TARTU ÜLIKOOL
Arvutiteaduse instituut
Informaatika õppekava

Kalle Ever

Keskkondade vaheliste portaalide algoritm ja selle kasutamine arvutimängus

Bakalaureusetöö (9 EAP)

Juhendajad: Raimond-Hendrik Tunnel, MSc
Margus Luik, MSc

Tartu 2017

Keskkondade vaheliste portaalide algoritm ja selle kasutamine arvutimängus

Lühikokkuvõte:

Käesolev töö kirjeldab virtuaalsete keskkondade vaheliste portaalide algoritmi, mis võimaldab läbi portaali näha teist virtuaalset keskkonda ning läbi mille on võimalik objektidel ühest keskkonnast teise liikuda. Portaalide renderdamist vaadeldakse kahel viisil: tekstuurile renderdamine ja šabloonpuhvri kasutamine ning võrreldakse nende implementeerimiskeerukust ja ressursikasutust. Töö raames valmis arvutimäng, milles kasutatakse portaale ühe mängumehaanikana mõistatuste lahendamiseks. Kirjeldatakse portaale ja keskkondi kasutavate mõistatuste disaini. Viimaseks viidi läbi loodud arvutimängu kasutusmugavuse, mängumehaanikate ja mõistatuste testimine.

Võtmesõnad:

Portaalid, virtuaalsed keskkonnad, arvutigraafika, arvutimängud

CERCS: P170 Arvutiteadus, arvutusmeetodid, süsteemid, juhtimine (automaatjuhtimisteooria)

Algorithm for portals between environments and using it in a computer game

Abstract:

In this thesis an algorithm for portals between virtual environments is described. Portals allow one to view another virtual environment through it and make it possible to transfer objects between the virtual environments. Rendering of portals is described in two ways: rendering to a texture and using the stencil buffer. Complexity of implementation and resource usage are compared between the rendering algorithms. A computer game was created as a part of this thesis where previously described portals are used as a game mechanic to solve puzzles. The design of those puzzles is described. Lastly the ease of use, the game mechanics and the puzzles of the computer game were tested.

Keywords:

Portals, virtual environments, computer graphics, computer games

CERCS: P170 Computer science, numerical analysis, systems, control

Sisukord

1. Sissejuhatus

Käesoleva töö kirjeldab keskkondade vaheliste portaalide algoritmi, mis võimaldab läbi selle vaadelda ühest virtuaalsest keskkonnast teist virtuaalset keskkonda ning keskkondade vahel objekte liigutada. Portaale on varasemalt kasutatud näiteks arvutimängus Portal [1], kuid sealsed portaalid erinevad käesolevas töös kirjeldatavatest portaalidest. Arvutimängus Portal olevad portaalid võimaldavad vaadelda ühe keskkonna erinevaid asukohti ning nende asukohtade vahel liikuda. Keskkondade vahetamist on kasutatud näiteks arvutimängus Fran Bow [2], kuid seal ei kasutata selleks portaale. Töö raames valmis ka arvutimäng, milles keskkondade vaheliste portaalide abil on võimalik liikuda viie erinevate omadustega ja kunstiliste stiilidega keskkonna vahel, et lahendada mõistatusi. Valminud arvutimängu täiustati aine MTAT.03.328 Arvutigraafika projekt raames. Seal aines parandati mängus olevaid vigu, disainiti juurde keerulisemaid mõistatusi ja loodi kolme keskkonna jaoks 3D mudelid ja nendele sobivad tekstuurid.

Töös antakse esmalt peatükis 2 lühiülevaade kasutatavast tehnoloogiast – mängumootorist Unity [3] ja võimalikest alternatiividest, mida töö autori sobiliku kogemuse korral oleks olnud võimalik kasutada. Seejärel kirjeldatakse peatükis 3 kahte keskkondade vaheliste portaalide renderdus-algoritmi: tekstuurile renderdamist kasutavat algoritmi ja šabloonpuhvrit (*stencil buffer*) kasutavat algoritmi. Võrreldakse algoritmide implementeerimiskeerukust ja ressursikasutust. Vastavalt võrdlustulemustele kirjeldatakse sobivama renderdusalgoritmi jaoks portaali läbimise algoritm.

Sellele järgnevalt kirjeldatakse peatükis 4 loodud arvutimängu mängumehaanikaid, mängus olevat viite keskkonda ja mängu jaoks disainitud mõistatusi, mis kasutavad kirjeldatud mängu-mehaanikaid ja keskkondi. Viimaseks toimus töö raames antud arvutimängu kasutusmugavuse ja mängumehaanikate testimine. Peatükis 5 kirjeldatakse testimist, avastatud probleeme ning nendele leitud lahendusi.

Töös kasutatud terminid on toodud Lisas I. Mängu käivitusjuhend ja mängujuhend on toodud vastavalt lisas II ja lisas III. Mäng on lisas IV ja mängu lähtekood lisas V.

2. Kasutatud tehnoloogiad

Antud töö üheks eesmärgiks on luua arvutimäng, mille loomise jaoks on hea kasutada mängumootorit. Mängumootoris sisalduvad arvutimängu loomiseks vajalikud tööriistad ja tehnoloogiad. Järgnevalt on kirjeldatud töös kasutatud mängumootor ja alternatiivid, mida oleks olnud võimalik sobiliku kogemuse korral kasutada.

2.1. Unity

Unity [3] on Unity Technologies poolt loodud mängumootor, mis võimaldab luua 2D ja 3D mänge erinevatele platvormidele (PC, Mac, Android, iOS jm.) [4]. Mänguobjektide skriptimine on võimalik programmeerimiskeeltes C# ja JavaScript ning varjutajad kirjutatakse varjutamis-keeltes Cg (Central Graphics), HLSL (High-Level Shader Language) ja GLSL (OpenGL Shading Language).

Unity Personal versioon on tasuta ja seda on võimalik kasutada nii kaua kuni eelneva aasta tulu sellega loodud arvutimängudest ei ületa 100 000 dollarit [5]. Kui tulu ületab ettemääratud summa, siis on vajalik osta litsents Unity Plus, kui tulu jääb alla 200 000 dollari [6] ning Unity Pro litsents vastasel juhul [7].

Töös kasutati mängumootorit Unity kuna töö autor on sellega tuttav ning seega oli võimalik keskenduda algoritmide implementatsioonile ning arvutimängu loomisele ilma, et peaks uue tehnoloogiaga tutvuma. Teiseks eelistati antud mängumootorit üle madalama taseme teekide (nt. OpenGL) kuna üheks töö osaks oli arvutimängu loomine, mis on oluliselt lihtsam ja vähem ajakulukas kasutades mängumootorit Unity. Mängumootor pakub mängude loomiseks vajalikke komponente, nagu füüsikamootor ja sisenditöötlus. Kolmandaks on Unity hinnapoliitika töö autori poolt eelistatud.

2.2. Alternatiivid

Käesolevas peatükis vaadeldakse korraks muid alternatiive, mida sobiliku kogemuse korral oleks olnud võimalik kasutada.

2.2.1. Unreal Engine 4

Unreal Engine 4 [8] on Epic Games poolt loodud mängumootor 2D ja 3D mängude loomiseks nii arvutile, nutitelefonidele kui ka mängukonsoolidele. Mänguobjektide skriptimine toimub Unreal Engine 4 mängumootoris kasutades programmeerimiskeelt C++ või visuaalset skriptimis-süsteemi Blueprints Visual Scripting [9]. Unreal Engine 4 on tasuta kasutatav, sellega loodud mängudest saadavast tulust tuleb maksta 5% Epic Games-le [10].

Töös ei kasutatud mängumootorit Unreal Engine 4 peamiselt seetõttu, et töö autoril oli antud mootoriga vähe kogemusi ning seetõttu poleks olnud võimalik keskenduda algoritmide implementatsioonile ja arvutimängu disainimisele ja loomisele.

2.2.2. Godot Engine

Godot Engine [11] on avatud lähtekoodiga tasuta mängumootor 2D ja 3D mängude loomiseks arvutile ja nutitelefonidele. Mänguobjektide skriptimine toimub programmeerimiskeelele Python sarnases keeles. Käesolevas töös ei kasutatud mängumootorit Godot Engine seetõttu, et töö autoril puudus kogemus antud mootoriga ja seega oleks olnud vajalik esialgu keskenduda mootori õppimisele, mis oleks jätnud vähem aega algoritmide implementatsioonile ja arvuti-mängu loomisele.

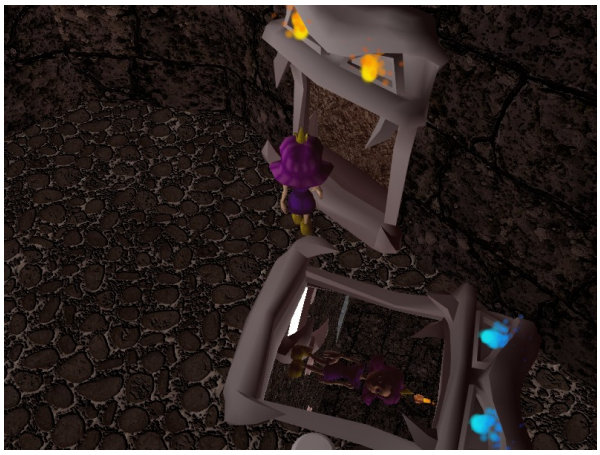
2.2.3. OGRE

OGRE (Object-oriented Graphics Rendering Engine) [12] on 3D graafikamootor, mis toetab Direct3D, OpenGL ja WebGL kasutamist [13]. Mootor võimaldab luua 3D graafikat kasutavaid rakendusi nii arvutile kui ka nutitelefonidele. Töös ei kasutatud antud mootorit seetõttu, et üheks töö osaks oli arvutimängu loomine. Arvutimängu loomine oleks mootoriga OGRE olnud keeru-lisem, kuna selles puuduvad osad selle jaoks vajalikud tehnoloogiad (näiteks füüsikamootor) [14].

3. Keskkondade vaheliste portaalide algoritm

Üldiselt mõistetakse arvutimängudes portaalide all selliseid portaale, mis võimaldavad näha sama virtuaalse keskkonna erinevaid punkte ning läbi portaali liigutada objekte ühest punktist teise. Sellist tüüpi portaale on kasutatud näiteks arvutimängudes Portal [1] (Joonis 1) ja Narbacular Drop [17] (Joonis 2).

Joonis 1: Portaalid arvutimängus Portal.



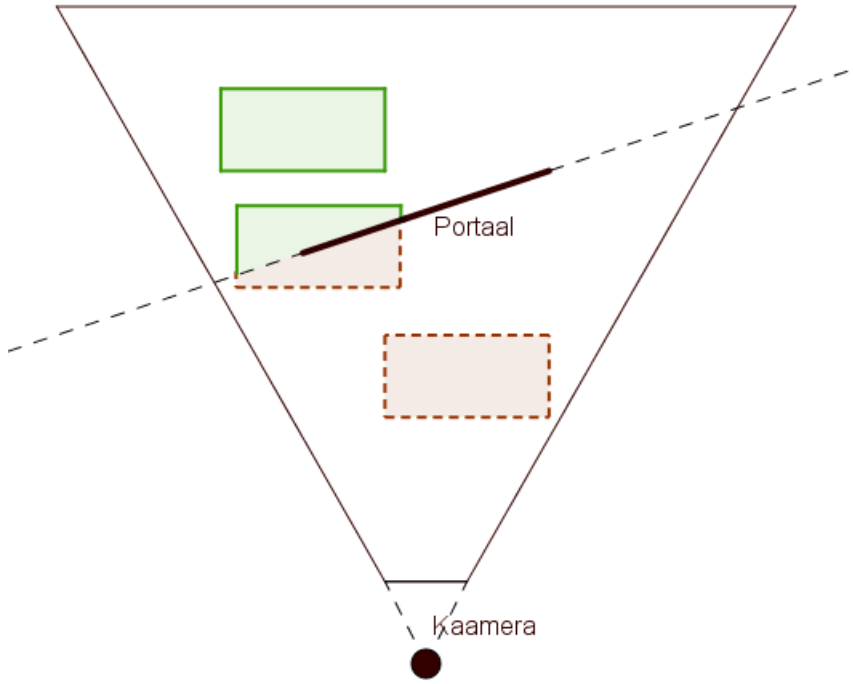
Joonis 2: Portaalid arvutimängus Narbacular Drop [17].

Antud töös kirjeldatakse portaale, mis võimaldab vaadelda ühest virtuaalsest keskkonnast teist virtuaalset keskkonda ning nende keskkondade vahel objekte liigutada. Käesolevas peatükis kirjeldatakse kahte algoritmi portaalide implementeerimiseks. Ka eelnevalt mainitud samasse keskkondade viivate portaalide juures kasutatakse sarnaseid lahendusi [18], siiski antud töös pole vajadust rekursiivsele lahendusele. Rekursiivset lahendust on vaja samasse keskkonda viivate portaalide puhul, kuna nendega on võimalik, et portaal on nähtav läbi selle sama portaali. Mõlemad töös kirjeldatud renderdusalgoritmid kasutavad kaldus vaatepüramiidi pügamist, mille kirjeldusega käesolev töö jätkub.

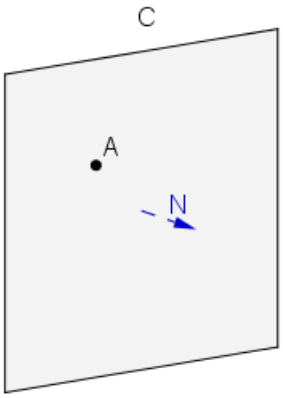
3.1. Kaldus vaatepüramiidi pügamine (*Oblique view frustum clipping)*

Läbi portaali nähtavat keskkonda renderdatakse virtuaalse kaameraga. Portaali kaudu on vaja näha teisest keskkonnast ainult neid objekte, mis asuvad portaali läbiva tasandi taga. Objektid, mis asuvad kaamera ja portaali vahel, ei tohi olla läbi portaali nähtavad. Joonisel 3 on näide

olukorrast, kus teise keskkonna objektid asuvad portaali ja kaamera vahel. Punase katkendliku joonega on tähistatud teise keskkonna objektid või objektide osad, mis asuvad portaali tasandist vaatleja pool ja seega ei peaks olema vaatlejale nähtavad. Rohelise pideva joonega on kujutatud objektide osad mis asuvad teisel pool portaali tasandit. Portaali läbivast tasandist kaamera poole jäävad objektid on vaja jätta renderdusest välja või renderdada nendest objektidest ainult osa, juhul kui tasand lõikab objekti.



