunnitteluohjelmistojen ja niiden toimintojen luomiseen, mutta minkäänlaisia oppaita tällaisten ohjelmistojen tekemiseen ei löytynyt. Tämän vuoksi päätettiin tutkia ja kokeilla luotuja sisustussuunnitteluohjelmistoja. Ajatuksena oli hakea tietämystä sisustussuunnitteluohjelmistojen toiminnoista, eli mitä ohjelmistoilla voidaan tehdä ja miten toiminnot on toteutettu. Tarkoituksena oli myös tarkastella, miltä tällaiset ohjelmistot näyttävät ja ottaa oppia lumi- ja jäärakennusohjelmiston sisustustoiminnon toteutukseen. Tavoitteena oli myös verrata ohjelmistojen huonoja ja hyviä ominaisuuksia, ja täten saada ideoita toteuttamiseen.

Tutkinnan kohteiksi päätyivät K-raudan kylpyhuoneen suunnitteluohjelmisto, Autodeskin Homestyler sekä RoomSketcher. Nämä ovat ilmaisia webalustoille julkaistuja sisustusohjelmistoja, joita voi käyttää ilman rekisteröitymistä. Näissä ohjelmistoissa keskitytään tilojen luomiseen ja sisustamiseen. Ohjelmistoista löytyi paljon samaa, mutta pieniä eroavaisuuksia havaittiin muun muassa toiminnoissa sekä näkymässä.

5.2 Sisustussuunnittelun ominaisuuksien tutkiminen

Sisustusohjelmat sisälsivät kaksi näkymää: 2D- ja 3D-näkymän. Näkymät oli jaettu käyttötarkoitukseltaan sisustamiseen, rakentamiseen ja huoneiden katseluun. Ohjelmasta riippuen sisustamistoiminto oli asetettu vain toiseen näistä tiloista ja rakentaminen toiseen. Tilojen toiminnot oli selkeästi erotettu toisistaan.

Huoneiden tutkiminen oli mahdollistettu sekä sisustamistilassa että pelkässä katselutilassa. Kameran pystyi kääntämään eri kuvakulmiin, jolloin huoneita ja sisustuselementtejä oli mahdollista katsella monesta suunnasta ja perspektiivistä. Kuviossa 13 näkyy Autodesk Homestyler -ohjelmiston tutkimistila. Näkymässä on hyödynnetty seinien läpinäkyvyyttä, jolloin kuvakulmasta riippumatta mikään ei ole näkymän tiellä. Sisustuselementtien korkeuseroja

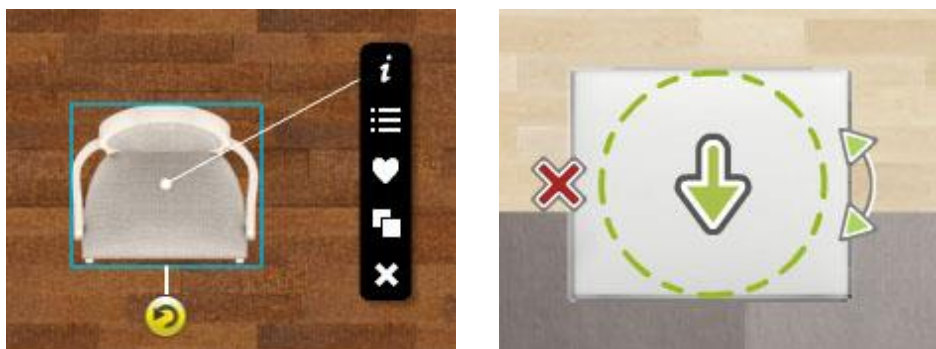
ja mittoja voitiin tarkastella hyvin tarkasti. RoomSketcher -sovelluksessa oli myös mahdollista tarkastella huoneita ihmisen näkymästä. Tilanvaihto onnistui helposti vain painiketta klikkaamalla. Lisätyt sisuselementit jäivät huoneeseen oikeisiin kohtiin, vaikka sisustaminen vaihdettiin katselutilaan. Sisustuselementin liikuttaminen ei ollut usein mahdollista katselutilassa, mutta niille pystyttiin suorittamaan muita toimintoja. Muun muassa elementtejä oli mahdollista poistaa ja niiden tietoja voitiin katsoa, kuten objektin mitat. Kuvassa 12 on esillä valitun objektin toimintapaneeli.



Kuvio 12. Autodesk Homestyler katselutila (ruudunkaappauskuva)

Sisustaminen oli mahdollista aloittaa joko tyhjästä tai valmiiksi sisustetusta huoneesta. Molemmissa tapauksissa voitiin suorittaa samat toiminnot. Vaikka huoneet olivat sisustettuja, rakennukseen oli silti mahdollista tehdä muutoksia. Seiniä voitiin liikuttaa eri kohtiin ja muuntaa erikokoisiksi ja -mallisiksi. Näin käyttäjä pystyi vaikuttamaan myös tilojen muokkaamiseen eikä vain sisustamiseen. Seinien liikuttelu oli kätevä toiminto, mutta välillä hankalaa. Jos huone oli jo sisustettu, seinien muuttaminen oli haastavaa, sillä usein jokin objekti oli seinän liikuttamisen tiellä. Päädyttiin siihen tulokseen, että on parempi rakentaa huone valmiiksi ennen sen sisustamista. Yhtäaikainen rakentaminen ja sisustaminen hankaloittivat molempien toimintojen suorittamista.

Kaikki toiminnot suoritettiin hiirtä käyttäen. Jos sisustusobjektia haluttiin liikuttaa, otettiin siitä hiirellä kiinni ja pidettiin painiketta pohjassa. Jokainen ohjelma sisälsi kuvapainikkeet huoneisiin lisättävistä sisustusobjekteista. Sisustuselementin vieminen huoneeseen tapahtui valitsemalla ensin sisustusobjekti listalta ja vetämällä se haluttuun kohtaan. Sisustusobjekteja pystyi liikuttamaan tarttumalla niistä kiinni ja raahaamalla niitä. Tällöin objekti pysyi hiiren keskipisteessä kiinni. Objekti pysähtyi, jos sitä ei ollut mahdollista viedä johonkin kohtaan. Pariin sisustussovellukseen oli tehty estot, ettei sisustuselementtejä ollut mahdollista asettaa seinien päälle. Oli myös tapauksia, jossa objektin pystyi viemään minne tahansa ja sen liikuttelua ei ollut rajattu, esimerkiksi toisen objektin päälle. Käyttäjän näkökulmasta tämä hieman ärsytti, kun oli mahdollista tehdä epärealistisia toimintoja. Esimerkiksi taulu voitiin asettaa ilmaan leijumaan. Tämä myös hankaloitti sisustamista, koska täytyi olla tarkkana sisustusobjektin asettelussa, jotta se ei jäisi ilmaan kiinni tai menisi seinästä läpi. Useat tällaiset tilanteet tapahtuivat 2D-sisustamisessa, jossa oli hankala arvioida syvyyseroja.



Kuvio 13. Homestyler- ja RoomSketcher -ohjelmistojen sisustusobjektien valintavalikot

Sisustusobjekti oli lisäämisen jälkeen mahdollista valita aktiiviseksi, mikä mahdollisti erilaisien toimintojen tekemisen, kuten objektin poistamisen. Valittu sisustusobjekti merkittiin esimerkiksi rajaamalla, jotta käyttäjä näki selkeästi, mikä objekti oli aktiivinen. Valittu sisustusobjekti oli mahdollista kääntää eri suuntiin. Objektia voitiin myös monistaa, ja näin saatiin nopeasti luotua samanlaisia kappaleita. Sisustusobjekteja pystyttiin myös poistamaan. Valitun objektin toimintopainikkeet oli aseteltu paneeliin, joka leijui objektin lähetyvillä. Tämä nopeutti ja helpotti toimintojen saavuttamista, kun painikkeet olivat mahdollisimman lähellä valittua objektia. Painikkeissa oli selkeät kuvat,

joista ymmärsi heti toimintojen tarkoitukset. Kuviossa 13 näkyvät Homestyler- sekä RoomSketcher -ohjelmistojen aktiivisten sisustusobjektien valintavalikot. Kuvioista nähdään kaksi erityyppistä paneelia: toinen on valitun objektin oikealla puolella ja toinen objektin päällä.

5.3 Lumi- ja jäärakennusohjelmiston sisustustoiminto

Sisustussuunnitteluohjelmistojen tutustumisen yhteydessä kirjattiin ideoita toteutettavaan sisustustyökaluun. Samalla pohdittiin työkalun toimintaa: mitä sillä voisi tehdä, miltä se mahdollisesti näyttäisi ja miten se liitettäisiin osaksi lumi- ja jäärakennusohjelmistoa. Haastatteleamalla asiakas Antti Yliharjua saatiin ideoita toteutukseen. Asiakas osallistui suuresti myös ohjelmiston määrittelyyn. Sisustusosuus haluttiin tehdä heti osaksi vanhaa Unity 3D -projektiä. Projektiin päätettiin luoda sisustamiselle omat skriptit, mallit ja käyttöliittymätoiminnot. Näin uusi lisä saatiin heti osaksi lumi- ja jäärakennusohjelmistoa, mikä helpotti uusien ja vanhojen ominaisuuksien sekä toimintojen yhteensopivuuden testaamista. Opinnäytetyössä toteutettava sisustussuunnittelu koskee lumi- ja jäärakennuksessa olevia rakennusmallien sisätiloja. Sisustamisesta haluttiin luoda ohjelmistoon lisäominaisuus, jolla käyttäjä pysyy nopeasti somistamaan luomiaan rakenteita. Ohjelmiston pääpaino pyritään pitämään lumi- ja jäärakenteiden suunnittelussa, sillä ohjelmisto on kehitetty niiden markkinointiin. Sisustamisella halutaan antaa käyttäjälle ideoita, mitä lumi- ja jäärakennuksiin voisi mahdollisesti tehdä.

