

TARTU ÜLIKOOL

Loodus- ja täppisteaduste valdkond

Arvutiteaduse instituut

Informaatika õppekava

Silver Spitsõn

# Mallide lisamine Blastronaut mängu protseduurilisele generaatorile

Bakalaureusetöö (9 EAP)

Juhendaja: Jaanus Jaggo, MSc

Tartu 2021

## **Mallide lisamine Blastronaut mängu protseduurilisele generaatorile**

### **Lühikokkuvõte:**

Käesolevas bakalaureusetöös täiendatakse Blastronaut mängu protseduurilist generaatorit mallide põhise süsteemiga. See süsteem kombineerib protseduuriliselt genereeritud maailma käsitsi loodud mallidega. Mallide koostamiseks loodi ka redaktor, mis teeb mallide loomise kasutajasõbralikuks ja võimaldab mängijatel ise malle luua. Redaktori hindamiseks viidi läbi kasutaja testimine, mille käigus koostas kasutaja oma malli ja andis sellele protsessile tagasisidet. Töö lõpus hinnatakse malli põhise süsteemi sobivust Blastronaut mängule ja pakutakse võimalusi selle edasi arendamiseks.

### **Võtmesõnad:**

Protseduuriline genereerimine, mallid, mängu arendamine

### **CERCS:**

P175 Informaatika, süsteemiteooria

## **Adding templates to Blastronaut game's procedural generator**

### **Abstract:**

In this Bachelor's thesis the existing procedural generator of the Blastronaut game is improved by adding a template based system. This system combines the procedurally generated world with handmade templates. For creating the templates a redactor was made which brings a user-friendly way of making templates and enables the players to create the templates themselves. In order to assess the editor, a user test was conducted in which the users created their own template and gave feedback to that process. At the end of the thesis, the template based system is evaluated and possible improvements upon it are given.

### **Keywords:**

Procedural generation, templates, game development

### **CERCS:**

P175 Informatics, systems theory

# Sisukord

<b>1</b>	<b>Sissejuhatus</b>	<b>5</b>
<b>2</b>	<b>Maailma protseduuriline genereerimine</b>	<b>7</b>
2.1	Mürafunktsioon . . . . .	9
2.1.1	Simplex müra . . . . .	9
2.1.2	Voronoi diagramm . . . . .	10
2.2	Mallipõhine genereerimine Spelunky mängu näitel . . . . .	11
<b>3</b>	<b>Blastronaudi protseduuriline generaator</b>	<b>14</b>
3.1	Eksisteeriv protseduuriline generaator . . . . .	15
3.2	Mallipõhine süsteem . . . . .	18
3.3	Väikesed mallid . . . . .	19
3.4	Suured mallid . . . . .	20
<b>4</b>	<b>Mallide redaktor</b>	<b>22</b>
4.1	Esimene redaktor . . . . .	22
4.2	Väikese malli redaktor . . . . .	24
4.3	Suure malli redaktor . . . . .	25
<b>5</b>	<b>Redaktori kasutajakogemuse testimine</b>	<b>27</b>
<b>6</b>	<b>Kokkuvõte</b>	<b>31</b>
	<b>Kasutatud materjalid</b>	<b>33</b>
	<b>Lisad</b>	<b>35</b>
	I. Küsitlus . . . . .	36
	II. Litsents . . . . .	39

# 1 Sissejuhatus

Paljudes mängudes on oluline osa mängust mitmekesine mängumaailm. Seda on mängijal põnev avastada ning seeläbi teeb see mängu huvitavaks. Väiksemate mängude puhul ei pruugi käsitsi maailma loomine väga keeruline olla, sest see on piisavalt väike. Suurema maailma loomiseks on vaja teha palju tööd, et see objektidega täita, sest need muudavad selle mängija jaoks huvitavaks. See protsess on aga aeganõudev ning ei ole sobilik mängude jaoks, kus on eriti suured või lõpmatud maailmad. Alternatiivseks lahenduseks on kasutada protseduurilist genereerimist, mis tähendab, et maailmad luuakse algoritmiliselt. Lisaks on protseduuriliselt genereeritud maailmadel see eelis, et iga kord mängu uuesti alustades on võimalik ka uus maailm genereerida. See annab mängijale põhjuse mängu mitu korda mängida, ilma et see tema jaoks igavaks muutuks [1, 2].