Joonis 3: Teise keskkonna objektid, mis on portaali kaamera vaateväljas.



Joonis 4: Tasand C, mille normaal on N ja A on suvaline punkt sellel tasandil.

Kaldus vaatepüramiidi pügamine on E. Lengyeli poolt kirjeldatud meetod [19], mis võimaldab kaamera projektsioonimaatriksit **P** muuta nii, et selle kaamera tavaline lähitasand asendatakse mingi teise tasandiga **C**. Tasandit **C** saab esitada 4-mõõtmelise vektoriga kujul

$$\boldsymbol{C} = \left(N_x, N_y, N_z, -\boldsymbol{N} \cdot \boldsymbol{A}\right) \quad ,$$

kus **N** on tasandi normaal ning **A** on suvaline punkt tasandil (Joonis 4).

Muudetud projektsioonimaatriks näeb välja järgmine:

$$\boldsymbol{P}' = \begin{bmatrix} \boldsymbol{P}_1 \\ \boldsymbol{P}_2 \\ \dfrac{2\boldsymbol{P}_4 \cdot \boldsymbol{Q}}{\boldsymbol{C} \cdot \boldsymbol{Q}} \boldsymbol{C} - \boldsymbol{P}_4 \\ \boldsymbol{P}_4 \end{bmatrix},$$

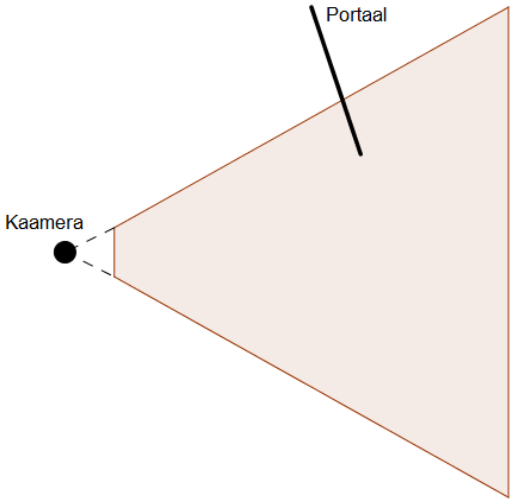
kus $\boldsymbol{P}_n$ tähistab esialgse projektsioonimaatriksi *n*-dat rida ja $\boldsymbol{Q} = \boldsymbol{P}^{-1}(sgn(C_x), sgn(C_y), 1, 1)$ [19]. Kui tegemist on perspektiivprojektsiooniga, siis tüüpiliselt arvutigraafikas on maatriksi neljas rida $\boldsymbol{P}_4 = (0, 0, -1, 0)$ ning muudetud maatriksi lihtsustatud kuju on järgnev [19]:

$$\boldsymbol{P}' = \begin{bmatrix} \boldsymbol{P}_1 \\ \boldsymbol{P}_2 \\ \dfrac{-2\boldsymbol{Q}_z}{\boldsymbol{C} \cdot \boldsymbol{Q}} \boldsymbol{C} + \langle 0, 0, 1, 0 \rangle \\ \boldsymbol{P}_4 \end{bmatrix}.$$

Antud töös kasutatakse maatriksi lihtsutatud kuju. Selline maatriks viib kaamera lähitasandi etteantud tasandi C kohale ehk sellest kaamera poole jäävad objektid jäetakse renderdusest välja. Objektidest, mida tasand C lõikab renderdatakse ainult need fragmendid, mis asuvad teisel pool tasandit.

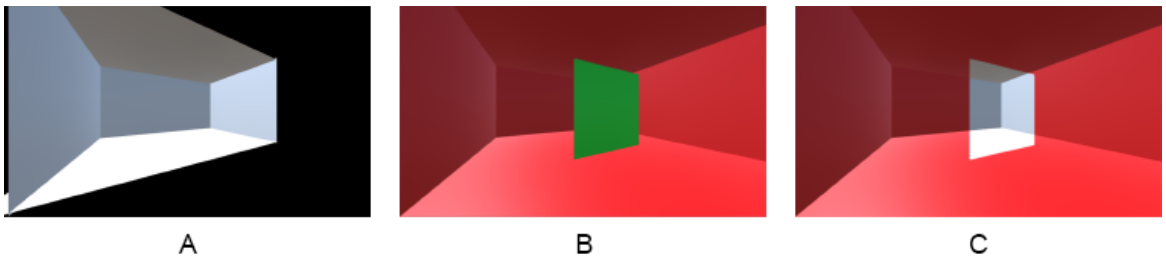
3.2. Tekstuurile renderdamist kasutav renderdusalgoritm

Antud algoritmi puhul luuakse iga portaali jaoks kaamera ning tekstuur, millele loodud kaamera renderdab – kutsume seda kaamerat portaali kaameraks. Olukorras, kus portaal on põhikaamerast nähtav (Joonis 5), seatakse portaali kaamera projektsioonimaatriks ja kaameramaatriks samaks põhikaamera omaga. Seejärel muudetakse portaali kaamera projektsioonimaatriksit kasutades kaldus vaatepüramiidi pügamist (vt. peatükk 3.1), et asendada selle kaamera lähitasand portaali tasandiga. Portaali kaamera renderdus salvestatakse tekstuurile.



Joonis 5: Portaal, mis on stseeni põhi-kaamerast nähtav.

Portaali kaamera pool renderdatud tekstuur sisaldab vaadet teisest keskkonnast (Joonis 6:A) arvestamata sealjuures portaali suurust ja asukohta. Eesmärk on näidata seda vaadet ainult läbi portaali nähtuna, seega on vaja kasutada tervest portaali kaamera renderdusest sobiliku osa. Joonisel 6:B on kujutatud roheliselt tekstuurita portaali objekt ekraanil ning joonisel 6:C on portaali objekt, millele on portaali kaamera renderdatud tekstuurist vastendatud sobilik osa.



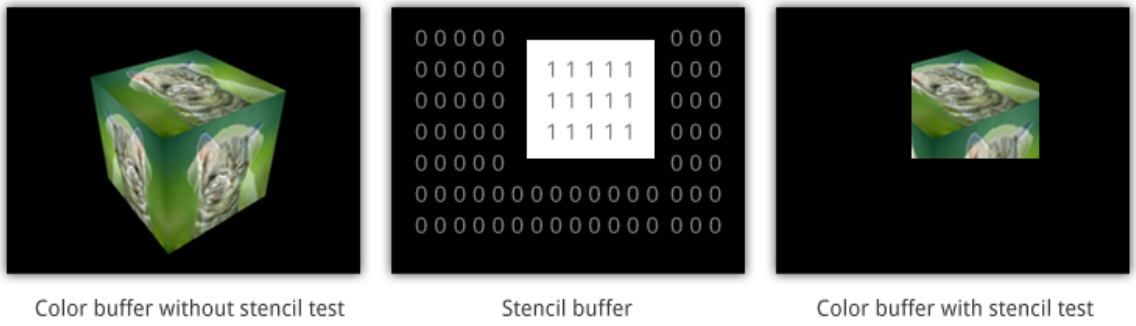
Joonis 6: Saadud tekstuuri vastendamine portaali objektile.

Sellise vastendamise saavutamiseks leitakse varjutajas iga tipu pügamisruumi koordinaadid ning nendele rakendatakse Unity poolt pakutavat abifunktsiooni *ComputeScreenPos* [20], mis leiab antud punkti koordinaadid ekraanil. Saadud tipu koordinaadid interpoleeritakse fragmendi-varjutajas, et leida iga fragmendi koordinaadid ekraanil. Ekraani vasak alumine nurk on koordinaatidega $(0,0)$ ning parem ülemine nurk koordinaatidega $(1,1)$. Tekstuuri tekslid kasutavad samasugust koordinaatsüsteemi ja nii ekraan kui ka saadud tekstuur on sama

kuvasuhtega. Seetõttu leitud fragmendi koordinaatid ekraanil vastavad sobiva(te)le teksli(te)le tekstuuril ning need vastendatakse fragmendile.

3.3. Šabloonpuhvrit kasutav renderdusalgoritm

Šabloonipuhver on 8-bitine puhver, mis võimaldab iga fragmendi kohta kirjutada täisarvulise väärtuse ning järgnevatel renderdustel vastavalt puhvris olevatele väärtustele fragmente alles või ära jätta.



Joonis 7: Šabloonpuhvri tööpõhimõtte näide [21].

Joonisel 7 on kujutatud näide šabloonipuhvri kasutusest. Esiteks täidetakse šabloonipuhvri kõikide fragmentide väärtused nullidega ja seejärel on antud näites osa puhvrist täidetud ühtedega. Kuubiku joonistamisel kuvatakse ainult need fragmendid, millele vastav šabloonipuhvri väärtus antud näites on üks [21].

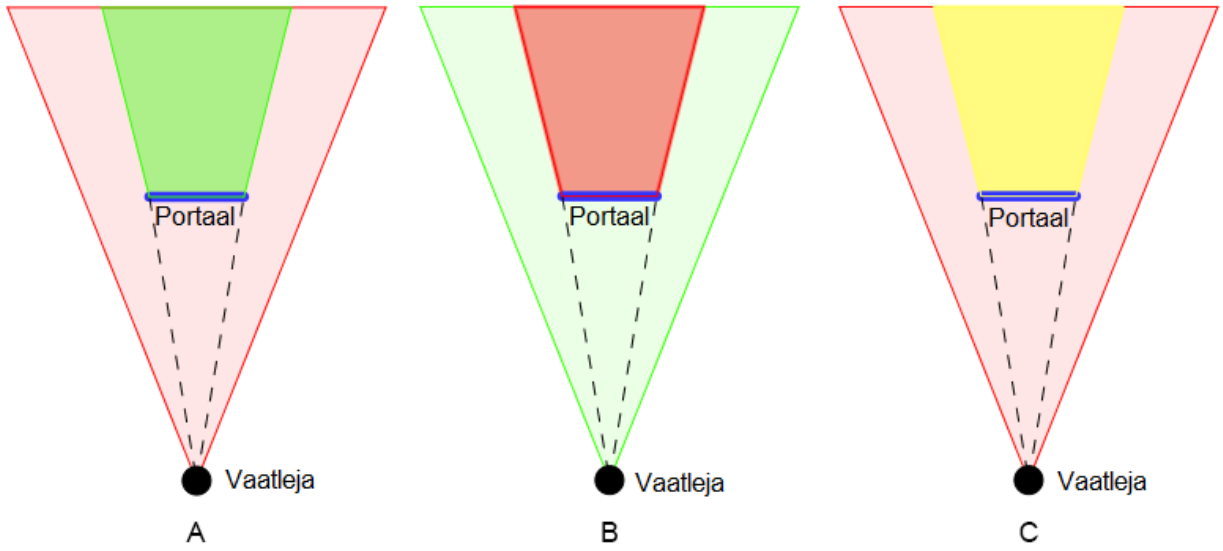
Portaali joonistamisel täidetakse portaalile vastav ala šabloonpuhvris täisarvuga **m**, mis vastab läbi portaali nähtavale virtuaalsele keskkonnale. Ülejäänud osa täidetakse täisarvuga **n**, mis vastab vaatleja virtuaalsele keskkonnale. Kuna Unity ei võimalda lihtsasti ühe kaameraga sama kaadri jooksul mitu korda renderdada, siis kasutatakse stseeni joonistamiseks kahte kaamerat. Esimese kaamera projektsioonimaatriksile rakendatakse kõigepealt kaldus vaatepüramiidi pügamist (vt. peatükk 3.1.), et asendada selle lähitasand portaali tasandiga ning seejärel renderdatakse selle kaameraga läbi portaali nähtava keskkonna need fragmendid millele šabloon-puhvrist vastav väärtus on **m**. Viimaseks renderdatakse teise kaameraga vaatleja keskkond kohtades, millele vastav väärtus šabloonpuhvris on **n**.

3.4. Renderdusalgorimide võrdlus

Eelnevalt kirjeldatud renderdusalgoritmid on levinud variandid portaalide jaoks. Vajalik on leida neist sobivaim töö käigus loodava arvutimängus kasutamiseks. Selle jaoks võrreldakse algoritmide implementeerimiskeerukust ja ressursikasutust.

3.4.1. Implementeerimiskeerukus

Olukorras, kus on vaja läbi portaali näha ühest virtuaalsest keskkonnast **A** teist virtuaalset keskkonda **B** (Joonis 8:A), on mõlema algoritmi implementatsioon sarnase keerukusega. Erinevus tekib siis kui on vajalik näha ka keskkonnast **B** keskkonda **A** (Joonis 8:B) või kui on olemas kolmas keskkond **C** mida on vaja näha kas keskkonnast **A** või **B** (Joonis 8:C).



Joonis 8: Erinevatest keskkondadest läbi portaali teiste keskkondade vaatamine. Punane tähistab keskkonda A, roheline keskkonda B ja kollane keskkonda C.

Sellisel juhul on tesktuurile renderdamist kasutava algoritm oluliselt lihtsam, sest vajalik on ainult muuta keskkond, mida portaali kaamera peab renderdama. Šabloonpuhvrit kasutava algoritmi korral peab esiteks muutma kaamera poolt renderdatavat keskkonda, šabloonpuhvrisse kirjutatavaid väärtuseid ning ka väärtuseid, mida võrreldakse keskkondade renderdamisel šabloonpuhvris olevate väärtustega.

3.4.2. Ressursikasutus

Ressursikasutuse testimiseks kasutati arvutit operatsioonisüsteemiga Windows 10, protsessoriga Intel® Core™ i7-4600U, graafikakaaridga NVIDIA GeForce 840m 4GB ning muutmälu mahuga 8GB. Mõõdeti erinevate stseenide puhul ühe kaadri renderdamiseks kuluvat aega üle 500 kaadri.