Lumi- ja jäärakennusohjelmiston sisustussuunnittelutoiminto päätettiin toteuttaa Unity 3D -pelimoottorilla, jolla myös aikaisempi ohjelmiston versio on tehty. Ohjelmointikieleksi valikoitui C#, sillä se oli tutumpi kieli kuin Javascript, ja myös lumi- ja jäärakennusohjelmisto on toteutettu sillä. Sisustamiseen tarvittiin uudet mallikappaleet, sillä vanhoista mallikappaleista ei löytynyt sisustamiseen sopivia malleja. Mallikappaleiden käsittelyyn käytettiin Blender 3D-mallinnusohjelmaa, jonka saa ladattua ilmaiseksi Internetistä. Blender-ohjelmalla voidaan luoda malleja, joiden tiedostomuodot ovat Unity 3D:n tukemia.

Sisustamiseen tarvittiin uusia toimintapaneeleja ja painikkeita, joiden avulla oli mahdollista suorittaa sisustustoimintoja. Uusiin painikkeisiin täytyi luoda selkeät kuvat, ja ne päätettiin tehdä Adobe Photoshopin avulla. Sillä saa ai-

kaiseksi hienon näköistä grafiikkaa ja se on ennestään tuttu työkalu. Ohjelmistoon tehdyt uudet paneelit ja painikkeet haluttiin tehdä lumi- ja jäärakennusohjelmiston tyyliin sopiviksi. Siksi niiden luomiseen päätettiin käyttää Unity 3D -projektiin asennettua iGUI-lisäosaa. Ohjelmiston aikaisemmat paneelit ja painikkeet on luotu lisäosan avulla, ja uusista paneeleista ja painikkeista haluttiin saada samantyylliset kuin vanhoista. Jotta kaikki paneelit olisivat yhtenevän näköisiä, täytyi niiden tuottamiseen käyttää samoja elementtejä, kuten painikkeiden kuvakkeita. iGUI-lisäosalla on kätevää luoda uusia käyttöliittymäelementtejä, toisin kuin käyttämällä Unity 3D:n omaa GUI-luokkaa. iGUI-lisä mahdollistaa sen, että paneelit ja painikkeet voidaan tehdä suoraan Unity Editorin avulla, eikä koodiin tarvitse kirjoittaa muuta kuin painikkeiden toiminta.

Asiakkaan kanssa pohdittiin toimintojen suorittamisjärjestystä, eli milloin käyttäjän olisi mahdollista aloittaa rakennuksen sisustaminen. Päätettiin, että lumi- ja jäärakennusten sisustamisen voi aloittaa heti, kun yksi rakennus on luotu. Lisätty rakennusobjekti voidaan saman tien sisustaa, ja sisustettuaan käyttäjällä on mahdollisuus jatkaa rakennuksien luomista. Toimintoja ei haluttu asettaa suorittamisjärjestykseen, joka olisi rajoittanut käyttömahdollisuuksia. Esimerkiksi vaihtoehtona olisi ollut tapahtuma, jossa käyttäjän täytyisi rakentaa ensin kaikki halutut lumirakennukset voidakseen sisustaa niitä. Tavoitteena oli, että käyttäjä saa itse valita missä järjestyksessä haluaa toiminnot suorittaa.

Sisustussuunnitteluohjelmistojen tutkimisen jälkeen oli mahdollista ideoida sisustussuunnittelussa tarvittavia toimintoja. Sisustustoimintoihin päätettiin luoda sisustuskappaleiden lisääminen ja poistaminen, jotka ovat olennaiset tapahtumat sisustamiselle. Näille toiminnoille aiottiin luoda omat tapahtumankäsittelyt. Lisäksi joihinkin sisustuskappaleisiin oli tarkoitus lisätä liikuttamismahdollisuus. Liikuttaminen voisi tapahtua raahaamalla valittua objektiä. Sisustusobjekteja on myös mahdollisuus kääntää lisäämisen jälkeen. Liikuttamista haluttiin rajata, jotta käyttäjä ei pystyisi tekemään mahdottomia tekoja, kuten viemään sisustusobjektin ulos huoneesta. Jokaista uutta toimintoa on mietitty käyttäjän näkökulmasta, ja ne on pyritty luomaan mahdollisimman sujuviksi ja käyttäjäystävällisiksi.

Ohjelmiston sisustusobjekteihin harkittiin liitettäväksi myös valoja. Sisustusobjektien valaistukseen ajateltiin käyttää Unity 3D:n omia valo-objekteja, joilla saa nopeasti lisättyä valaistuksen peliobjektiin. Valoja ei aiottu lisätä joka sisustusobjektiin mahdollisen, sillä se olisi saattanut kuormittaa ohjelmaa.

Ideoitiin, että sisustaminen ei olisi täysin vapaata, eli käyttäjä ei voi valita mitä tahansa sisustusobjekteja rakennukseen. Sisustamisesta haluttiin tehdä nopea prosessi. Jotkin rakennukset ovat pienikokoisia, joten niihin ei voida asettaa montaa sisustusobjektia. Siksi on parempi rajata sisustuselementtien lisäämistä estämällä täysin vapaa sisustaminen. Rakennuksiin päätettiin luoda valmiit huonevaihtoehdot, joita käyttäjä pystyy luomaan erilaisten sisustusobjektien avulla. Sisustusmallit luotiin huoneideoihin sopiviksi. Toteutettavat huoneet ja uudet mallit suunniteltiin yhdessä asiakkaan Antti Yliharjun kanssa. Päätettiin luoda tavanomaisesta poikkeavat huonevaihtoehdot kuin yleensä lumirakennuksissa on, kuten muun muassa makuuhuone ja kirkko. Huonevaihtoehdoiksi päätyivät teatteri, projisointitila sekä baari. Haluttiin luoda sellaiset tilavaihtoehdot, joissa voisi hyödyntää samoja sisustuselementtejä. Näin saadaan huoneet muutettua nopeasti toiseen käyttötarkoitukseen vain vaihtamalla yksi sisustuselementti. Jos vaihtaa esimerkiksi projisointiseinän lamaan, muuntuu projisointitila saman tien teatteriksi. Tilojen käyttötarkoitusta ei ole rajattu suunniteltuihin huonevaihtoehtoihin, vaan käyttäjä saa itse päättää millaiseen tarkoitukseen tila suunnitellaan. Jokainen sisustuselementti on mahdollista asettaa samaan tilaan kaikkien elementtivaihtoehtojen kanssa. Käyttäjällä on valta sisustaa sellaisen tilan kuin haluaa annettujen sisustuselementtien avulla.

Asiakkaan toiveena oli myös, että ohjelmistoon tehtäisiin kuvalliset ohjeet ohjelmistojen toiminnoista. Ohjeet tulisi asettaa niin, että ne ilmestyisivät ruutuun, kun ohjelmisto käynnistyy. Ohjelmisto sisältää jo ohjeet, mutta niitä haluttiin tarkentaa ja päivittää. Käyttäjä voisi katsoa ohjeista kuinka ohjelmiston toiminnot suoritetaan ja eri painikkeiden selitykset.

6 SISUSTUSSUUNNITTELUOHJELMISTON TOTEUTUKSEN KUVAUS

6.1 Sisustus toiminnon liittäminen ohjelmistoon

6.1.1. Sisustaminen 2D-tilassa

Tutkittaessa erilaisia sisustussuunnitteluohjelmistoja havaittiin niiden sisältävän kaksi erilaista tilaa. 2D-näkymä, joka useimmissa toimi sisustustilana ja 3D-tila, jossa sisustettuja huoneita pystyi katsomaan eri suunnista ja kulmista. Lumi- ja jäärakennussovelluksessa ovat vastaavanlaiset tilat. Sisustus-toimintoa suunniteltaessa huomattiin, ettei ole tarvetta toteuttaa erillistä kolmatta tilaa, sillä sisustaminen voisi tapahtua hyvin molemmissa tiloissa. Täysin uusi tila olisi ollut hankala liittää osaksi valmista ohjelmistoa. Monet ominaisuudet ovat riippuvaisia näistä kahdesta näkymästä, joten ohjelmistoon olisi täytynyt tehdä huomattavia muutoksia. Täten sisustustoiminto päätettiin luoda toiseen näistä tiloista.