Tabel 1: Kaadri renderdamiseks kulunud keskmine, minimaalne ja maksimaalne aeg üle 500 kaadri sõltuvalt kolmnurkade arvust ühes keskkonnas.

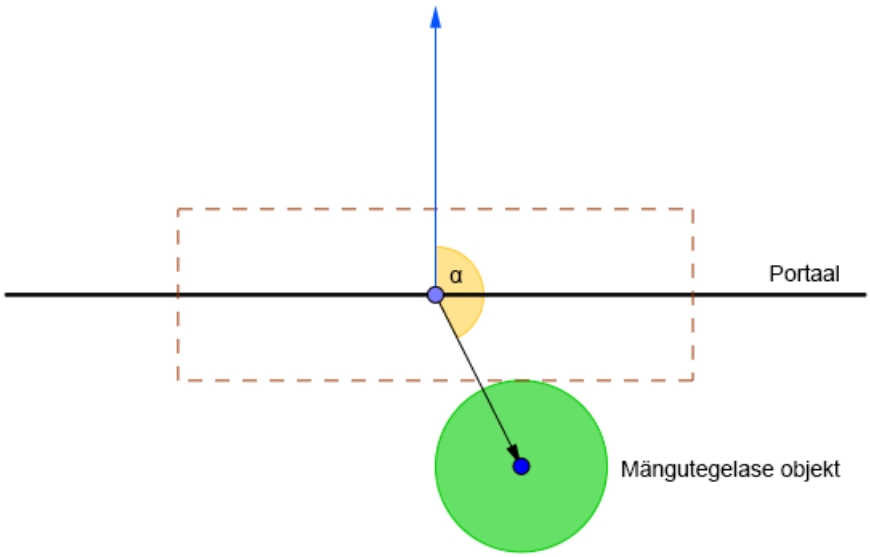
Kolmnurkade arv ühes keskkonnas	**Kaadri renderdamiseks kulunud aeg (ms)**					
	Tekstuurile renderdamist kasutav algoritm			**Šabloonpuhvrit kasutav algoritm**		
	Vähim	**Keskmine**	**Suurim**	**Vähim**	**Keskmine**	**Suurim**
1500	3,85	5,85	17,58	2,94	6,13	13,90
750000	5,92	8,29	21,87	3,99	5,76	20,60
1550000	8,35	11,71	24,01	6,14	9,20	22,00
4600000	21,72	28,51	49,60	20,01	27,12	39,45
19200000	100,00	115,02	140,22	99,28	116,51	171,72
23000000	131,52	149,00	190,71	131,53	146,74	192,63

Tabelis 1 on kujutatud kolmnurkade arv ühes keskkonnas (st. mõlemas keskkonnas oli märgitud arv kolmnurkasid) ning mõlema algoritmi kohta ühe kaadri renderdamiseks kulunud minimaalne, maksimaalne ning keskmine aeg. Tulemustest on näha, et šabloonpuhvrit kasutaval algoritmil kulus stseeni renderdamiseks keskmiselt 1.15 millisekundit vähem aega kui tesktuurile renderdamist kasutaval algoritmil. Leitud vahe on üsna väike antud töö raames loodud arvuti-mängu jaoks, seega antud kontekstis olulist erinevust algoritmide ressursikasutuses ei ole.

3.5. Portaalide läbimise algoritm

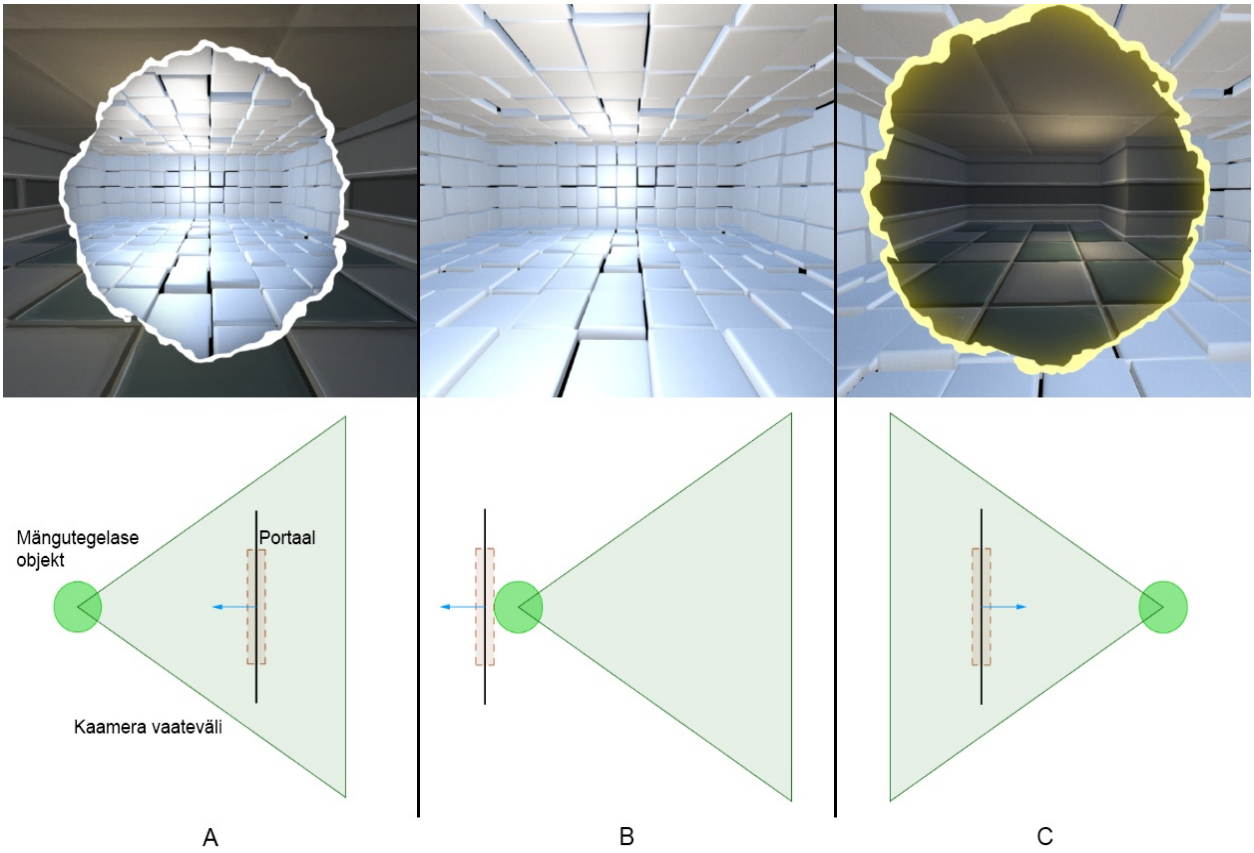
Läbi portaali liikumiseks kontrollitakse mängutegelase objekti kokkupuute (*collision*) lõpetamist portaali objektiga. Kokkupuute lõpu korral võrreldakse portaali objekti asukoha ja mängija objekti asukoha vahelise vektori suunda portaali normaaliga, et leida, kas mängutegelane asub portaali ees või taga. Joonisel 9 on kujutatud näide mängutegelase kokkupuute lõpust portaali alaga. Punane katkendlik joon tähistab portaali kokkupuutetuvastusala (*collider*) ning sinise noolega on märgitud portaali tasandi pinnanormaal, mis määrab ühtlasi portaali suuna. Mängija näeb uut keskkonda ainult siis, kui ta paikneb portaali positiivses suunas ja vaatab portaali poole.

Portaali tagant uut keskkonda näha ei ole. Kui mängija asub peale kokkupuute lõppu portaali taga, siis seatakse põhikaamera renderdama uut keskkonda ning mängija objekti viiakse uuele keskkonnale vastavale kokkupuute füüsikakihile. Unity mängumootoris on erinevad kokkupuute füüsikakihid, mis võimaldavad kokkupuudete arvutamisi teha ainult sobilikel kihtidel olevate objektide vahel.



Joonis 9: Mängutegelase objekti kokkupuute lõpp portaali pinnaga. Mängija liikunud läbi portaali ala portaali taha.

Lisaks eelnevale kustutatakse eksisteeriv portaali objekt ning selle asemel luuakse uus portaal vastupidises suunas, mis kuvab eelnevat keskkonda ning võimaldab mängijal sellesse tagasi liikuda. Joonis 10:A kujutab mängijat enne portaali läbimist, joonis 10:B kujutab keskkonna-vahetuse hetke ja joonis 10:C kujutab olukorda, kus mängija vaatab uue portaali poole, kust avaneb eelmine keskkond.



Joonis 10: Mängija liigub läbi portaali. Ülemine rida kuvab stseeni mängija vaatepunktist ja alumine rida kuvab sama olukorda ortograafilises pealtvaates.

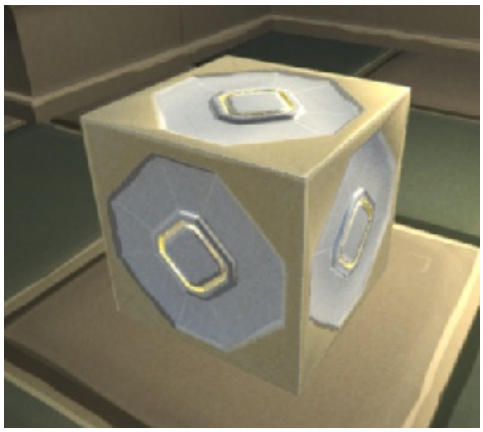
Käesolevas peatükis kirjeldati keskkondadevaheliste portaalide algoritmide tehnilisi keerukusi ja implementatsioone. Kirjeldatud algoritmid rahuldavad püsitatud eesmärke edukalt. Edasi suundutakse mängudisaini osa juurde, kus on seletatud portaalide mehaanikast tulenevad mängu-mehaanikad ja mängu ülesehitus. Arvutimängus kasutatakse portaalide renderdamiseks tekstuurile renderdavat algoritmi, kuna see võimaldab lihtsamalt läbi portaali nähtavat keskkonda muuta.

4. Mängu realisatsioon

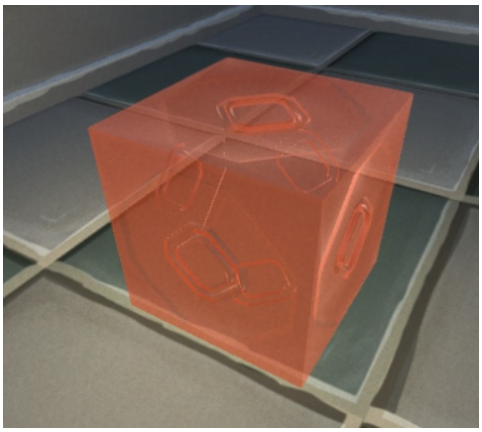
Töö raames valmis arvutimäng, milles peatükis 3 kirjeldatud portaale kasutatakse ühe mängumehaanikana mõistatuste lahendamiseks. Mäng toimub esimese isiku (*first-person*) vaates ning eesmärgiks on liikuda erinevates alades iga ala algusest lõppu. Selle saavutamiseks kasutab mängija erinevate omadustega keskkondadesse (vt. Peatükk 4.2.) viivaid portaale, et lahendada antud alas sisalduv(ad) mõistatus(ed) (vt. Peatükk 4.3.). Alad on ehitatud selliselt, et nende edukaks läbimiseks on vajalik liikuda läbi erinevate keskkondade. Järgnevalt on kirjeldatud mängu kulgu mõistatuste lahendamisel, mängumehaanikaid ning detailsemalt mängus olevaid keskkondasid.

4.1. Mängumehaanikad ja eesmärk

Mängijal on võimalik liikuda kahel erineval kiirusel, hüpata piiratud kõrgusele ja avada portaale erinevatesse keskkondadesse. Lisaks võib alades leiduda kaste (Joonis 11), mida on mängijal võimalik kätte võtta ning liigutada (ka erinevate keskkondade vahel). Kui kast paikneb erinevas keskkonnas mängija omast, siis on kast mängijale nähtav poolläbipaistvana (Joonis 12). Kasti poolläbipaistva kujutise värv sõltub keskkonnast, milles kast tegelikult paikneb.



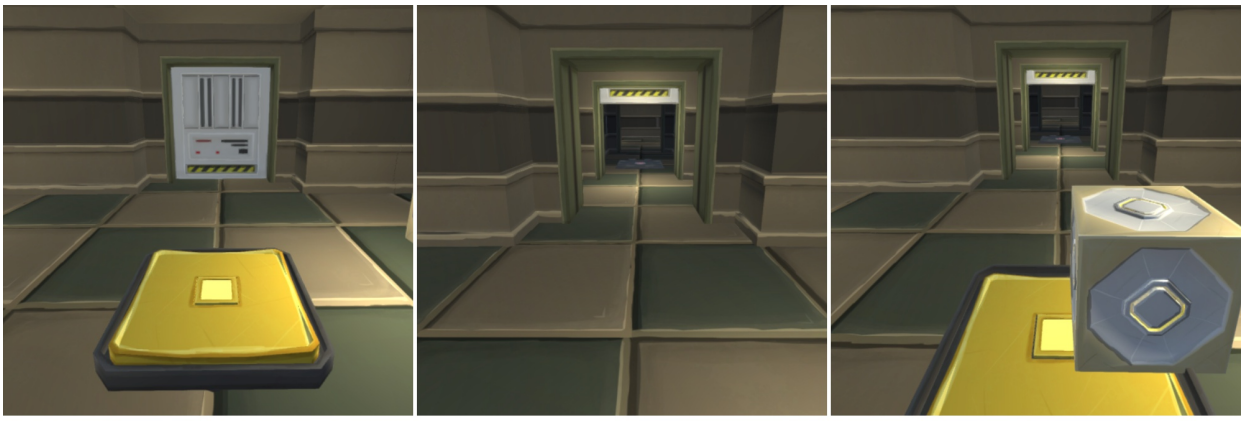
Joonis 11: Mängija poolt liigutatav kast.



Joonis 12: Kast, mis paikneb erinevas keskkonnas mängija omast.

Alades paiknevad tihti uksed, mille avamiseks on vajalik vajutada sellega seotud põrandal asuvat nuppu (Joonis 13:A). Nupp võib ukse ka sulgeda või mõjutada mitut ust korraga. Nupu

vajutamiseks liigutab mängija mängutegelase nupule (Joonis 13:B). Peale nupult maha kõndimist sulgub uks pärast *n* sekundit (kus *n* on määratud mõistatuse loomisel). Ukse pidevalt lahti hoidmiseks on mängijal võimalik asetada nupu peale kast (Joonis 13:C). Kasti nupule asetamine on samade omadustega nagu astuks mängija ise nupule. See tähendab, et ka kasti eemaldamisel nupu pealt sulgub uks *n* sekundi pärast.



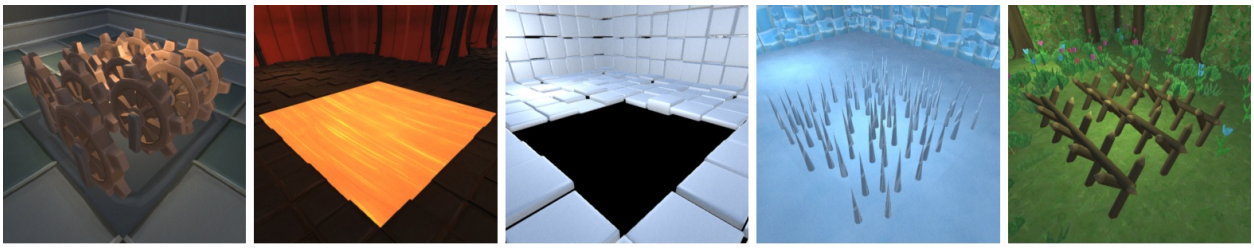
Joonis 13: Ukse avamine nupu abil. A – mängija seisab nupu ees. B – mängija on astunud nupule ja uks avaneb. C – mängija on nupule asetanud kasti.

Mängus on ka lõksud, millega kokkupuutel viiakse mängija käesoleva ala algusesse tagasi. Olukorras, kus kast läheb lõksu vastu viiakse seegi tagasi oma alguskohta. Lõksud võivad olla nii statsionaarsed kui ka liikuvad (vt. peatükk 4.2.1.).

Ala lõpetamisel sulgub mängija taga uks, mis blokeerib eelmisesse alasse tagasimineku. Lisaks sellele hävitatakse uude alasse sisenemisel eelnevasse alasse kuuluvad mängija poolt liigutatavad kastid. Alad on täielikult eraldiseisvad, seega eelnevatest aladest objektide kaasa võtmine võib muuta ala lahendamise mängija jaoks liiga lihtsaks. Teiseks, kui lubada mängijale objektide kaasa võtmine, siis on vajalik sellega mõistatuste loomisel arvestada. See muudaks nende disainimise oluliselt keerulisemaks, kuna ei ole võimalik teada kui palju ja millised objektid iga mängija kaasa otsustab võtta.

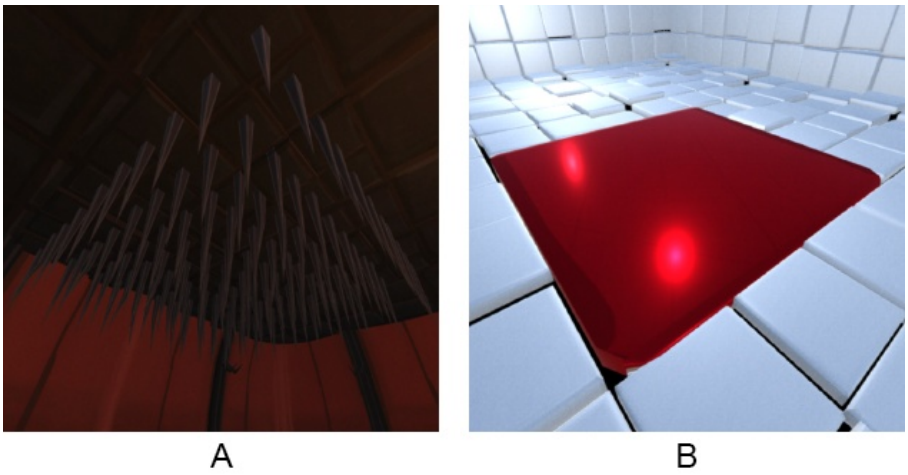
4.2.1. Lõksud

Mängus on kahte tüüpi lõksud: statsionaarsed ja liikuvad. Statsionaarsed lõksud paiknevad määratud asukohtades ja ei liigu. Asukohad võivad olla nii põrandal, seintel kui ka laes. Lõksudel on erinev välimus igas keskkonnas (Joonis 14).



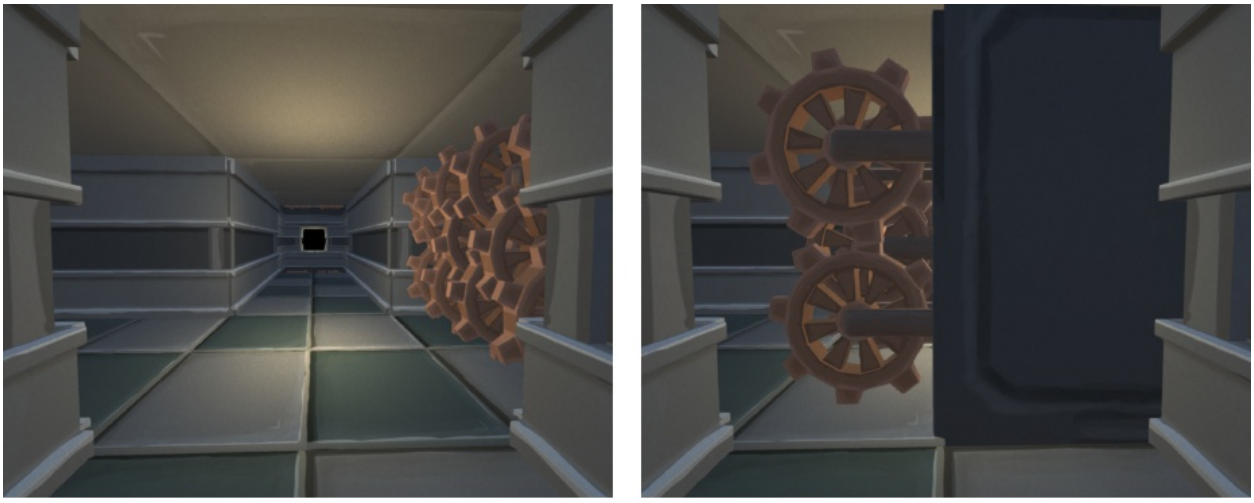
Joonis 14: Lõksud erinevates keskkondades. A – keskkond A, B – keskkond B, C – gravitatsioonita keskkond, D – aeglane keskkond, E – elav keskkond.

Lisaks sellele on keskkonnas B seintel ja laes asuvate lõksude välimus erinev (Joonis 15:A). Gravitatsioonita keskkonnas on lõksud üldiselt keskkonna sees olevad augud (Joonis 14:C). Eriolukord tekib, kui augu järel läheb maailm edasi, kuid gravitatsioonita keskkonnas peaks seal lõks olema. Sellisel juhul kujutatakse gravitatsioonita keskkonna lõksu teistsugusena - augu peal oleva punase alana (Joonis 15:B).



Joonis 15: Lõksude alternatiivsed välimused. A – keskkonnas B, B – gravitatsioonita keskkonnas.

Liikuvad lõksud liiguvad sirgjooneliselt kahe punkti vahel (Joonis 16), peatudes kummaskis otspunktis mõistatuse loomisel määratud ajaks.

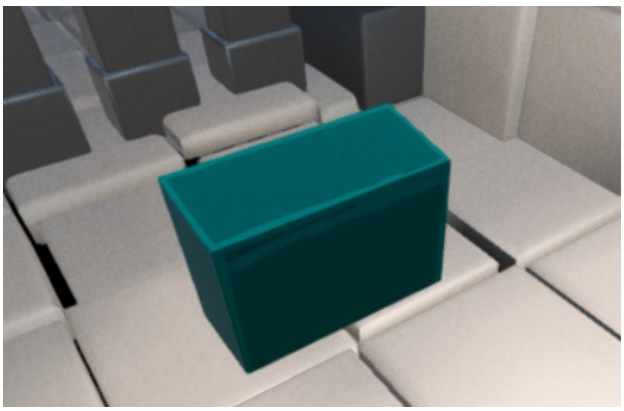


Joonis 16: Liikuv lõks keskkonnas A. A – lõks paikneb liikumistee ühes otspunktis. B – lõks paikneb liikumistee teises otspunktis.

4.2. Keskkonnad

Mängus on viis erinevate omadustega ning välimustega keskkonda. Mängijal on võimalik nende vahel liikumiseks avada mängutegelase ette portaale. Portaalide avamiseks peab mängija eelnevalt mängus üles korjama portaalide avamise seadme (Joonis 17). Esimest korda seadme üles korjamisel on võimalik mängijal liikuda keskkondade A ja B vahel. Ülejäänud kolme keskkonda liikumiseks peab mängija koguma mängu jooksul vastavaid seadme lisamooduleid (Joonis 18).

Joonis 17: Portaalide avamise seade mängutegelase käes.

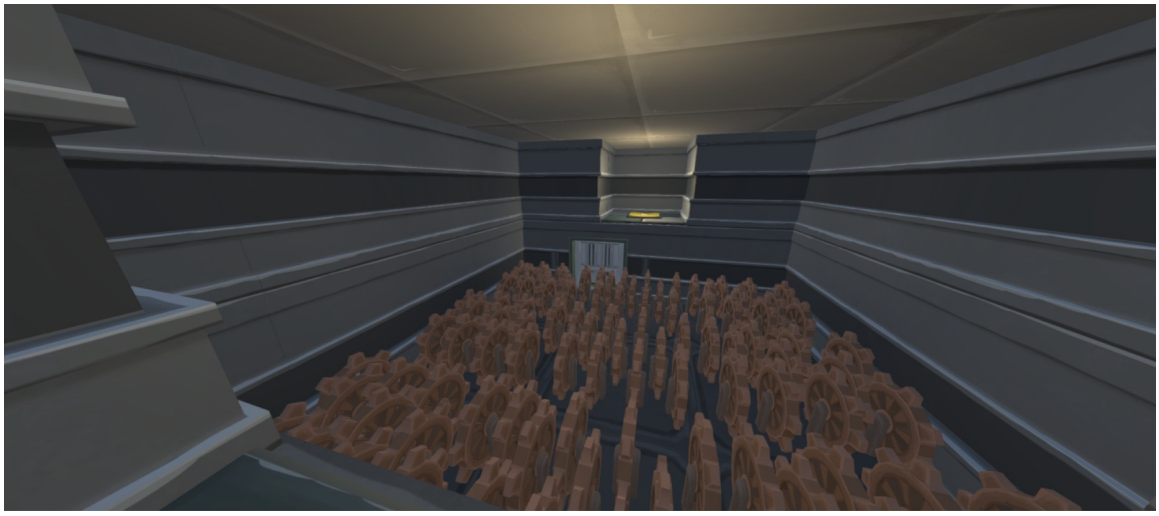


Joonis 18: Keskkonda portaali avamiseks vajalik lisamoodul.

Portaali objekti loomisel kontrollitakse portaali lõikumist objektidega ning vajaduse korral liigutatakse piiratud ulatuses portaali objekti nii, et lõikumist ei toimuks. Kui sobivat asukohta ei leita, siis portaali objekti ei looda. Järgnevates alapeatükkides kirjeldatakse detailsemalt mängus olevat viite keskkonda.

4.2.1. Keskkond A

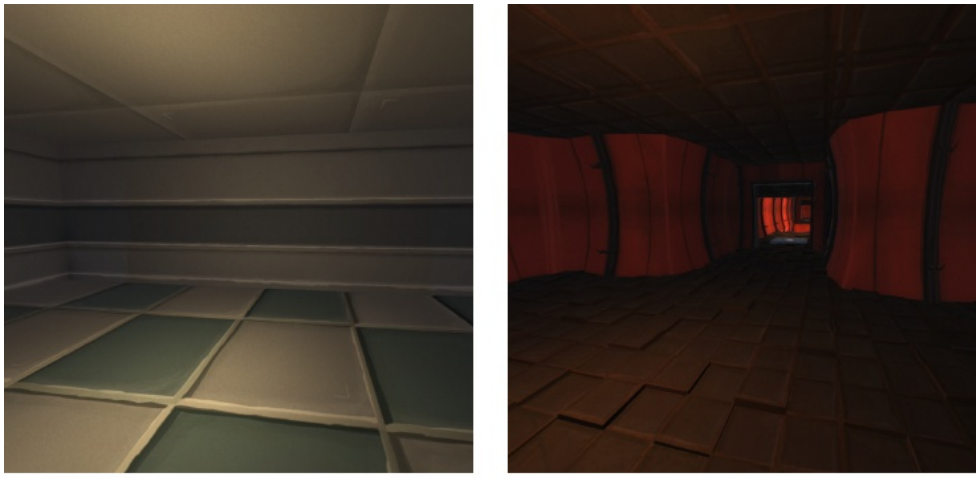
Keskkonnas A asub mängija alguses. Keskkond koosneb seintest, põrandatest ja kaldpindadest. Antud keskkonna kunstiline stiil on tehislik (Joonis 19). Keskkonna välimuse eesmärgiks on mängijas tekitada tunne, et see keskkond on inimeste loodud. Seetõttu on keskkond A sarnane päris maailmale ja jätab mulje, et seal kehtivad päris maailmale sarnased reeglid.



Joonis 19: Keskkonna A tehislik kunstiline stiil.