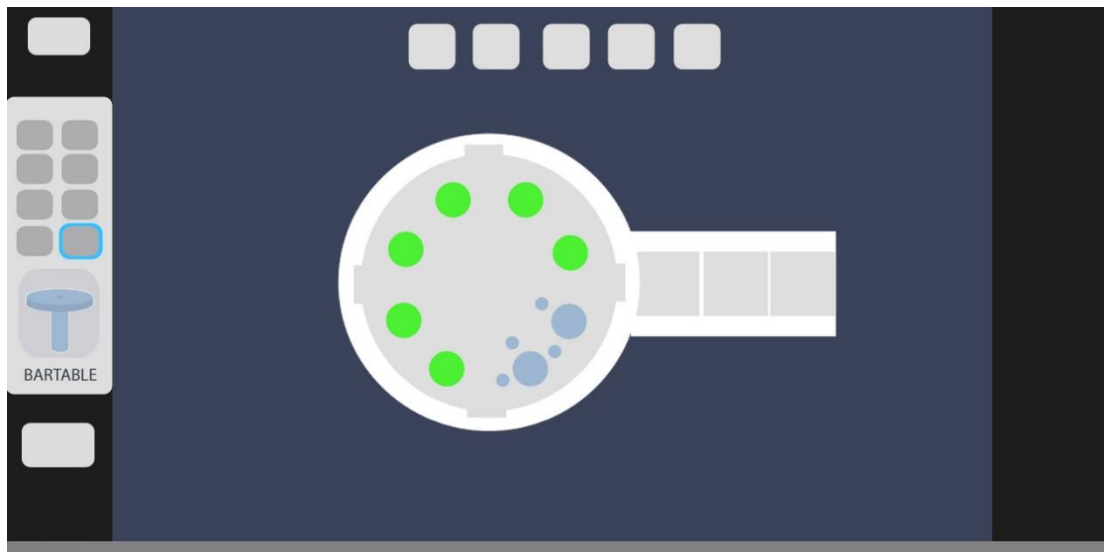
Ensimmäisenä ajatuksena oli sijoittaa sisustusominaisuus osaksi 2D-näkymää. Sisustaminen haluttiin kuitenkin erottaa selkeästi rakennustoiminnasta, jotta käyttäjä saa keskittyä yhteen asiaan kerralla. Sisustaminen ja rakentaminen samassa tilassa olisi voinut hankaloittaa käyttöliittymän käyttöä, sillä molempien toiminnot olisi täytynyt näyttää ruudulla. Ohjelmistoa olisi mahdollisesti ollut epämiellyttävä käyttää, jos käyttöliittymä olisi sisältänyt paljon erilaisia paneeleja ja painikkeita. Tilat olisi joka tapauksessa pitänyt erottaa toisistaan, jotta käyttäjä olisi huomannut selvästi rakennus- ja sisustamistoimintojen eron. Käyttäjälle olisi täytynyt kertoa, mitkä elementit ovat ulkoelementtejä ja mitkä voi laittaa sisälle.

2D-näkymässä sisustustilaan vaihtuessa rakennukset olisi täytynyt kiinnittää paikalleen. Rakennuksen sisustaminen helpottuu, kun rakennukset pysyvät paikallaan, eikä käyttäjän tarvitse pelätä siirtävänsä niitä vahingossa. Sisustustilaan mentäessä rakennustoimintopit olisivat vaihtuneet sisustustoimintoihin. Näin olisi varmistettu se, ettei käyttäjä voi tehdä rakentamiseen kuuluvia toimintoja eikä muutoksia itse rakennukseen. Käyttäjä pystyy jatkamaan rakennusten luomista vaihtamalla tilan takaisin rakennusnäkömään.

2D-tilassa sisustaminen mahdollistaa niin sanotun vapaan sisustamisen. Tällöin käyttäjä pystyisi laittamaan sisustusobjektin mihin tahansa kohtaan huonetta. Vapaasta sisustamisesta luovuttiin jo suunnitteluvaiheessa. Käyttäjän

toimintoja haluttiin rajata, ja tehdä sisustamisesta nopeaa. Sisustaminen on käyttäjän luomien rakennusten ehostamiskeino.

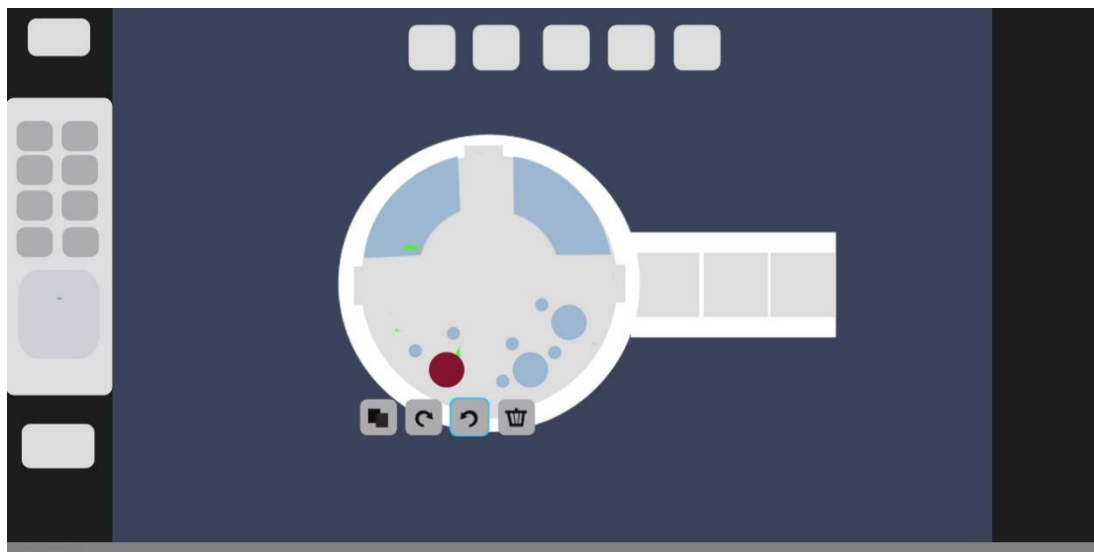
Sisustuselementeille on määrätty valmiiksi paikat rakennuksissa, johon ne voidaan asettaa. Tämä nopeuttaa itse sisustamista, kun käyttäjä ei voi asettaa jokaista objektia haluamaansa paikkaan. Tällä myös rajataan sisustusobjektien lisäämistä, sillä käyttäjä ei pysty loputtomasti lisäämään uusia objekteja. Objektien esimääritetyllä paikkavalinnalla estetään objektien päällekkäisasettelu ja helpotetaan lisäystoiminnon saavuttamista. Sisustusobjekteille päätettiin luoda esimääritetyt paikat jo ensimmäisessä suunnitteluversiona. Käyttäjä olisi valinnut sisustusobjektin valikosta, ja rakennukseen olisi ilmestynyt kyseisen objektin paikkavaihtoehdot. Kuvio 14 on ensimmäisen suunnitteluversion ideakuva. Kuvassa näkyy rakennusobjekti ja sen sisältämät esimääritetyt paikat. Rakennuksen sisällä olevat vihreät ympyrät ovat paikkavaihtoehtoja, johon valitun objektin olisi voinut asettaa.



Kuvio 14. Ensimmäinen suunnitteluversiono sisustuselementtien lisäämisestä 2D-tilassa

2D-näkymän varjopuolena on syvyyksien uupuminen näkymästä. Käyttäjä ei pysty näkemään eri kappaleiden korkeuseroja, joten hänen on vaikea päätellä, minkä korkuisia elementit ovat. Lisäksi 2D-näkymässä ei ole mahdollista asettaa kappaleita päällekkäin, joten esimerkiksi pöydän yläpuolelle ei saisi lisättyä mitään. Seinille olisi mahdotonta kiinnittää mitään, sillä käyttäjä ei näkisi mille korkeudelle objekti asettuisi. Seiniin lisättävät objektit ovat aika ohuita, joten niitä olisi hankala asettaa mihinkään tai saada niistä otetta.

Myös monet muut sisustusobjektit ovat pienikokoisia, joten näkymän zoomaustoiminto olisi täytynyt muuttaa.



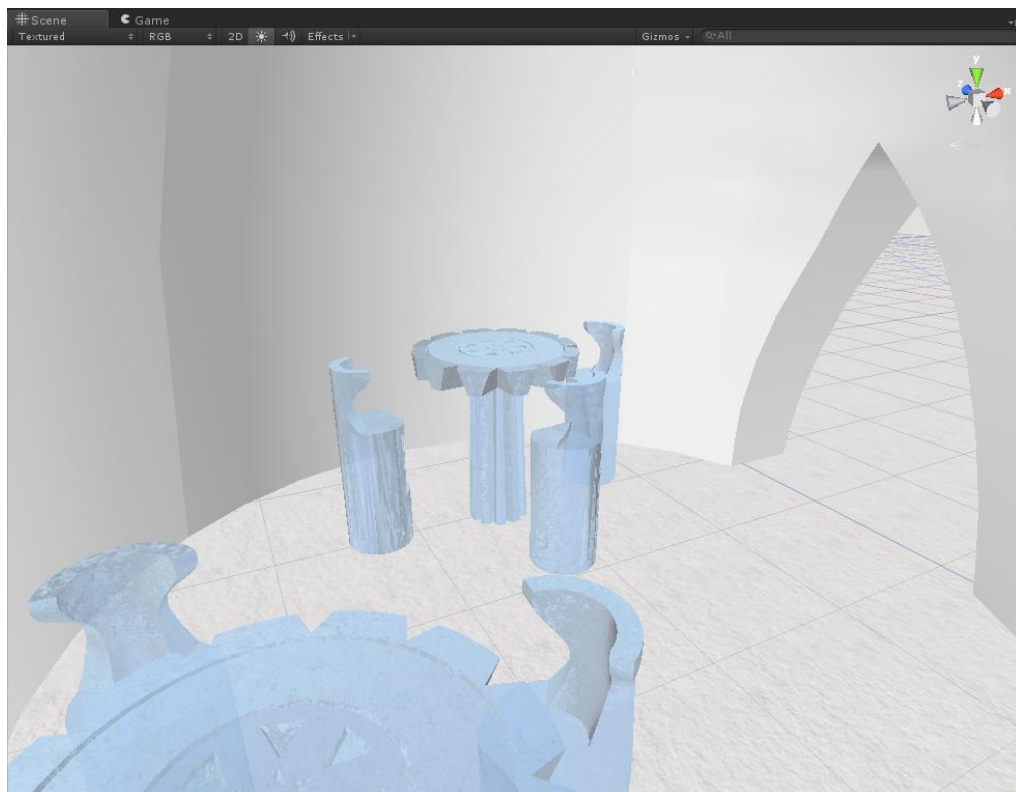
Kuvio 15. Ensimmäinen suunnitteluversio sisustusobjektin valitsemisesta

Kuviosta 15 näkyy ensimmäinen suunnitteluversio sisustusobjektin valitsemisesta 2D-tilassa. Sisustuselementtiä olisi voinut liikuttaa vetämällä sitä, ja kääntämään painamalla kääntöpainikkeita. Myös sisustusobjektin poistaminen olisi ollut mahdollista. Valitun sisustusobjektin olisi pystynyt monistamaan eli luomaan täsmälleen samanlaisen objektin, jolloin käyttäjän ei olisi tarvinnut valita sisustusobjektivalikosta kyseistä objektia toista kertaa. Tämä toiminto olisi nopeuttanut sisustamista.