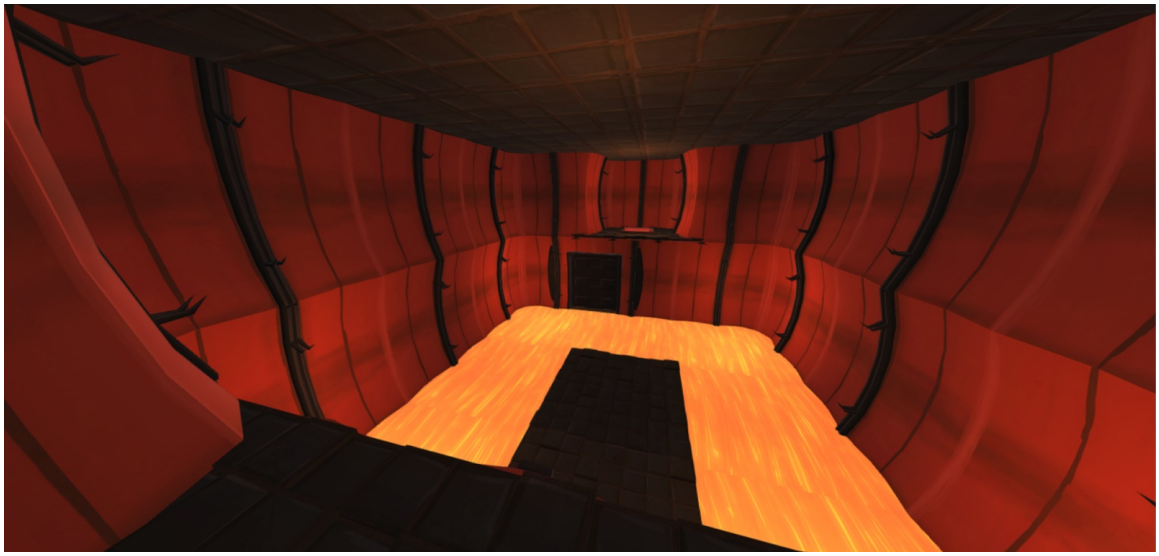
4.2.2. Keskkond B

Antud keskkonnas on olulised erinevused objektide paigutuses. Näiteks võib võrreldes keskkonnaga A mõni sein puududa või olla mõni objekt kohas, kus seda keskkonnas A eelnevalt ei olnud. Joonisel 20:A on kujutatud ruum keskkonnas A ja joonisel 20:B on kujutatud sama ruum keskkonnas B, kus üks seina osa on puudu võrreldes keskkonnaga A.



Joonis 20: Sama ruum keskkondades A ja B. Keskkonnas B on osa seinast puudu ja avaneb teekond edasi.

Seetõttu võimaldab keskkond B mängijal liikuda ruumis kohtadesse, kuhu keskkonnas A pole võimalik liikuda. See võimaldab luua näiteks sellise mõistatuse, kus mängija peab ületama lõksud, mille kohal keskkonnas A ei ole midagi ja keskkonnas B on põrand.

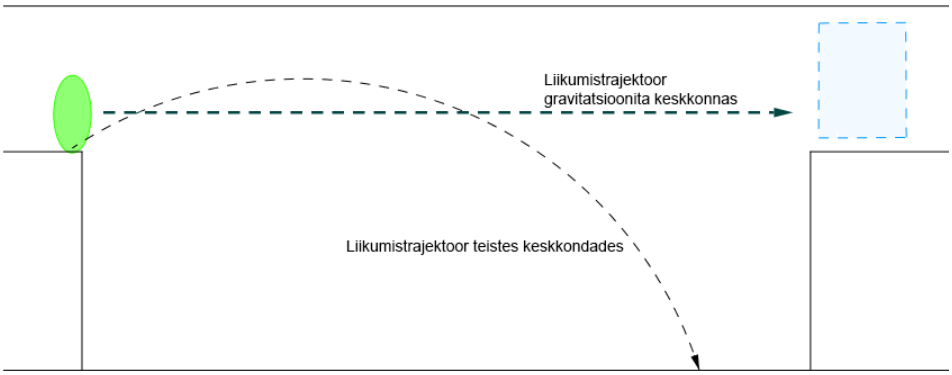


Joonis 21: Keskkonna B kunstiline stiil.

Keskkonna B kunstiline stiil on toodud joonisel 21. Antud keskkonna välimuse eesmärgiks on tekitada mängijas tunne, et on liigutud oluliselt erinevasse keskkonda võrreldes keskkonnaga A.

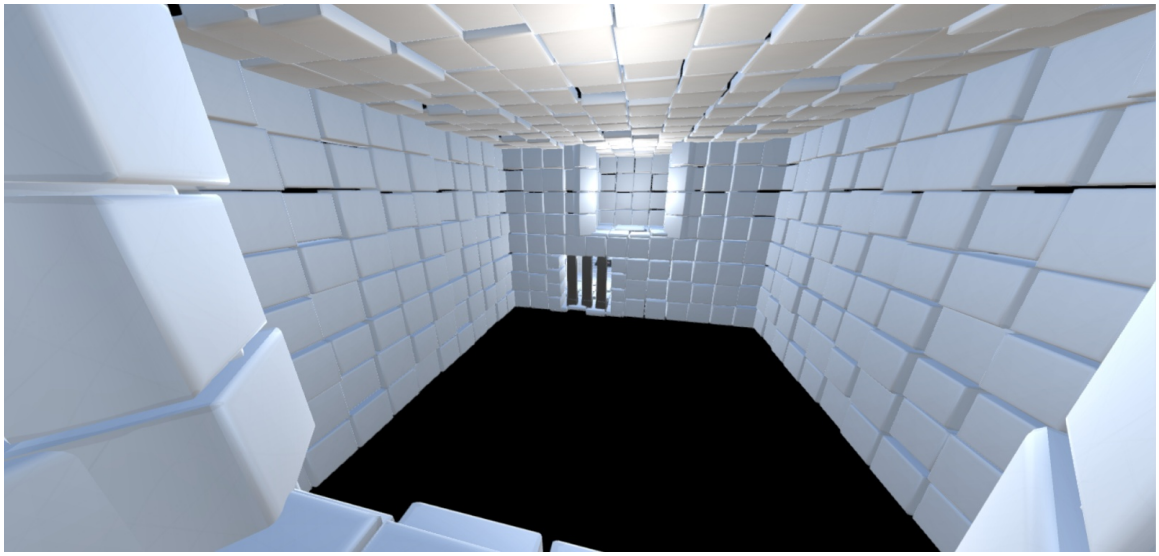
4.2.3. Gravitatsioonita keskkond

Gravitatsioonita keskkonnas on objektid asetatud sarnaselt keskkonnale A. Samas ei rakendata selles keskkonnas gravitatsiooni mängijale ega objektidele. See võimaldab koostada ja lahendada mõistatusi kus mängia peab pääsema kohta, kuhu tavalise liikumise ja hüppamisega pole võimalik jõuda. Joonisel 22 on toodud näide sellisest mõistatusest. Antud mõistatuses on mängija (roheline ovaal) eesmärgiks on jõuda ala lõppu, mis on joonisel tähistatud sinise ristkülikuna. Lihtsalt hüppamisega on seda võimatu teha, hüppetrajektoor on kujutatud joonisel musta katkendliku joonega. Augu ületamiseks peab mängija kasutama gravitatsioonita keskkonda.



Joonis 22: Gravitatsioonita keskkonda kasutav mõistatus.

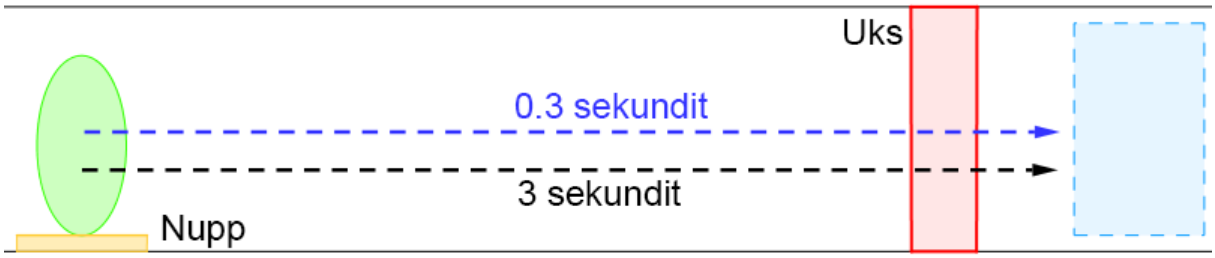
Antud keskkonna kunstiline stiil on minimalistlik (Joonis 23). Seinad ja põrandad koosnevad valgetest kuubikutest, uste ja nuppude juures on kasutud tumehalli värvi. Kuubikute taga on näha musta värvi, mille eesmärgiks on tekitada mängijas tunne, et gravitatsioonita keskkond paikneb tühjuses.



Joonis 23: Gravitatsioonita keskkonna kunstiline stiil.

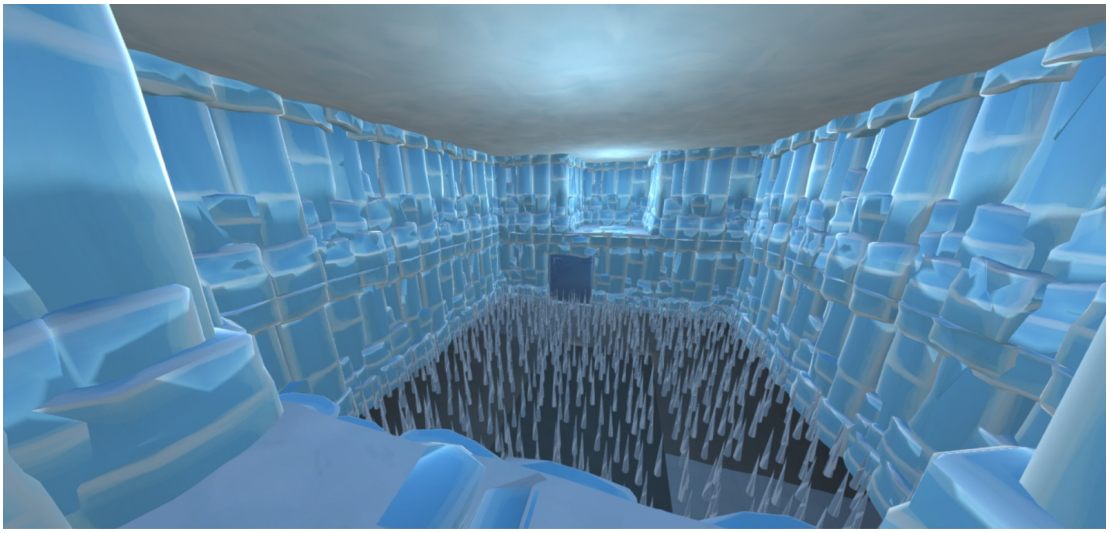
4.2.4. Aeglane keskkond

Aeglases keskkonnas liigub aeg 10% normaalkiirusest, aja aeglustumine ei mõjuta mängijat. Aja kiirus aeglases keskkonnas valiti 10% kuna sellel kiirusel on mängijale näha, et aeg liigub ikkagi edasi, aga tegevuste sooritamiseks on oluliselt rohkem aega. Aja aeglustamine võimaldab luua mõistatusi, mille teatud osad on sellise ajapiiranguga, et tavakiirusel on seda võimatu lahendada. Näide sellisest mõistatusest on toodud joonisel 24. Mängija (roheline ovaal) eesmärgiks on jõuda ala lõppu (sinine ristkülik). Oranži ristkülikuga tähistatud nupp avab punase ristkülikuga tähistatud ukse 2 sekundiks. Tavalise ajakiirusega keskkonnas liikuval mängial kuluks uksest läbi jõudmiseks 3 sekundit (joonisel 24 must nool), kuid selle aja peale on uks juba sulgunud. Aeglases keskkonnas kuluv relatiivne aeg sama teekonna läbimiseks on 0.3 sekundit (joonisel 24 sinine nool), seega mängija jõuab uksest läbi enne selle sulgumist.



Joonis 24: Aeglast keskkonda kasutav mõistatus.

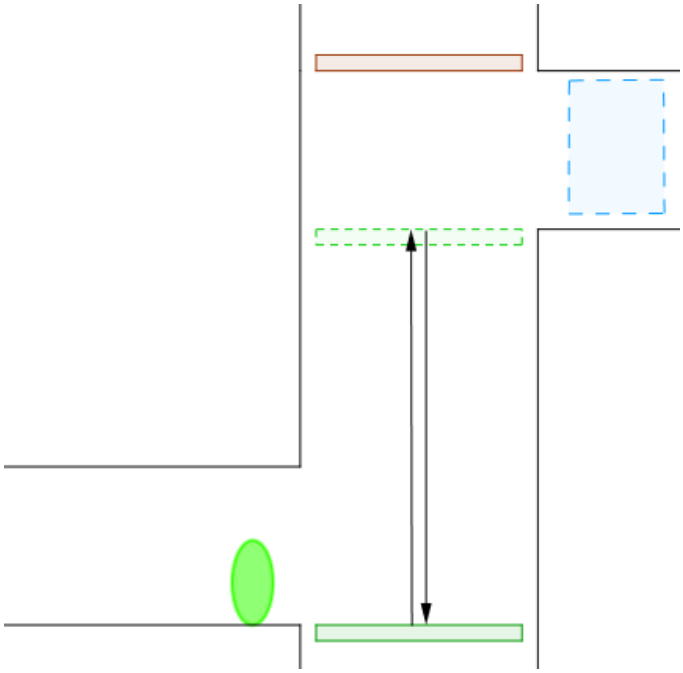
Aeglane keskkond koosneb jääst ja lumest (Joonis 25). Antud välimus valiti aeglasele keskkonnale seetõttu, et arvutimängudes seostatakse külmust tihti aeglustumisega (näiteks külmunud mängutegelane liigub aeglasemalt kui tavaliselt).



Joonis 25: Aeglase keskkonna kunstiline stiil.