6.1.2 Toteuttaminen 3D-tilassa

Sisustuselementtien lisääminen päätettiin lopulta toteuttaa 3D-tilassa. Ensimmäisenä ajatuksena syntyi luoda tilaan uusi kamerakulma. Kameralla olisi pystynyt näkemään rakennuksen seinien läpi, ja sisustamaan niitä ulkoapäin. Lähimmäinen seinä olisi ollut läpinäkyvä. Kuvio 16 on esitys tästä näkymästä miltä se olisi mahdollisesti näyttänyt. Ideasta päätettiin luopua, sillä se olisi ollut vaikea toteuttaa nykyisiin mallikappaleisiin. Mallien materiaaleihin olisi pitänyt luoda uusi pintatoiminto (shader), joka mahdollistaisi peliobjektin muuttamisen läpinäkyväksi. Kameraan lähimpänä ollut rakennusobjekti olisi asetettu läpinäkyväksi, jottei se olisi haitannut näkyvyyttä. Takaseinä ja taajimpana olleet objektit puolestaan olisivat olleet näkyvissä. Kameran ja rakennusobjektin etäisyyksiä olisi täytynyt laskea koko ajan. Tilaan olisi pitänyt asettaa tarkat määritykset milloin kappale on läpinäkyvä ja milloin ei, sekä

luoda tarkat rajat mistä läpinäkyvyys alkaa ja mihin se loppuu. Tämä olisi ollut haastavaa toteuttaa, joten päätettiin miettiä parempi keino sisustamisen toteutukseen 3D-tilassa.



Kuvio 16. Sisustamisnäkyminen seinän läpi

Lopulta sisustamismahdollisuus päätettiin liittää osaksi kävelytilaa. Myös tässä tilassa haluttiin sisustaminen erottaa selkeästi muista toiminnoista, joten sille luotiin ”oma” näkymä. Ohjelmiston päävalikkoon luotiin uusi painike sisustustilan vaihtamiselle. Sisustuspainikkeesta saadaan sisustamistoiminnot näkyviin. Painike on oletuksena passiivisena, sillä käyttäjä ei voi sisustaa ulkotilaa ja sisustaminen kohdistuu vain sisätiloihin. Painike muuttuu aktiiviseksi, kun käyttäjä lisää rakennuksen. Tämän jälkeen käyttäjä voi valita sisustusnäkyvän ja aloittaa rakennuksen sisustamisen. Käyttäjän ei ole pakko sisustaa, jos hän ei halua, sillä sisustaminen on tehty rakennuksien ehostamiseksi.

Sisustamistoiminnon ollessa aktiivisena rakennuksiin ei voida klikata valoja päälle. Myös päinvastoin estetään sisustaminen, kun käyttäjä haluaa asettaa rakennuksiin valot päälle tai katsoa aluetta lentokameralla. Näin ei häiritä eri toimintojen käyttöä ja vältetään virhetilanteita, esimerkiksi tapahtumasta, jos-

sa valon värivalinta tulee vahingossa aktiiviseksi käyttäjän koskettaessa rakennuksen seinää.

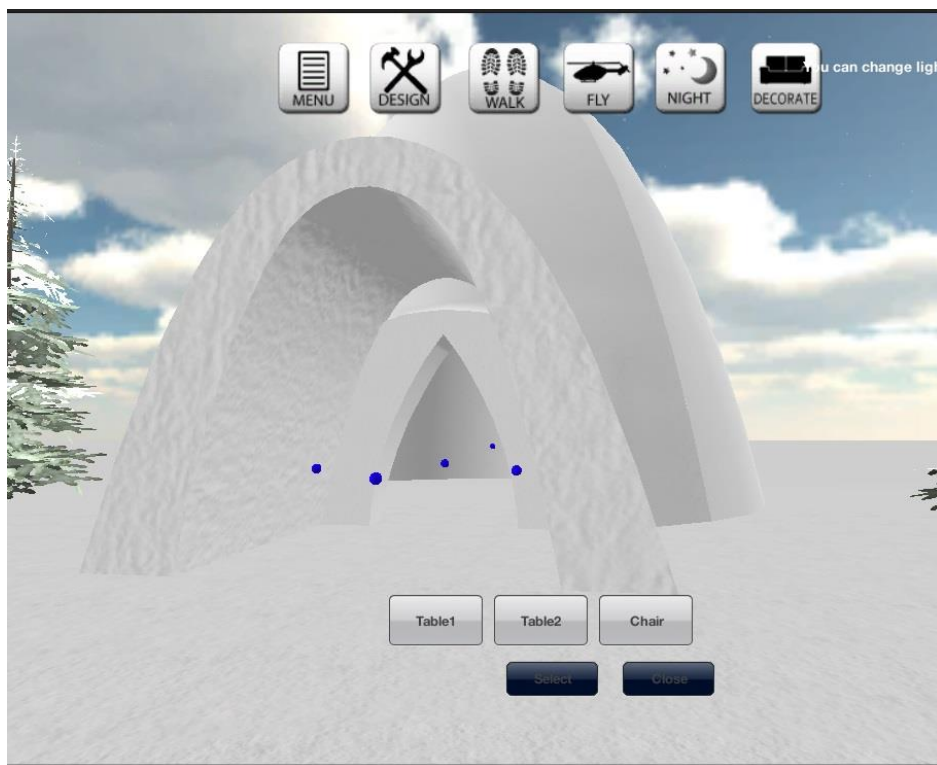
Kappaleiden syvyydet ja korkeudet ovat helpommin havaittavissa 3D-tilassa, joten käyttäjä pystyy arvioimaan sisustusobjektien ja tilojen suuruudet. Käyttäjä näkee suoraan millaisia sisustuselementit ovat. Jos sisustaminen tapahtuisi 2D-tilassa, käyttäjän täytyisi vaihtaa kävelytilaan nähdäkseen sisustamansa tilat tarkemmin. 3D-tilassa käyttäjä näkee sisustuselementit samantien. Näin nopeutetaan myös ohjelmiston käyttöä, kun käyttäjän ei tarvitse kulkea edestakaisin kahden näkymän välillä.

Objektien liikuttaminen 3D-tilassa on käyttäjälle haasteellisempaa kuin 2D-tilassa, sillä etäisyyksien hahmottaminen ei ole helppoa 3D-maailmassa. Tätä ominaisuutta on helpotettu estämällä sisustusobjektien täysin vapaa liikuttaminen. Samalla estetään objektien päällekkäisasettelun mahdollisuus, joka vähentää käyttäjän sopimattomien toimintojen saavuttamista. Muutamia sisustuskappaleita pystyy rajoitetusti liikuttamaan sisätilassa.

6.2 Sisustusobjektien lisäysominaisuuden kehittäminen

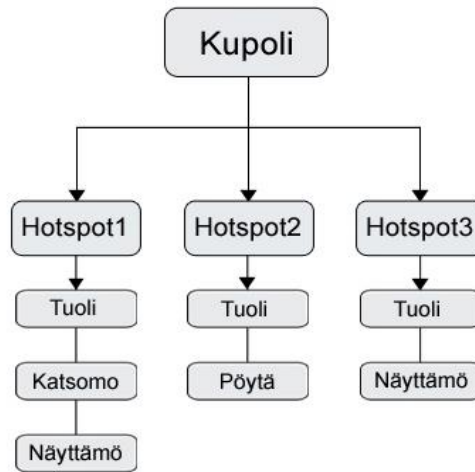
Sisustusobjektien lisäämiseen täytyi kehittää jokin toimiva ja käytännöllinen ratkaisu, joka istuisi hyvin 3D-ympäristöön. Rakennuksiin päätettiin luoda esimääritetyt paikat, joihin sisustusrakenteet kiinnitetään niiden lisäyksen yhteydessä. Jokainen rakennusobjekti sisältää paikkaobjekteja (hotspot), jotka ilmestyvät käyttäjälle näkyviin hänen alkaessaan sisustaa. Paikkaobjekti näyttää käyttäjälle, mihin on mahdollista asettaa sisustuselementti. Käyttäjän klikatessa paikkaobjektia näkyviin ilmestyy valikko sisustuselementeistä, jotka on mahdollista lisätä kyseiseen kohtaan. Joka paikkaobjektilla on omat sisustusobjektinsa, sillä kaikkia luotuja sisustusobjekteja ei ole mahdollista lisätä jokaiseen rakennukseen. Osa sisustusobjekteista ovat kooltaan liian suuria joihinkin rakennusmallien tiloihin. Kuviossa 17 näkyvät objektin paikkaobjektit. Kuvassa paikkaobjektia on klikattu, ja näkyviin on ilmestynyt valintapaneeli, jossa on paikan sisustuselementtivalikoita. Sisustusobjektien lisäämistä täytyi rajoittaa niin, etteivät kaikki sisustuselementtivalikoita ole koko ajan näkyvillä. Näin saadaan myös vaihtelevuutta näkymään, kun kaikkia rakennuksia ei ole mahdollista sisustaa täysin samanlaisiksi. Tällä keinol-

la karsittiin myös mahdollisuus tehdä mahdottomia lisäyksiä, kuten asettaa baarituoli seinään kiinni.



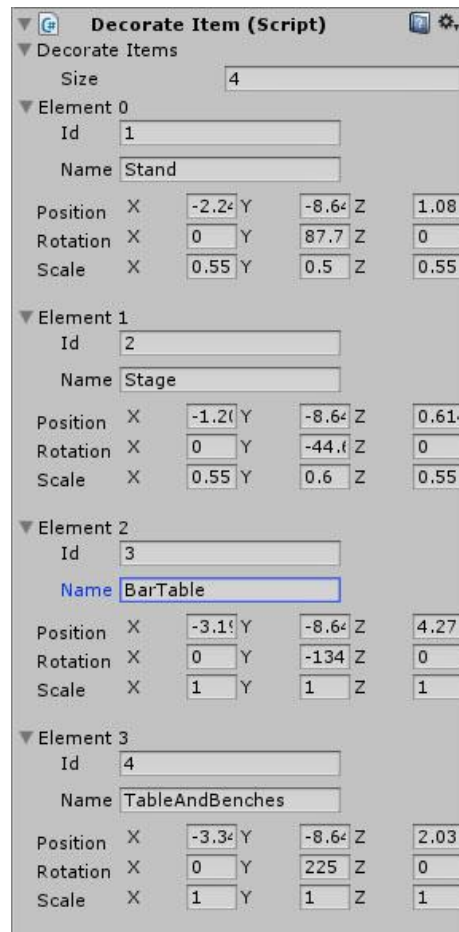
Kuvio 17. Paikkaobjektit on asetettu ja valikkopaneelit ovat näkyvillä

Paikkaobjektit on lisätty rakennusobjektien lapsiobjekteiksi Unity Editorin kautta. Aluksi myös paikkaobjekteille asetettiin tyhjiä lapsiobjekteja, joita käytettiin hyödyksi sisustusobjektin lisäämisessä oikeaan kohtaan. Lapsiobjektit sisälsivät koordinaatit sisustusobjektin paikasta rakennuksessa. Lapsessa oli myös tieto mihin suuntaan sisustusobjekti käännetään ja kuinka suureksi se skaalataan. Kuvio 18 näkee perintäkaavion rakennus-, paikka- ja sisustusobjekteista. Tyhjät peliobjektit nimettiin sisustusobjektien mukaisesti, jotta osattiin yhdistää sisustusobjektit suoraan lapsiobjekteihin nimen perusteella.



Kuvio 18. Rakennuksen paikkaobjektien ja niiden sisustusobjektien perintäkaavio

Paikkaobjektien tyhjästä peliobjekteista päätettiin luopua, sillä keksittiin parempi keino välittää sisustusobjektien tiedot paikkaobjektille. Sisustusobjektien tiedot päätettiin lisätä listaan, joka välitetään paikkaobjektille. Jokaiseen paikkaobjektiin on lisätty skripti, jonka ansiosta objektille voidaan luoda luettelo sisustusobjektivaihtoehdoista. Lista sisältää tiedot sisustuselementeistä: tunnistusluvun (id), nimen, kohdan rakennuksessa, kiertokulman ja skaalauksen. Listan elementit voidaan asettaa Unity Editorin Inspector-ikkunan kautta, jolloin lähdekoodiin ei tarvitse luoda listoja ja elementtejä. Unity Editoria on muutettu niin, että sen kautta on mahdollista lisätä elementtejä listaan, joka ei ole oletuksena mahdollista. Kaikki listat olisi täytynyt luoda ja alustaa lähdekoodissa, joka olisi vienyt aikaa, ja sitä olisi epämiellyttävä ylläpitää. Unity Editorin avulla helpotetaan ja nopeutetaan listan luomista syöttämällä kenttiin arvoja. Myös uusien sisustusobjektien ohjelmistoon tuominen helpottuu, kun lähdekoodiin ei tarvitse tehdä muutoksia ja lisäyksiä. Kuvio 19 näkyy kupoliin luodun listan elementit ja tiedot.



Kuvio 19. Kupolin paikkaobjektiin luotu lista

Jotta elementtien listaan lisäys onnistuisi Unity 3D:n avulla, täytyi sen Editoriin tehdä lisäyksiä. Unity Editorin Inspector-ikkuna sisältää tekstikentät, otsikot ja useita muita visuaalisia elementtejä, joiden avulla muokataan muun muassa peliobjekteja. Täytyi kirjoittaa skriptejä, joissa esiteltiin listaan lisättävät elementit ja niiden tekstikentät, jotka ilmestyivät Inspector-ikkunaan. Luodut skriptit täytyi asettaa Unity 3D -projektin Editor-kansioon. Tällöin muutokset tulivat automaattisesti Inspector-ikkunaan näkyviin. Lisättävät kentät tuli määrittellä niin, että kenttiin syötettävät tiedot osattiin yhdistää listan muuttujiin. Lista sisältää erityyppisiä muuttujia, joten luotiin skripti, jossa esiteltiin listan muuttujat: nimi, paikka, rotaatio ja skaalaus. Luotua skriptiä käytetään listan tietotyyppinä. Kuviossa 20 näkyy luotu skripti, josta kerrotaan listan muuttujat.

```

1
2 using UnityEngine;
3 using System.Collections;
4 using System;
5 using iGUI;
6
7 [Serializable]
8 public class DecoItem{
9
10     public int id;
11     public string name;
12     public Vector3 position;
13     public Vector3 rotation;
14     public Vector3 scale;
15
16
17 }
18

```

Kuvio 20. Listan elementit luodussa skriptissä

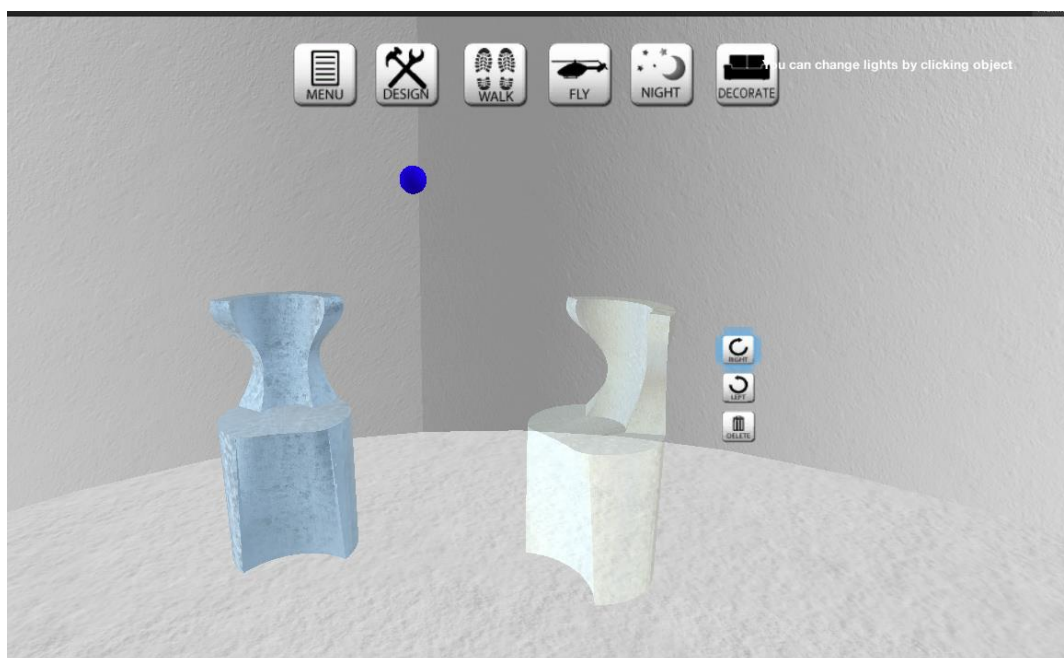
Paikkaobjektit täytyy olla tunnistettavissa käyttäjän klikatessa niitä. Paikkaobjektin tunnistamiseen käytettiin Raycast-toimintoa, jonka osuessa peliobjektin collideriin katsottiin osutun peliobjektin tagi. Paikkaobjekteille on asetettu hotspot-tagit, joista tiedetään objektien kuuluvan paikkaobjekteihin.