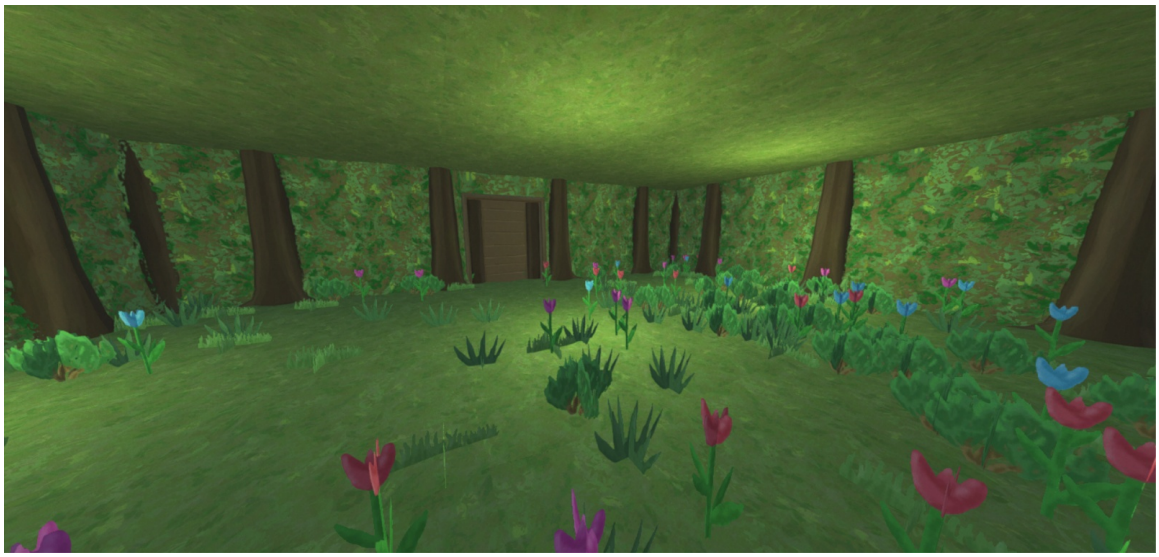
4.2.5. Elav keskkond

Elavas maailmas liiguvad teatud horisontaalsed pinnad ehk platvormid vertikaalselt. Need pinnad eksisteerivad ka teistes keskkondades, aga on seal statsionaarsed ja paiknevad algus-asukohas. Sellised pinnad võimaldavad luua ja lahendada mõistatusi, milles mängija peab liikuma vertikaalselt ning gravitatsioonita keskkonna kasutamine ega hüppamine ei ole sobilikud. Joonisel 26 on kujutatud näide sellisest mõistatusest. Mängija (roheline ovaal) eesmärgiks on jõuda ala lõppu, mis on joonisel tähistatud sinise ristkülikuna. Hüppamise jaoks asub ala lõpp liiga kõrgel ning gravitatsioonita keskkonna abil ei ole võimalik üles liikuda, kuna kokkupuutel lõksuga (tähistatud punase ristkülikuga) viiakse mängutegelane ala algusesse tagasi. Seega ainus viis ala lõpuni jõuda on kasutada elavas keskkonnas vertikaalselt liikuvat platvormi, mille liikumistee üks otspunkt on tähistatud pideva joonega rohelise ristkülikuga ning teine otspunkt rohelise katkendliku joonega.



Joonis 26: Elavat keskkonda kasutav mõistatus.

Elava keskkonna kunstiline stiil on looduslik (Joonis 27). Maailm koosneb puudest ja taimedest. Antud välimus valiti seetõttu, et selles tundub liikuvate pindade olemasolu üsna loomulik, kuna loodus on elav ja muutuv.



Joonis 27: Elava keskkonna kunstiline stiil.

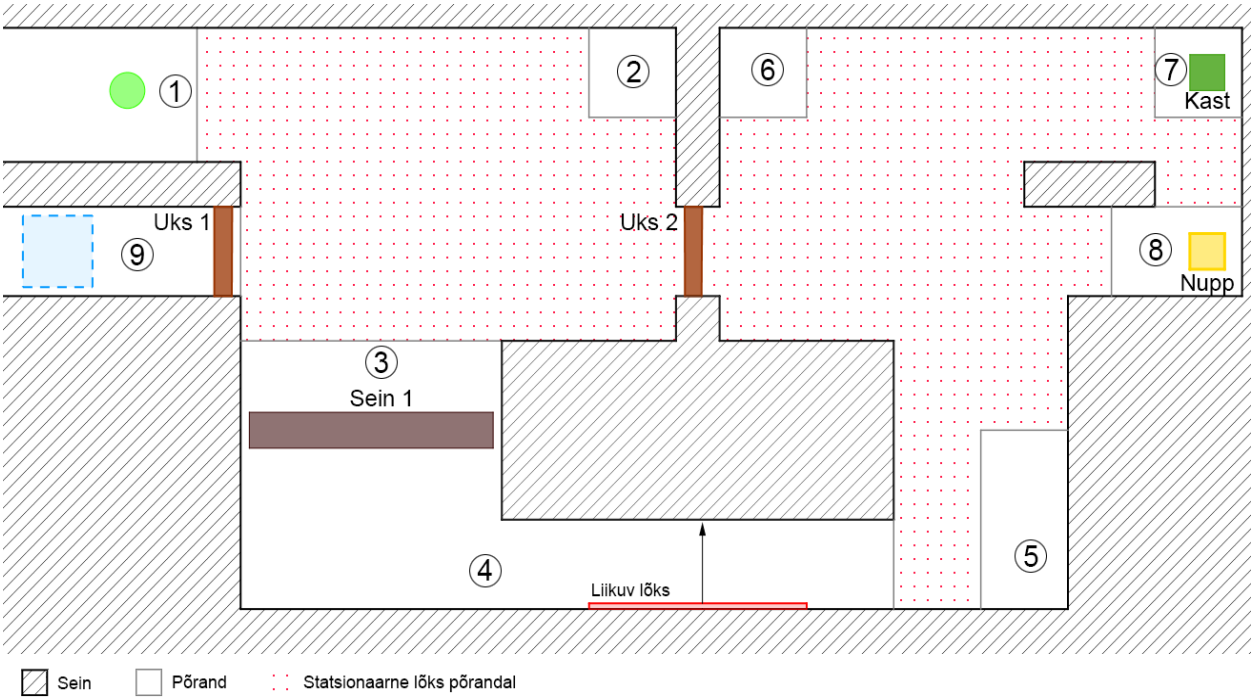
4.3. Mõistatused

Mängu esimesed 17 mõistatust tutvustavad mängijale eelnevalt kirjeldatud mängumehaanikaid ja keskkondasid. Mängumehaanikaid tutvustavad mõistatused nõuavad mängijalt ainult konkreetse mängumehaanika kasutamist. Näiteks kasti abil nupu vajutamise tutvustamiseks on alas ainult üks uks, seda ust avav nupp ja kast. Keskkondade tutvustamisel on iga keskkonna kohta üks lihtne mõistatus, mille lahendamiseks on vaja ainult vastavat keskkonda kasutada. Need mõistatused on sarnased peatükis 4.2. kirjeldatud võimalike mõistatustega iga keskkonna jaoks. Igale uut keskkonda tutvustavale mõistatusele järgnevad 1–3 pisut keerulisemat mõistatust, milles mängija peab kasutama seda keskkonda ja mõnda eelnevalt tutvustatud keskkonda või mängumehaanikat. Need mõistatused tutvustavad mängijale erinevaid olukordasid, milles eelnevalt tutvustatud keskkonda on vajalik kasutada.

Sissejuhatavatele mõistatustele järgnevad 10 keerulisemat kombineeritut mõistatuset, mille lahendamiseks peab mängija kasutama mitut erinevat keskkonda korduvalt. Järgnevalt kirjeldatakse nendest nelja.

4.3.1. Keskkonnad B, gravitatsioonita ja aeglane

Esimene kombineeritud mõistatus on toodud joonisel 28 pealtvaates. Mängija (roheline ring) eesmärgiks on jõuda ala lõppu (sinine ristkülik). Alad millel mängija saab seista on tähistatud valgelt, punaste punktidega on tähistatud lõksud põrandal. Must kaldjoon tähistab seina. Kollase ristkülikuga tähistatud nupp avab uksed 1 ja 2 üheks sekundiks. Sein 1 puudub keskkonnas B.

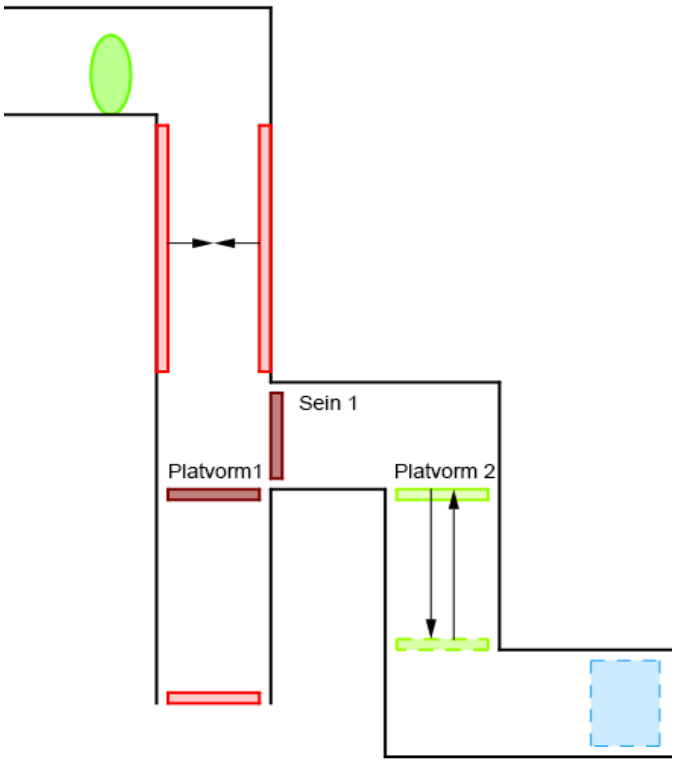


Joonis 28: Keskkonda B, gravitatsioonita keskkonda ja aeglast keskkonda kasutav mõistatus ortograafilises pealtvaates. Numbritega on näidatud mängija liikumistee.

Punktist 1 punkti 2 ja sealt punkti 3 liikumiseks kasutab mängija gravitatsioonita keskkonda. Punktist 3 punkti 4 liikumiseks on vajalik keskkonna B kasutamine, et saada mööda seinast 1. Edasi liikumist takistab liikuv lõks (punane ristkülik). Sellest mööda pääsemiseks peab mängija kasutama aeglast keskkonda, sest teistes keskkondades liigub lõks liiga kiiresti. Punktist 5 punkti 6 ja sealt punkti 7 liikumiseks on vajalik gravitatsioonita keskkonna kasutamine. Seejärel võtab mängija punktis 7 kasti ning hüppab sellega punkti 8 juures asuvale nupule, et avada uksed 1 ja 2. Seejärel saab mängija liikuda sirgjooneliselt ala lõppu kasutades gravitatsioonita keskkonda.

4.3.2. Keskkonnad A, B, aeglane ja elav

Joonisel 29 on kujutatud teine kombineeritud keerulisem mõistatus külgvaates. Mängija (roheline ovaal) eesmärgiks on jõuda ala lõppu (sinine ristkülik). Lõksud on tähistatud punase ristkülikuga, kusjuures nooled lõksude juures tähistavad liikuvat lõksu. Sein 1 ja platvorm 1 eksisteerivad ainult keskkonnas B. Platvorm 2 liigub elavas keskkonnas nooltega näidatud teekonnal.



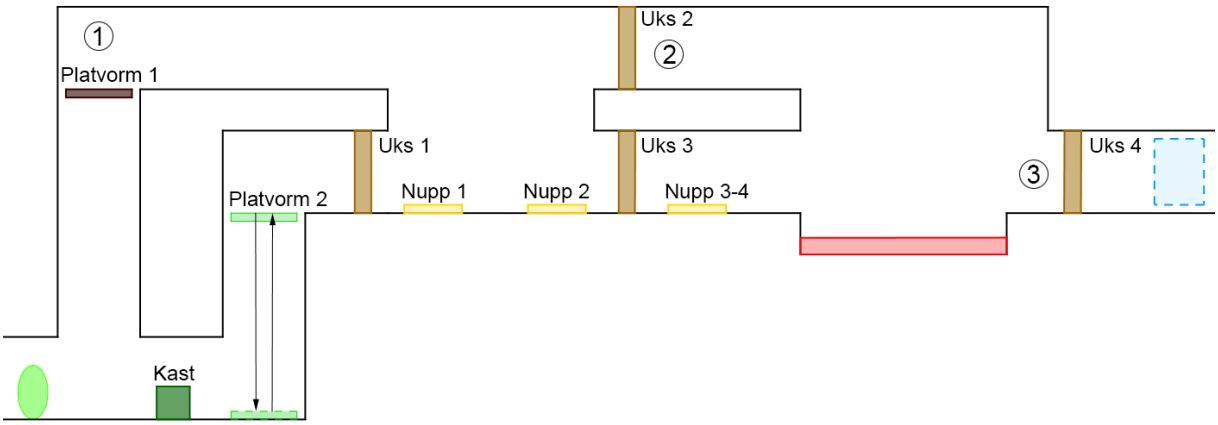
Joonis 29: Keskkonda A, keskkonda B, aeglast keskkonda ja elavat keskkonda kasutav mõistatus ortograafilises külgvaates.

Liikuvatest lõksudest läbi pääsemiseks peab mängija kasutama aeglast keskkonda, teistes keskkondades liiguvad lõksud liiga kiiresti. Seejärel on vaja kukkumise ajal avada portaal keskkonda B, et mängija maanduks platvormil 1, vastasel juhul satub mängija kokkupuutesse allpool oleva statsionaarse lõksuga. Seinast 1 mööda pääsemiseks tuleb kasutada mõnda keskkonda, mis pole B. Platvormist 2 läbi pääsemiseks on vaja elavas keskkonnas oodata, kuni platvorm on liikunud alla ning seejärel avada portaal mõnda teise keskkonda platvormi esialgse asukoha alla.

4.3.3. Keskkonnad B, gravitatsioonita, aeglane ja elav

Kolmas keerulisem mõistatus on kujutatud joonisel 30. Mängija (roheline ovaal) eesmärgiks on jõuda ala lõppu (sinine ristkülik). Lõksud on tähistatud punase ristkülikuga. Platvorm 1 eksisteerib ainult keskkonnas B ja platvorm 2 liigub elavas keskkonnas. Nupp 1 avab ukse 1 kahekümneks sekundiks. Nupp 2 avab ukse 2 ning nupp 3-4 avab uksed 3 ja 4. Nupud, mille

kohta eelnevalt aega ei ole mainitud, sulgevad nendega seotud uksed kohe pärast nupu lahti laskmist.