6.3 Sisustusobjektien toiminnallisuudet

Useita peliobjekteja käsiteltäessä on hyödyllistä käyttää niiden ryhmittelytoimintoa. Näin varmistetaan objektien riippuvaisuus toisista objekteista ja täten helpotetaan toimintoja. Sisustusobjektien asettelussa hyödynnettiin ryhmittelyä. Aina kun lisätään uusi sisustuselementti, se tulee kuuluvaksi rakennusobjektille, jonka sisällä sisustaminen tapahtuu. Täten sisustuselementti tulee riippuvaiseksi rakennusobjektista, ja jos rakennusta siirretään, liikkuu sisustuselementti sen mukana. Tämä helpottaa ja vähentää käyttäjän turhia toimintoja, sillä hänen ei tarvitse huolehtia erikseen muun muassa sisustuselementtien poistamisesta ja liikuttamisesta, jos toiminta kohdistuu itse pääobjektiin eli rakennukseen.

Käyttäjän liikkuessa sisustetun rakennuksen sisällä, hän voi valita luotuja sisustusobjekteja klikkaamalla niitä ja tehdä objekteille toimintoja. Objektien valitseminen on vain mahdollista sisustustilan ollessa päällä. Klikattaessa sisustuselementti muuttuu aktiiviseksi, ja uusi toimintavalikko ilmestyy näkyviin. Aktiivisen sisustusobjektin materiaalin väri muutetaan keltaiseksi, jotta

valittu objekti erotettaisiin helpommin. Kuviossa 21 nähdään aktiivinen objekti ja sen oikealla puolella toimintavalikko.



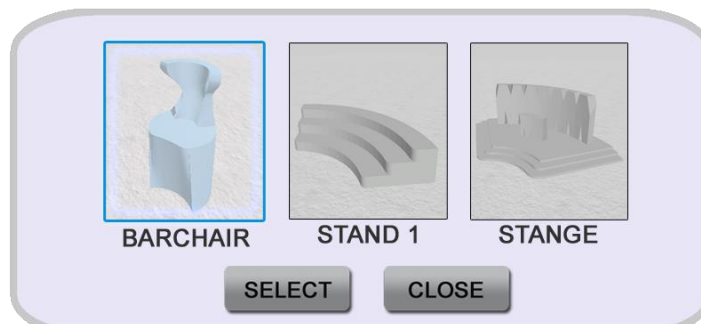
Kuvio 21. Valittu sisustusobjekti ja sen toimintavalikko

Osa valituista objekteista on mahdollista liikuttaa. Esimerkiksi tuolin paikkaa voi muuttaa tarttumalla siitä kiinni ja vetämällä sitä. Kaikille sisustusobjekteille ei voitu tehdä liikuttamismahdollisuutta, sillä jotkin objektit ovat hyvin suuria ja niitä on haastavaa liikutella. Sisustusobjektin liikuttelu tapahtuu hiirtä vetämällä. Objekti pysyy hiiren keskipisteessä niin kauan kunnes klikataan uudestaan. Objektin liikuttelua on rajattu määrittelemällä alue, jonka sisäpuolella on vain mahdollista muuttaa objektin paikkaa. Alue on ympyrän muotoinen, ja sen keskipiste on valitun sisustusobjektin paikkaobjekti. Ympyrän suuruuteen on asetettu arvo, jota voidaan halutessa muuttaa. Objektia liikuteltaessa, tutkitaan hiiren etäisyyttä keskipisteeseen koko ajan. Jos hiiri poikkeaa ympyrän ulkopuolelle, valittu objekti liikkuu ympyrän kehällä. Alueen avulla halutaan estää sisustuselementin vieminen ulos tilasta, sillä sisustusobjektit kuuluvat vain sisätiloihin. Objektin paikan muuttamista on rajattu myös niin, ettei objektia ole mahdollista viedä seinän läpi. Objekteja pystyy asettamaan sisäkkäin, sillä sitä ei ole estetty. Päällekkäisasettelu olisi voitu estää esimerkiksi käyttämällä ylimääräisiä collidereita. Tätä ominaisuutta ei ehditty testata.

6.4 Uudet valikko-ominaisuudet

6.4.1 Sisustusobjektin valintavalikko ja -painikkeet

Käyttäjän valitessa sisustettavan kohdan rakennuksessa ilmestyy hänelle näkyviin valintavalikko sisustusvaihtoehdoista. Oletuksena valintavalikko on näkymätön ja ilmestyy näkyviin, kun paikkaobjektia on klikattu. Jokainen paikkaobjekti sisältää listan sisustusobjekteista, jotka pystytään lisäämään kyseiseen kohtaan. Sisustusvaihtoehdot generoituvat valintapaneeliin valintapainikkeiksi. Jokaiselle sisustusobjektille on luotu valmiiksi valintapainikkeet iGUI-lisäosan avulla. Kaikki luodut valintapainikkeet eivät ilmesty näkyviin kerralla. Ensin tutkitaan mitä paikkaobjektia on klikattu, sitten mitä sisustusobjekteja se sisältää listan perusteella. Listasta valitaan nimen perusteella oikeat sisustusobjektien painikkeet valintavalikkoon. Täten käyttäjällä on vain näkyvillä ne vaihtoehdot, jotka on mahdollista lisätä rakennuksen kohtaan. Kuvioista 22 näkyy suunniteltuversio valintavalikosta.

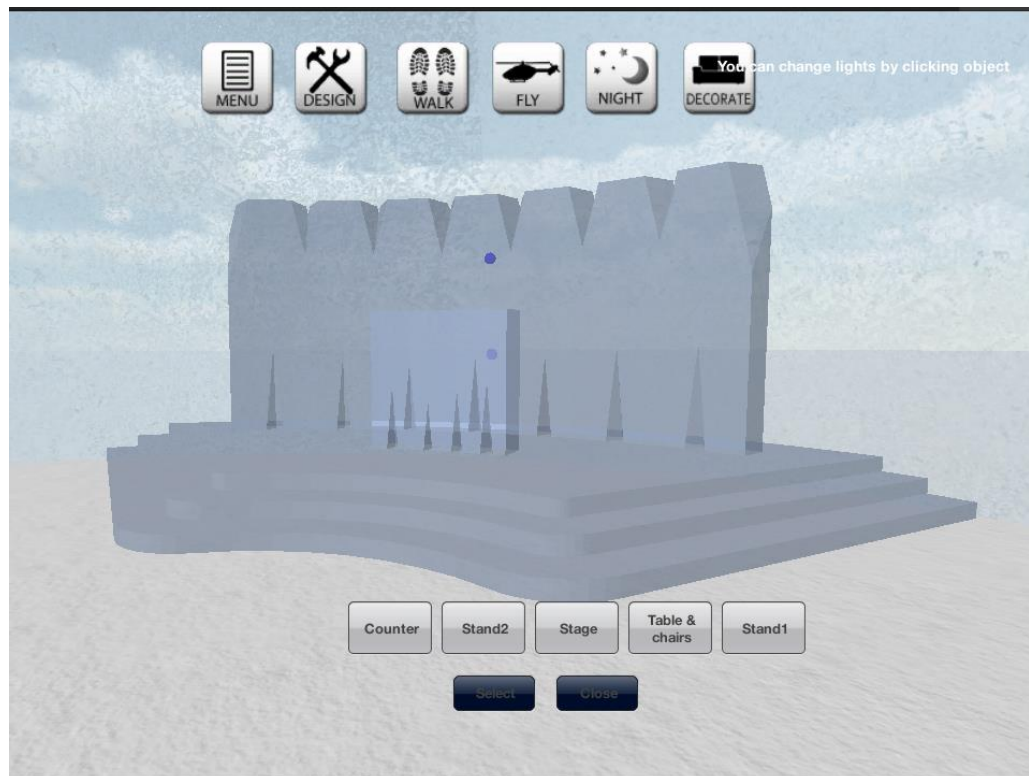


Kuvio 22. Suunnitelmakuva sisustuselementin valintavalikosta

Käyttäjän klikatessa jotain valintapainiketta luodaan kyseisen painikkeen sisustusobjekti pelimaailmaan. Sisustusobjekti on niin kutsutussa esikatselutilassa eli sitä ei ole lopullisesti lisätty. Vasta käyttäjän hyväksyessä objektin, painamalla valintavalikon Select-nappia, sisustusobjekti jää pysyvästi pelimaailmaan. Painettaessa toista valintapainiketta, edellisen painikkeen luoma sisustuselementti katoaa ja uuden painannan objekti ilmestyy tilalle. Näin pystytään esikatselamaan kaikki vaihtoehdot ennen lopullista valintaa. Käyttäjällä annetaan mahdollisuus perua sisustusobjektin lisäämisaikomus sulkemalla valintavalikko Close-painikkeella. Sisustusobjektin valitsemisen jälkeen paikkaobjekti piilotetaan ja sen collider asetetaan pois päältä. Tällöin käyttäjä ei pysty lisäämään uutta objektiä ennen kuin vanha on poistettu.

6.4.2 Haamuobjekti

Sisustuselementtien esikatselussa luodaan pelimaailmaan näkyviin peliobjekti. Tämä on niin kutsuttu haamuobjekti (ghost), eli oikea sisustusobjekti, jolla on vain eri shader, materiaali ja väri. Materiaalia on myös hieman himmennetty, jolloin objekti muuttuu läpinäkyväksi. Kuviossa 23 näkyy luodun objektin haamuobjekti. Haamuobjekti jätetään pelikenttään tai tuhotaan riippuen käyttäjän toiminnoista.



Kuvio 23. Näyttämömallin haamuobjekti

Käyttäjän painaessa valintavalikon painikkeita luodaan aina uusi sisustusobjekti ja edellinen tuhotaan. Sisustusobjekti jätetään pelikenttään vasta kun käyttäjä hyväksyy sen. Tämän jälkeen sisustusobjektille palautetaan sen alkuperäiset materiaaliominaisuudet. Tiedot materiaaleista haetaan luodusta skriptistä, joka on lisätty skenessä tyhjäan peliobjektiin. Kyseiseen skriptiin on esitelty sisustusobjekteissa hyödynnettäviä ominaisuuksia, kuten materiaaleja ja värejä. Tyhjäan peliobjektiin on asetettu arvot Unity Editorin avulla. Luodulle sisustusobjektille lisätään myös DecorateObject-tagia, joka kertoo sen olevan sisustusobjekti.

6.4.3 Aktiivisen sisustusobjektin toimintavalikko

Klikattaessa lisättyä sisustusobjektia tämä muuttuu aktiiviseksi, ja tälle ilmestyy näkyviin toimintavalikko. Aktiivisen sisustusobjektin toimintovalikko sisältää painikkeet sisustusobjektiin koskevista toiminnoista. Eri sisustusobjekteilla on erilaisia suoritettavia toimintoja. Skriptissä kerrotaan, mitä toimintoja jokaiselle sisustusobjektille voidaan tehdä ja sen mukaan asetetaan oikeat toimintapainikkeet näkyviin. Muun muassa kaikkiin sisustusobjekteihin ei ole mahdollista asettaa valoja päälle, sillä vain muutamat objektit sisältävät valot.

Toimintavalikon poistopainikkeella käyttäjällä on mahdollisuus poistaa valittu objekti. Poistamisen jälkeen kohdan paikkaobjekti ilmestyy takaisin näkyviin ja muuttuu aktiiviseksi. Näin käyttäjälle annetaan mahdollisuus asettaa uusi sisustuselementti poistetun objektin tilalle. Pienikokoisia sisustusobjekteja pystyy myös kääntämään eri suuntaan painamalla toimintavalikon kääntämisspainikkeita.

Tunnelman luomiseksi osaan sisustusobjekteista lisättiin valo-objektit, jotka ovat oletuksena pois päältä. Toimintavalikon valopainikkeella saadaan valot päälle. Valot toteutettiin erilailla kuin rakennusobjektien valojen valinta. Rakennuksiin on luotu erilliset valotekstuurit, jotka asetetaan rakennusobjekteille, kun valot laitetaan päälle. Uusien valotekstuurien toteuttaminen olisi ollut työlästä ja vienyt aikaa, siksi niitä ei ajateltu luoda sisustusobjekteille. Sisustuksen valoihin ei myös ehditty liittää rakennusobjekteihin luotua värin valinta-toimintoa ja -paneelia. Täten päädyttiin käyttämään Unity 3D -ohjelmiston omia valopeliobjekteja. Valo-objektit asetettiin sisustusobjektien lapsiobjekteiksi. Valot ovat objektin sisäpuolella ja niille on vain määrätty yksi väri.

Painikkeiden kuviksi päätyivät rakennustilan painikkeet. Ajateltiin käyttää samoja kuvia, jotta ei aiheutettaisi käyttäjälle hämmennystä uusien painikkeiden avulla.

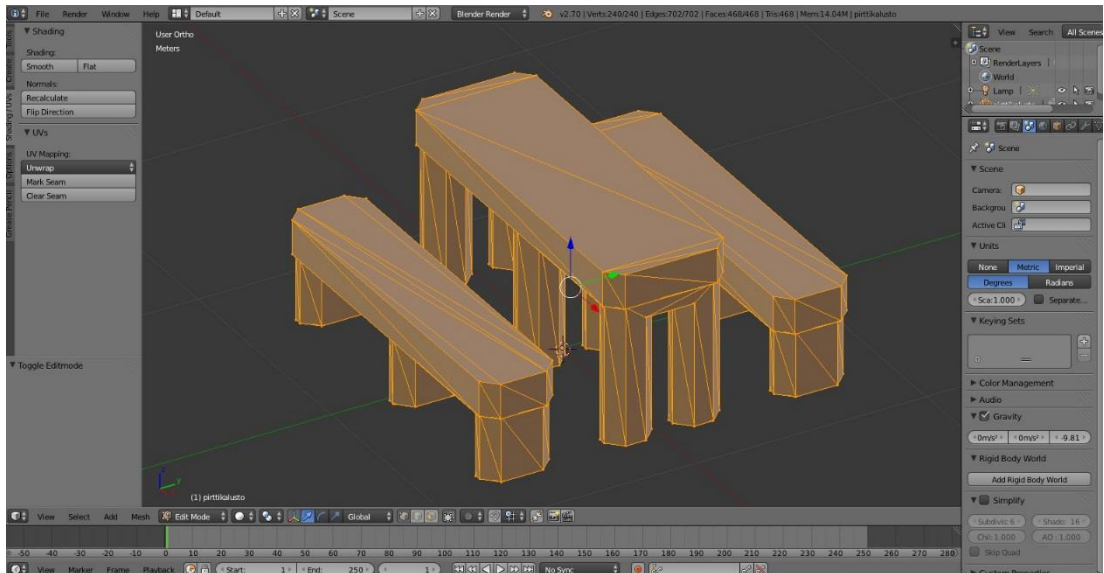
6.5 Sisustusmallit

Asiakas Antti Yliharju toteutti mallit päätetyistä sisustusobjekteista. Asiakkaan luomien mallien tiedostomuoto stl. ei kuulu Unity 3D:n tukemiin tiedostomuotoihin, joten mallit täytyi käsitellä Blender 3D-mallinnusohjelmalla. Blenderiin

voidaan tuoda stl.-tiedostoja, jotka saadaan muutettua fbx.-tiedostomuotoon. Unity 3D tukee tätä tiedostomuotoa.

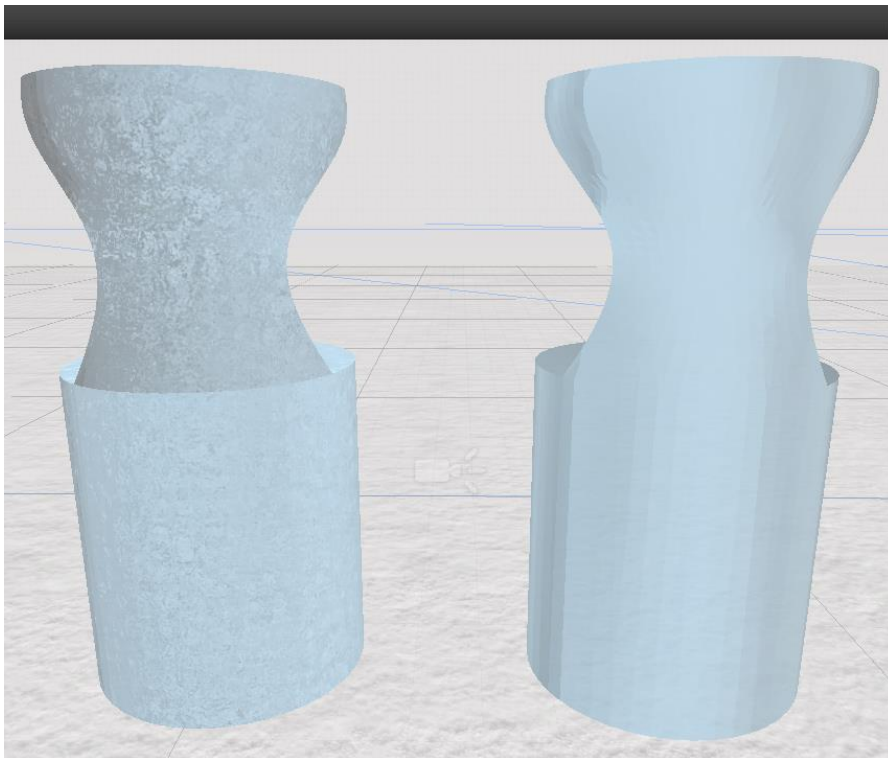
Uudet mallit liitettiin ohjelmistoon viemällä ne Unity 3D -projektin sisältämiin kansioihin. Sitä ennen mallit täytyi muuttaa Blenderillä Unity 3D:n asettamiin formaatteihin. Näin uudet mallit voitiin ottaa heti käyttöön projektissa. Blenderin avulla mallit skaalattiin oikeaan mittasuhteeseen joka akselilla. Myös mittayksikkö muutettiin metreiksi, jota käytetään ohjelmistossa mallien mittayksikkönä. Mallit täytyi myös kääntää oikeiden akseleiden suuntaisesti, niin että ne olisi projektiin vietäessä valmiiksi Unity 3D:n akseleiden suuntaisia. Jokainen mallikappale sisältää keskipisteen, jonka paikkaa voidaan muuttaa mallinnusohjelmilla. Lumi- ja jäärakennusohjelmiston objektien keskipiste on määritelty objektin alareunaan. Tällöin objektin y-akselin arvo on nolla, kun objekti asetetaan maan pinnalle. Myös uusien mallien keskipiste päätettiin siirtää alareunaan keskelle mallia. Keskipiste asetettiin kääntelemällä objektia Blender-työkalulla ja vaihtamalla keskipistettä työkalun avulla.

Saatujen mallien tekstuurit eivät asettuneet oikein mallien pinnalle, joten myös pintojen valitseminen täytyi käsitellä Blender-työkalulla. Mallien pintojen niin kutsuttu UV-mappaus asettui malleihin väärin. UV-mappaus tarkoittaa 2D-grafiikan asettumista 3D-malliin. Kun malliin asetetaan tekstuuri, se asetuu jokaisen valitun pinnan mukaiseksi, eikä huomioi mallin muotoa ollenkaan. Kuviossa 24 näkyy luotu mallikappale, ja kuinka UV-mappaus asettuu sen pintaan. Saatujen mallien UV-mappaus kohdistui kaikkiin mallin rakennettuihin pintoihin. Kuviossa 24 näkyy mallikappale, jonka kaikki pinnat on UV-mapattu.