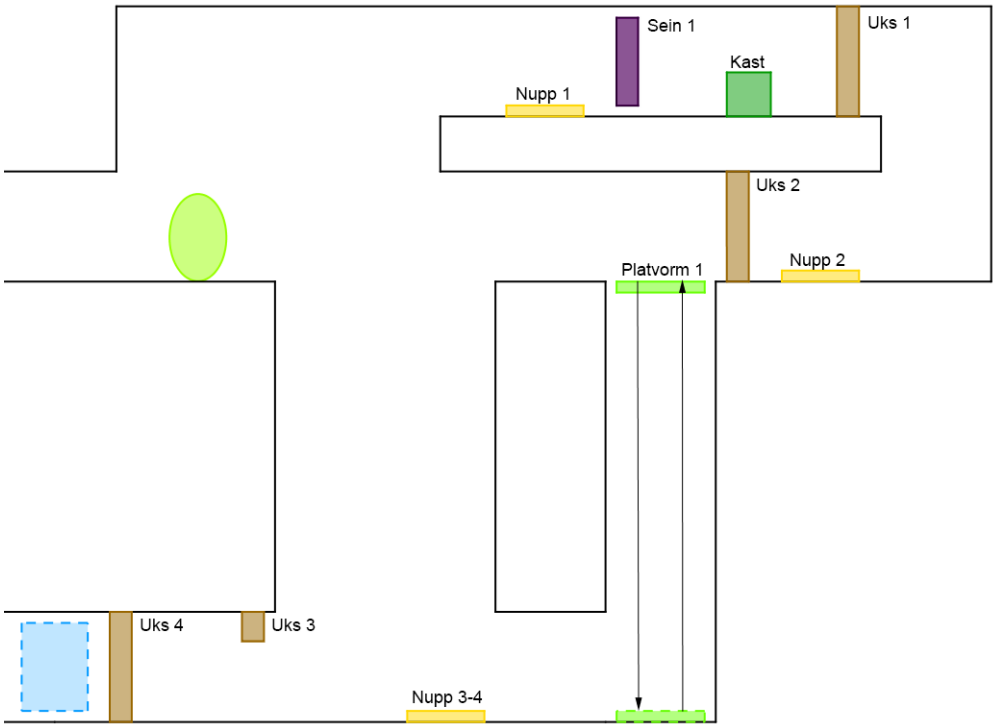


Joonis 30: Keskkonda B, gravitatsioonita keskkonda, aeglast keskkonda ja elavat keskkonda kasutav mõistatus ortograafilises külgvaates.

Mängija algusasukohast punkti 1 jõudmiseks peab mängija kasutama esiteks gravitatsioonita keskkonda, et liikuda laeni ning seejärel keskkonda B, et maanduda platvormil 1. Seejärel astub mängija nupule 1, et avada uks 1. Kasutades elavat keskkonda on mängijal võimalik liikuda ala algusesse tagasi, et võtta kaasa kast ning minna tagasi läbi ukse 1 enne kui see sulgub. Järgmiseks viib kasti aeglasesse keskkonda ja asetab selle nupule 2, et hoida uks 2 lahti. Ukseni 2 jõudmiseks saab mängija kasutada gravitatsioonita keskkonda. Seejärel liigub mängija punktist 2 punkti 3 hüpates või kasutades gravitatsioonita keskkonda. Punktist 3 liigub mängija nupule 3 kasutades gravitatsioonita keskkonda. Pärast ukse avanemist kasutab mängija aeglast keskkonda, et võtta nupu 2 pealt sinna eelnevalt asetatud kast ning viib selle nupule 3 enne ukse sulgumist. Viimaseks kasutab mängija gravitatsioonita keskkonda, et liikuda ala lõppu.

4.3.4. Kõik keskkonnad

Joonis 31 kujutab neljandat kombineeritud keerulisemat mõistatust. Mängija (roheline ovaal) eesmärgiks on jõuda ala lõppu (sinine ristkülik). Sein 1 ei eksisteeri keskkonnas B. Platvorm 1 liigub elavas keskkonnas. Nupp 1 avab ukse 1 viieks sekundiks. Nupp 2 avab ukse 2. Uks 3 on vaikimisi avatud, nupp 3-4 sulgeb selle ja avab ukse 4. Nupud, mille kohta eelnevalt aega ei ole mainitud, sulgevad nendega seotud uksed kohe pärast nupu lahti laskmist.



Joonis 31: Kõiki keskkondi kasutav mõistatus ortograafilises külgvaates.

Mängija kasutab esiteks gravitatsioonita keskkonda, et liikuda nupu 1 juurde. Sealt edasi kasutatakse keskkonda B, et pääseda mööda seinast 1. Seejärel kasutatakse keskkonda A, et seal võtta kast ja liikuda läbi ukse 1. Järgnevalt liigub mängija nupule 2, et saada läbi uksest 2. Seistes uksega 2 samal tasemel viib mängija kasti aeglasesse keskkonda ning laseb selle nupu 3 kohal lahti ja hüppab seejärel ise alla uste 3 ja 4 vahele. Aeglases keskkonnas kukub kast aeglasemalt kui mängija, seega mängija jõuab minna läbi ukse 3 enne kasti nupule 3 kukkumist. Kui kast jõuab nupuni, siis uks 3 sulgub ja uks 4 avaneb, viimane võimaldab mängijal liikuda ala lõppu. Kui mängija laseb kasti vales kohas lahti ja seega kast ei kuku nupule 3, siis on mängijal võimalik kasutada elavas keskkonnas liikuvat platvormi 1, et liikuda koos kastiga ülemisele tasemele tagasi.

5. Testimine ja tulemused

Töö raames toimus ka eelnevalt kirjeldatud arvutimängu kasutusmugavuse ja mängumehaanikate testimine. Järgnevalt kirjeldatakse testimise metoodika, leitud probleemid ja nendele probleemidele võimalikud lahendused.

5.1. Metoodika ja testijad

Arvutimängu kasutusmugavuse testimiseks viidi läbi 3 testsessiooni. Kõikidel testijatel oli eelnevalt kogemusi nii esimese isiku vaates toimuvate arvutimängudega kui ka mõistatustega arvutimängudega. Testijad mängisid esiteks läbi 17 sissejuhatavat mõistatust ja seejärel neli keerulisemat kombineeritud mõistatust. Mängimise ajal vaadeldi seda, kas erinevad mängumehaanikad ja keskkonnad on mängijale piisavalt hästi selgitatud. Lühim testsessioon kestis 54 minutit ja pikim 81 minutit. Testsessioonide videolindistused on toodud lisas VI.

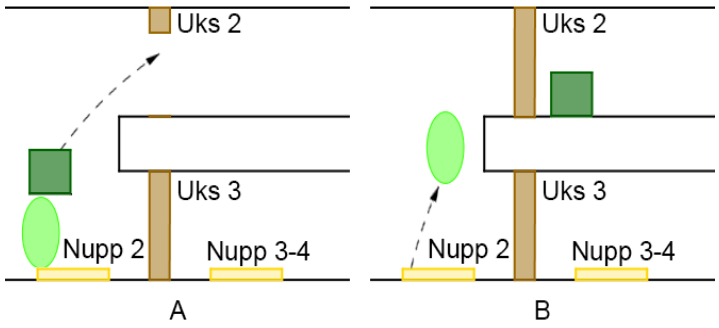
Olukorras kus mängija ei suutnud mõistatusele lahendust leida, oli testijatel võimalik küsida töö autori käest vihje. Vihje saamiseks pidi konkreetse mõistatuse alustamisest olema möödunud kolm minutit sissejuhatavate mõistatuste puhul ja kümme minutit keerulistemate mõistatuste puhul.

5.2. Leitud probleemid ja lahendused

Testimise käigus leiti probleemid mängumehaanikatega, nende tutvustamisega ja teatud mõistatustega. Järgnevalt on kirjeldatud leitud probleemid ning nendele võimalikud lahendused. Suurem osa lahendustest implementeeriti arvutimängu, ülejäänud võimalikke lahendusi ei implementeeritud, kuna need oleksid muutnud mõistatuste disainimise oluliselt keerulisemaks või muutnud mõistatuste lahendamise mängija jaoks liiga lihtsaks.

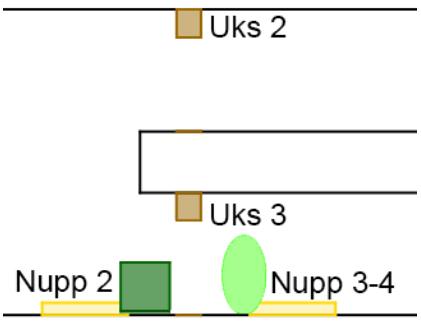
5.2.1. Mõistatus, kus kasutatakse keskkondasid B, gravitatsioonita, aeglane ja elav

Kaks testijat proovisid peatükis 3.3.3. kirjeldatud mõistatuses visata kasti läbi ukse 2 seistes ise nupul 2 (Joonis 32:A). Ehkki kumbki testijatest polnud edukad, on tegu võimaliku probleemse kohaga, sest kui mängijal õnnestub saada see kast läbi ukse 2 (näiteks kasutades gravitatsioonita keskkonda), siis on seda sealt võimatu kätte saada ning mõistatust pole võimalik lahendada (Joonis 32).



Joonis 32: Kasti läbi ukse 2 viskamine.

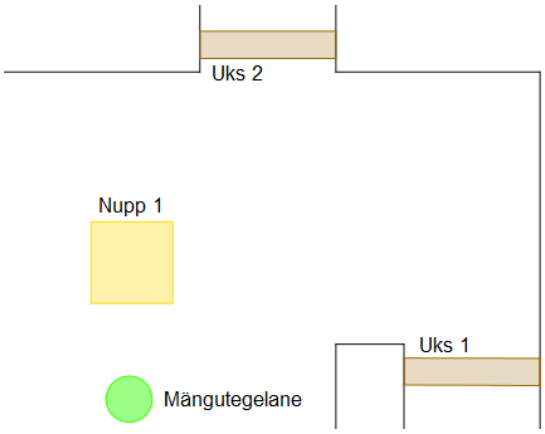
Üks testija leidis antud mõistatusele lahenduse, mis erines töö autori poolt mõistatuse loomisel kavatsetud lahendusest. Üldiselt on alternatiivsete lahendusviiside kasutamine osa mängust, aga see konkreetne testija poolt leitud lahendusviis on mängija suhtes ebaaausalt raske. Peatükis 3.3.3. kirjeldatud mõistatuses ei kasutanud üks testija aeglast keskkonda, et saada nupul 2 paiknev kast läbi ukse 3. Selle asemel paigutas testija kasti nupu servale nii, et see avaks ukse 2 ja oleks võimalikult lähedal uksele 3 (Joonis 33). Seejärel kasutas ta keskkonda B, et võtta kast ja liigutada see läbi ukse 3 enne selle tavakiirusel sulgumist. Antud viisil lahendamise ajal sai testija aru, et tegelikult oleks oluliselt lihtsam kasutada aeglast keskkonda, aga otsustas siiski kasutada keerulisemat lahendusviisi. Antud probleemi lahendamiseks liigutati nupp 2 uksest 3 kaugemale, et aeglase keskkonna kasutamine oleks vajalik ja mängijale paremini arusaadav. Nupu kaugemale viimine tagab selle, et kast asub mängijast piisavalt kaugel ja teeb oluliselt selgemini arusaadavaks, et tavakiirusega seda läbi ukse 3 tuua ei saa.



Joonis 33: Kasti nupu 2 servale asetamine.

5.2.2. Kasti ja ukse nähtavus

Ühel testijal tekkis probleem mõistatusega, millest osa on kujutatud joonisel 34 pealtvaates. Antud mõistatuses avab nupp 1 ukse 1. Selles mõistatuses on mängijale alguses näha üks uks (uks 2) ja üks nupp (nupp 1), aga see konkreetne nupp avas ukse 1, mida mängija ei pruugi esialgu märgata.



Joonis 34: Nupu ja ukse nähtavuse probleemiga mõistatuse osa.

Kuna kasti asetamine nupule ei avanud seda ust, mille poole testija vaatas, siis arvas ta, et ukse avamiseks on vajalik asetada nupule kaks kasti. Samuti ei märganud testija seda ust, mille nupp tegelikult avas, sest see oli juba avatud. Probleemi lahendamiseks oleks võimalik muuta objektide asetust selliselt, et nupuga seotud uks on nupu juurest kohe nähtav. See pole aga parim lahendus, sest see piiraks oluliselt mõistatuste loomisel võimalikku objektide paigutust.

Teiseks tekkis kõikidel tesitjatel probleem mõistatuses, kus kasutati esimest korda nuppu, mis avab ühe ukse ja sulgeb teise. Üks kast ja eelmainitud nupp olid mängijatele nähtaval enne kui uksed. Seetõttu asetasid kõik testijad kasti nupule enne kui nad nägid uksi. Kuna nupp sulges esimese ukse, siis mängijad nägid seda suletud olekus ja neile jäi arusaamatuks, miks nupp ust ei ava. See oli ainus mõistatus, mille juures kõik testijad küsisid vihje. Antud probleemi lahendamiseks lisati sissejuhatav mõistatus, mis tutvustab mängijale ühe nupuga mitme ukse avamist ja/või sulgemist. Mõistatuses on ühes ruumis näha kaks ust, üks neist avatud ja teine suletud. Ruumi keskel paikneb nupp mis avab suletud ukse ja sulgeb avatud ukse.

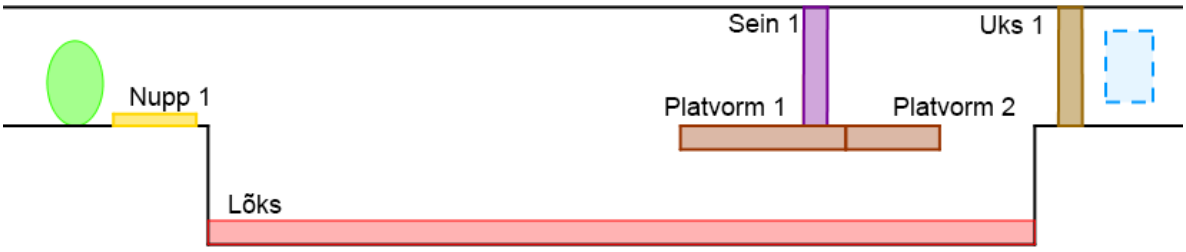
5.2.3. Kastiprobleemid

Kastidega seoses esines testijatel kaks probleemi. Esiteks jäi kahele testijale esialgu arusaamatuks kasti poolläbipaistev kujutis (kui kast ei asu samas keskkonnas nagu mängija). Testijad arvasid, et tegu on reaalse kastiga ja üritasid seda kätte võtta. Mõlemal testijal kulus üle minuti, et antud mängumehaanikast aru saada. Antud probleemile lahendust leida on keeruline, sest tegu on mõistatusega, kus kasti poolläbipaistvat kujutist näidatakse mängijale väga lihtsas olukorras (mängija peab olema keskkonnas B, et mõistatuse alguses näha kasti, mis paikneb keskkonnas A seina taga).

Teiseks häiris ühte testijat see, et kui kast pöörleb enne selle kätte võtmist, siis pöörlemine ei lõppe kasti kätte võttes. Testija väitis, et kasti pöörlemine tõmbab sellele tähelepanu ning ei lase keskenduda ümbritseva keskkonna vaatlemisele. Probleemi lahendamiseks aeglustatakse kasti pöörlemist kuni peatumiseni, kui kast on kätte võetud.

5.2.4. Muud probleemid

Järgmine probleem, mis testimise käigus tekkis oli see, et kaks testijat unustasid ära teatud tegevuste sooritamiseks kasutatavad klaviatuuri klahvid. Üks testija ei mäletanud erinevate keskkondade valimist ja teine portaalide sulgemist ja kiiremat liikumist. Klahvide meelde tuletamiseks lisati mängusisesesse menüüsse nimekiri kasutatavatest klaviatuuri klahvidest ja nendele vastavatest tegevustest.



Joonis 35: Sissejuhatav mõistatus, milles testijatel tekkis sooritusega probleeme.

Testijatel tekkis probleem ka sissejuhatavas mõistatuses, mis on kujutatud joonisel 35. Mängija (roheline ovaal) eesmärgiks on jõuda ala lõppu (sinine ristkülik). Nupp 1 avab 3,3 sekundiks ukse 1. Platvormid 1 ja 2 eksisteerivad ainult keskkonnas B. Sein 1 puudub keskkonnas B. Mõistatuse lahendamiseks peab mängija astuma esialgu nupule 1, seejärel kasutama gravi-

tatsioonita keskkonda, et liikuda platvormi 1 juurde. Selleni jõudmisel peab mängija kasutama keskkonda B, et maanduda platvormil 1 ja pääseda seinast 1 mööda. Platvormil 2 olles peab mängija avama portaali aeglasesse keskkonda ja enne portaali läbimist hüppama platvormil 2. Hüppamine on vajalik, sest aeglases keskkonnas platvormi 2 ei eksisteeri ja mängija kukuks seal lõksuga kokkupuutesse. Aeglase keskkonna kasutamine on vajalik, et jõuda läbi ukse 1 enne kui see sulgub. Kõik testijad said lahenduskäigust kiiresti aru, aga sooritamisel tekitas probleeme platvormil 2 hüppamine enne aeglasesse keskkonda viiva portaali läbimist, kuna antud platvormil on vähe ruumi liikumiseks. Vähim arv katseid mõistatuse sooritamiseks peale lahenduskäigust aru saamist oli 8 katset ja suurim 16 katset. Probleemi lahendamiseks lisati platvorm 2 ka teistesse keskkondadesse, mis tähendab, et hüppamist saab sooritada ka peale portaali läbimist.

Viimaseks kasutasid testijad keskkonda B üldiselt ainult siis kui see oli vältimatu. Näiteks olukord, kus ala lõpp paikneb seina taga, mis eksisteerib teistes keskkondades aga mitte keskkonnas B. Olukorras, kus keskkonna B kasutamine oli valikuline (näiteks keskkonnas B on platvorm, mis võimaldab mängijal lihtsamalt või kiiremini liikuda) kasutati seda harva, sest nendes olukordades ei märganud testijad, et keskkonnas B on erinevus.

Keskkonna B vähese kasutamise jaoks mängijale vihje andmiseks oleks üheks võimaluseks kujutada erinevusi ka teistes keskkondades. Näiteks kui sein eksisteerib keskkonnas A ja puudub keskkonnas B, siis selle kujutamiseks oleks üks võimalus seda seina kujutada keskkonnas A erineva välimusega võrreldes seintega, mis eksisteerivad mõlemas keskkonnas. Teine võimalus oleks anda peale teatud aja möödumist või teatud arvu ala uuesti alustamist mängijale vihje antud mõistatuses vajaliku keskkonna kohta.

Esimene lahendusviis eemaldab aga ühe mängu olulise osa, keskkondade vaheliste erinevuste avastamise, ning võib muuta mõistatuste lahendamise liiga lihtsaks. Teine variant ei ole autori poolt eelistatud, kuna mängumehaanikate ja keskkondade tutvustamine võiks toimuda läbi mängimise, mitte ekraanil kuvatavate vihjete või juhiste kaudu.

6. Kokkuvõte

Käesolev töö kirjeldas kahte keskkondade vaheliste portaalide renderdusalgoritmi ning nende implementeerimiskeerukuse ja ressursikasutuse võrdlust. Lisaks kirjeldati portaalide läbimise algoritmi tekstuurile renderdamist kasutava renderdusalgoritmi puhul.

Töö käigus valmis arvutimäng, milles portaale kasutatakse ühe mängumehaanikana mõistatuste lahendamiseks. Mängijal on võimalik portaale avada suvalisel hetkel ja nende kaudu liikuda mängus oleva viie keskkonna vahel. Keskkonnad disainiti erinevate omadustega (näiteks ühes keskkonnas puudub gravitatsioon) ja erinevate kunstiliste stiilidega. Mängu loodi 17 sissejuhatavat mõistatust, mis tutvustavad mängijale mängumehaanikaid ja erinevaid keskkondi. Sissejuhatavatele mõistatustele järgneb 10 keerulisemat kombineeritud mõistatust, mille lahendamiseks peab mängija kasutama mitut keskkonda korduvalt.

Viimaseks toimus töö käigus loodud avutimängu kasutusmugavuse ja mängumehaanikate testimine. Testimise käigus leitud probleemidele pakuti välja lahendused, mis ka suuremas osas implementeeriti loodud avutimängu. Osadele tekkinud probleemidele võimalikke lahendusi ei implementeeritud, sest need oleksid muutnud mõistatuste disainimise keerulisemaks või mängija jaoks mõistatuste lahendamise liiga lihtsaks.

Töö autor on valminud arvutimänguga rahul ja leiab, et testijatel sissejuhatavate mõistatuste ja nelja keerulisema mõistatuse mängimiseks kulunud aeg (54 kuni 81 minutit) on piisav arvestades mängu disainimiseks ja loomiseks kulunud aega. Edasi on töö autoril plaanis loodud arvutimängu edasi arendada: täiustada mängumehaanikaid ja lisada mängu rohkem keerulisemaid mõistatusi. Pärast edasiarendust tehakse mäng tasuta kättesaadavaks.

7. Viidatud kirjandus

[1] *Portal.* http://www.valvesoftware.com/games/portal.html (10.05.2017)
[2] *Fran Bow.* http://www.franbow.com/index.php (10.05.2017)
[3] *Unity - Game Engine.* https://unity3d.com/ (10.05.2017)
[4] *Unity - Multiplatform.* https://unity3d.com/unity/multiplatform (10.05.2017)
[5] *Unity - Unity Personal.* https://store.unity.com/products/unity-personal (10.05.2017)
[6] *Unity - Unity Plus.* https://store.unity.com/products/unity-plus (10.05.2017)
[7] *Unity - Unity Pro.* https://store.unity.com/products/unity-pro (10.05.2017)
[8] *Unreal Engine Technology.* https://www.unrealengine.com/ (10.05.2017)
[9] *Blueprints Visual Scripting.* https://docs.unrealengine.com/latest/INT/Engine/Blueprints/ (10.05.2017)
[10] *Unreal Engine End User License Agreement.* https://www.unrealengine.com/eula (10.05.2017)
[11] *Godot Engine – Free and open source 2D and 3D game engine.* https://godotengine.org/ (10.05.2017)
[12] *OGRE – Open Source 3D Graphics Engine.* http://www.ogre3d.org/ (10.05.2017)
[13] *Features | OGRE – Open Source 3D Graphics Engine.* http://www.ogre3d.org/about/features (10.05.2017)
[14] *About | OGRE – Open Source 3D Graphics Engine.* http://www.ogre3d.org/about (10.05.2017)
[15] Overvoorde, A. (2015). *OpenGL.* https://www.opengl.org/ (10.05.2017)
[16] Overvoorde, A. (2015). *OpenGL Overview.* https://www.opengl.org/about/ (10.05.2017)
[17] *Narbacular Drop.* https://www.digipen.edu/fileadmin/website_data/gallery/game_websites/NarbacularDrop/ (10.05.2017)
[18] *Rendering recursive portals with OpenGL.* https://th0mas.nl/2013/05/19/rendering-recursive-portals-with-opengl/ (10.05.2017)
[19] Lengyel, E. (2005). *Oblique View Frustum Depth Projection and Clipping*. http://www.terathon.com/lengyel/Lengyel-Oblique.pdf (10.05.2017)
[20] *Unity - Manual: Built-in shader helper functions.* https://docs.unity3d.com/Manual/SL-BuiltinFunctions.html (10.05.2017)
[21] Overvoorde, A. (2015). *OpenGL - Depth and stencils.* https://open.gl/depthstencils (10.05.2017)

Lisad

I. Terminid

Pügamine (*clipping*) Arvutigraafikas nimetatakse pügamiseks objektide või objektide osade eemaldamist renderdusest.
Projektsioonimaatriks (*projection matrix*) Maatriks, mis teisendab koordinaadid kaameraruumist pügamisruumi.
Kaameramaatriks (*view matrix*) Maatriks, mis teisendab koordinaadid maailmaruumist kaameraruumi.
Renderdamine (*rendering*) Arvutigraafikas nimetetakse renderdamiseks kahe- või kolmemõõtmelise mudeli teisendamist kahemõõtmeliseks kujutiseks, mida saab kuvada ekraanil.
Tekstuur (*texture*) Arvutigraafikas kolmemõõtmelisele mudelile rakendatud pilt, mis võimaldab anda mudelile detailsema välimuse (näiteks mudeli pinna värvi).
Tekstuuri vastendamine (*texture mapping*) Arvutigraafikas kahemõõtmelise teksturiseeritud pinna panemist kolmemõõtmelise kujutise peale.
Fragment (*fragment*) Andmed ühe piksli joonistamiseks kaadripuhvrisse.
Varjutaja (*shader*) Arvutigraafikas programm, mis määrab renderdamisel kolmemõõtmelise objekti kujutamisviisi (näiteks objekti pinna värvi vastavalt valguse langemisnurgale).

Fragmendivarjutaja (*fragment shader*) Varjutaja, mida rakendatakse objekti renderdamisel igale fragmendile.
Pügamisruum (*clip space*) Koordinaadisüsteem, milles asuvad objektid, mis on kaamera vaateväljas.
Interpolatsioon (*interpolation*) Funktsiooni vahepealsete väärtuste leidmine selle antud väärtuste põhjal.
Teksel (*texel*) Lühend sõnadest „*texture element*”. Tekstuur koosneb tekslitest.
Kuvasuhe (*aspect ratio*) Kujutise laiuse ja kõrguse vaheline suhe.
Ortograafiline (*orthographic*) Ruumilise eseme projekteerimine tasandile omavahel paralleelsete joontega, mis on risti tasandiga.
Perspektiiv (*perspective*) Ruumilise eseme kujutamine tasandil nii, et säilib mulje ruumilisusest.

II. Käivitusjuhend

Pakkida lahti ZIP-fail „lisa_iv.zip” ja käivitada fail „RealityB.exe”. Avanenud aknas saab vajaduse korral muuta mängu seadistusi. Mängu alustamiseks vajutada avanenud aknas nuppu „Play!”.

Minimaalsed nõuded riistvarale:

Protsessor	Intel® Core™ i5-4310U
Graafikakaart	Intel® HD Graphics 4400
Muutmälu maht	8 GB
Operatsioonisüsteem	Windows 8.1 või uuem.

III. Mängujuhend

Klaviatuuri klahv	**Tegevus**
W	Edasi liikumine
S	Tagasi liikumine
A	Vasakule liikumine
D	Paremale liikumine
Shift	Kiirem liikumine
Vasak hiirenupp	Portaali avamine
Parem hiirenupp	Kasti kättevõtmine
R	Portaali sulgemine
Tühik	Hüppamine
1	Keskkonna A valimine
2	Keskkonna B valimine
3	Gravitatsioonita keskkonna valimine
4	Aeglase keskkonna valimine
5	Elava keskkonna valimine
Escape	Mängusisese menüü avamine/sulgemine

IV. Mäng

Mäng asub tööga kaasas olevas ZIP-failis „lisa_iv.zip”.

V. Lähtekood

Mängu lähtekood on kättesaadav Git repositooriumist, mis asub aadressil: https://bitbucket.org/ever_kalle/realityb

VI. Testsessioonid

Testsessioonide videolindistused on kättesaadavad järgnevatel aadressidel:

Testsessioon 1	https://www.youtube.com/watch?v=aZaZMDxhTn8
Testsessioon 2	https://www.youtube.com/watch?v=2DBLWOJdOdQ
Testsessioon 3	https://www.youtube.com/watch?v=PXbTurP69KU

VII. Litsents

Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks

Mina, **Kalle Ever**,
(*autori nimi*)

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) enda loodud teose
   Keskkondade vaheliste portaalide algoritm ja selle kasutamine arvutimängus,
   (*lõputöö pealkiri*)

   mille juhendajad on Raimond-Hendrik Tunnel ja Margus Luik,
   (*juhendaja nimi*)

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

Tartus, **11.05.2017**