Kuvio 24. Mallin UV-map ennen käsittelyä

Jotta materiaali asettuisi luonnollisesti malliin, täytyy mallin UV-mappausta muuttaa niin, että pelkät ääriviivat on UV-mapattu. Mallit käsiteltiin niin, että pelkät reunat ja ääriviivat maalattiin ja mallille luotiin uusi UV-map. Tämän jälkeen materiaali asettui malliin halutulla tavalla, ja muutti mallin näkyvyyttä paremmaksi. Kuvio 25 näkee eron materiaalien välillä. Vasemmalla on UV-mapattu sisustusobjekti ja oikealla objekti, jolla on alkuperäinen UV-mappaus. Jokaiselle uudelle mallikappaleelle täytyi luoda uusi UV-map.



Kuvio 25. UV-mapattu sisustusobjekti (vasen) ja alkuperäinen mallikappale

7 POHDINTA

Opinnäytetyön tavoitteena oli luoda lumi- ja jäärakennusohjelmistoon uutta toiminnallisuutta, jonka avulla kehitettiin ohjelmiston toiminnallisuutta ja visuaalisuutta. Tavoitteessa onnistuttiin, ja ohjelmistoon saatiin luotua ja lisättyä sisustustoiminto, jonka avulla käyttäjä voi sisustaa luomiaan lumi- ja jäärakennuksia. Haasteena oli uuden toiminnon luominen vanhaan ohjelmistoon niin, että kaikki osiot toimisivat hyvin keskenään. Myös tässä onnistuttiin. Keksittiin toimiva ratkaisu kuinka toteuttaa rakennuksen sisustustoiminto. Sisustamisesta luotiin helppo ominaisuus, jolla käyttäjä saa nopeasti somistettua rakennelmia. Tähän opinnäytetyössä pyrittiin. Sisustusobjekttilistan ansiosta ohjelmistoon on jatkossa helppo lisätä lisää uusia sisustusobjekteja.

Opinnäytetyössä saatiin halutut sisustustoiminnot toteutettua. Sisustusobjektin liikuttamistoiminto oli haasteellista toteuttaa, sillä peliobjektin liikuttaminen on hankalampaa 3D-maailmassa kuin 2D-näkymässä. Kokeilemalla ja tutkimalla keksittiin viimein keino, miten liikuttaminen saatiin sujuvaksi. Liikuttamistoimintoa olisi voinut vielä enemmän kehittää. Sisustusobjektien valojen vaihto -toiminnon kehittäminen jäi hieman kesken ajan puutteen vuoksi. Myös asiakkaan pyytämiä kuvallisia toimintaohjeita ei ehditty toteuttaa. Ne voisi tehdä ohjelmiston mahdolliseen seuraavaan versioon, sillä ne olisi käyttäjän kannalta hyvä olla olemassa.

Opinnäytetyön aikana ongelmaksi kehittyi kirjoitettavan ja tutkittavan teoriaosuuden aiheen valinta. Aihe oli vaikea päättää ja tämä vei liikaa aikaa opinnäytetyön prosessista. Työn aikana huomattiin kuinka tärkeää on suunnitella aikataulutukset hyvin ja pitää siitä kiinni. Opinnäytetyön sisustustoiminnon kehittäminen oli loppujen lopuksi mukavaa ja tuntuu siltä, että sitä olisi voinut vielä enemmän työstää. Unity 3D -pelimoottorin osaaminen kehittyi ja uusia pelimoottorin ominaisuuksia opittiin prosessin aikana. Myös Blender 3D-mallinnusohjelman käyttö kehittyi entisestään.

Opinnäytetyössä keskityttiin rakennusten sisätiloihin, jatkossa voitaisiin kehittää ohjelmiston rakennusaluetta. Esimerkiksi luomalla keino, jolla ohjelmiston lumirakenteiden rakentaminen ja suunnittelu voitaisiin tehdä keskellä kaupunkia tai markkina-alueella. Myös ohjelmiston rakennus- ja sisustamisobjek-

tien valotoimintoa voisi kehittää eteenpäin. Valotoiminto voitaisiin muuttaa sellaiseksi, että se olisi helppo liittää ohjelmistoon lisättäviin uusiin objekteihin. Olisi mukavaa, jos lumi- ja jäärakennusohjelmistoa kehitettäisiin vielä eteenpäin, sillä ohjelmistosta löytyy potentiaalia vaikka mihin käyttötarkoitukseen.

LÄHTEET

- Auer, L. 2006. Nielsenin säännöt. Osoitteessa <http://www2.amk.fi/digma.fi/www.amk.fi/opintojaksot/030308/1111676348138/1111677021119/1161290796532/1161290917294.html>. 08.03.2014.
- Auer, L. 2009. Käyttöliittymän määrittelyä. Osoitteessa <http://www2.amk.fi/digma.fi/www.amk.fi/opintojaksot/030308/1111676348138/1111677021119/1111677160787/1111677410876.html>. 14.04.2014.
- Autodesk Inc. 2014. Adatable file format for 3D animation software. Osoitteessa <http://www.autodesk.com/products/fbx/overview>. 11.05.2014.
- Blender 2014. UV Mapping. Osoitteessa <http://wiki.blender.org/index.php/Doc:2.6/Manual/Textures/Mapping/UV>. 11.05.2014.
- Kuoppala, H. – Parkkinen, J. – Sinkkonen, I. – Vastamäki, R. 2006. Käytettävyyden psykologia. Helsinki: Edita Prima Oy. 3. uudistetty painos.
- Kuutti, W. 2003. Käytettävyys, suunnittelu ja arviointi. Helsinki: Talentum Media Oy
- Laakko, J. 2014. Opinnäytetyön palaveri. 31.01.2014.
- Lapland Snow Design –hanke. 2013. Lapland Snow Design – Lapin lumiosaaminen vientituotteeksi –hanke. Osoitteessa <http://lapland-snow-design.blogspot.fi/p/info.html>. 06.03.2014.
- Lennes, M. 2004. 9.1. Mikä on skripti?. Osoitteessa <http://www.helsinki.fi/puhetieteet/atk/praat/node46.html>. 11.05.2014.
- Menard, M. 2011. Game Development with Unity. Boston: Course Technology / Cengage Learning.
- Nielsen, J. 1995. 10 Usability Heuristics for User Interface Design. Osoitteessa <http://www.nngroup.com/articles/ten-usability-heuristics/>. 27.03.2014
- pLAB 2014. Lapland Snow Design: Lapin lumiosaamisenvientituotteeksi (2011-2013). Osoitteessa <http://plab.ramk.fi/fi/lapland-snow-design-2011-2013/>. 06.03.2014.
- Unity 3D 2010. GameObjects. Osoitteessa <http://docs.unity3d.com/Documentation/Manual/GameObjects.html>. 27.04.2014.

- Unity 3D 2011. Rays from the Camera. Osoitteessa <http://docs.unity3d.com/Documentation/Manual/CameraRays.html>. 21.04.2014.
- Unity 3D 2012. The GameObject-Component Relationship. Osoitteessa <http://docs.unity3d.com/Documentation/Manual/TheGameObject-ComponentRelationship.html>. 05.05.2014.
- Unity 3D 2013a. 3D formats. Osoitteessa <http://docs.unity3d.com/Documentation/Manual/3D-formats.html>. 27.04.2014.
- Unity 3D 2013b. Creating Scenes. Osoitteessa <http://docs.unity3d.com/Documentation/Manual/CreatingScenes.html>. 13.05.2014.
- Unity 3D 2013c. Materials and Shaders. Osoitteessa <https://docs.unity3d.com/Documentation/Manual/Materials.html>. 11.05.2014.
- Unity 3D 2013d. Meshes. Osoitteessa <http://docs.unity3d.com/Documentation/Components/class-Mesh.html>. 11.05.2014.
- Unity 3D 2013e. What is a Tag?. Osoitteessa <http://docs.unity3d.com/Documentation/Components/Tags.html>. 11.05.2014.
- Unity 3D 2014a. Asset Workflow. Osoitteessa <http://unity3d.com/unity/workflow/asset-workflow>. 27.04.2014.
- Unity 3D 2014b. Colliders. Osoitteessa <https://unity3d.com/learn/tutorials/modules/beginner/physics/colliders>. 21.04.2014.
- Unity 3D 2014c. Multiplatform. Osoitteessa <https://unity3d.com/unity/multiplatform>. 12.04.2014.
- Unity 3D 2014e. Raycasting. Osoitteessa <http://unity3d.com/learn/tutorials/modules/beginner/physics/raycasting>. 21.04.2014.
- Unity 3D 2014f. The Unity Editor. Osoitteessa <http://unity3d.com/unity/workflow/integrated-editor>. 06.05.2014.
- Unity 3D 2014g. Unity Scripting. Osoitteessa <https://unity3d.com/unity/workflow/scripting>. 21.04.2014.

Wii, A. 2004. Käyttäjäystävällisen sovelluksen suunnittelu. 1. painos. Helsinki: Edita Prima Oy.

Yliharju, A-J. 2014. Opinnäytetyön palaveri. 19.03.2014.