Blastronaut<sup>1</sup> on kahemõõtmeline kõrvalt vaates seiklusmäng. Mängija saab lennata maailmas ringi, hävitada maastikut ja kaevandada ressursse. Selleks on ta käsutuses spetsiaalne relv, mis laseb plahvatavat geeli. Geel kleepub plokkide külge ning plahvatab lühikese aja jooksul. Kogutud ressursse on võimalik spetsiaalsete masinate juures maha müüa. Saadud raha eest on võimalik uut varustust osta. Mängul on ka mitmikmängu tugi olemas, mis võimaldab sõpradel üle võrgu mängida. Joonisel 1 on välja toodud ekraanitõmmis Blastronaut mängust. Mängu arendatakse Godot<sup>2</sup> mängumootoriga. Antud töö ajal arendasid mängu töö autorile lisaks veel kaks inimest. Mitmikmängu tugi lisati 2020. aasta sügissemestril aine “Tarkvaraprojekt” raames.

Käesoleva bakalaureusetöö raames täiendatakse protseduurilise genereerimise süsteemi mallipõhise genereerimisega, mis võimaldab tükke maailmast käsitsi valmis teha ja neid maastiku genereerimisel sinna paigutada. Selle abil on võimalik maailm mängija jaoks huvitavamaks muuta, et tal oleks põnevam seda avastada. Samuti valmis selle töö käigus mallide loomise redaktor, mis teeb mallide loomise kasutajasõbralikuks ja annab

---

<sup>1</sup><https://store.steampowered.com/app/1392650/BLASTRONAUT/>

<sup>2</sup><https://godotengine.org>



Joonis 1. Ekraanitõmmis Blastonaut mängust.

ka mängijatele võimaluse ise malle luua.

Töö teises peatükis kirjeldatakse lühidalt erinevaid protseduurilise genereerimise meetodeid ning vaadeldakse põhjalikumalt kahte meetodit, mida Blastonaut mängus kasutatakse. Kolmandas peatükis kirjeldatakse olemasolevat protseduurilist generaatorit ning autori poolt loodud mallide põhjal töötavat süsteemi. Samuti kirjeldatakse selles peatükis täpsemalt mallide struktuuri. Neljandas peatükis kirjeldatakse redaktorit, mille abil saab malle luua. Viimases peatükis tuuakse välja kasutajate tagasiside redaktorile ning antakse ülevaade valminud tööst.

## 2 Maailma protseduuriline genereerimine

Selles peatükis kirjeldatakse mõningaid protseduurilise genereerimise meetodeid. Samuti kirjeldatakse lühidalt mürafunktsioone ning mallidel põhinevat protseduurilist genereerimist Spelunky mängus.

Protseduurilise genereerimise jaoks on olemas erinevaid võimalusi. Alexander Gellel ja Penny Sweetser [3] on välja toonud järgmised meetodid:

1. **Markovi ahelad** koosnevad olekutest  $S = \{s_1, s_2, s_3, \dots, s_n\}$  ja tõenäosusjaotusest  $P(S_t|S_{t-1})$ , mis näitab tõenäosust, et jõuti olekusse  $S_t$ , kui eelmine olek oli  $S_{t-1}$ . Näiteks on antud meetodit kasutatud mängu Super Mario Bros. tasemete genereerimiseks [4].
2. **Rakuautomaadid** (ing. k. *cellular automata*) koosnevad homogeensetest ühikutest ehk rakkudest. Iga ajaühiku tagant uuendatakse rakke ning iga rakk arvestab arendamisel enda ümbritsevate rakkude olekuid [5]. Näiteks kasutatakse sarnast süsteemi maailma simuleerimiseks mängus Noita [6].
3. **Formaalsed grammatikad** on reeglite kogumid, kus määratakse mitteterminaalide ja terminaalsümbolite abil, kuidas järgmisesse olekusse minna. Näiteks on kasutatud formaalse grammatika mudelit, et genereerida rollimängudele (ing. k. *Role-playing games*) omaseid ülesandeid, mis antakse mängijale [7].
4. **Masinõppe tehnikad** vajavad palju eelnevaid andmeid, mille põhjal mudelit treenida, et see suudaks siis mingite sisendite puhul väljundit ennustada. Näiteks on kasutatud masinõpet mängu The Legend of Zelda tasemete genereerimiseks [8].
5. **Evolutsioonilised algoritmid** (ing. k. *evolutionary algorithm*) põhinevad evolutsiooniteoorial ning kasutavad sobivusfunktsioone, et hinnata igat lahendust ja tekitada uued lahendused, võttes aluseks eelnevalt parimateks hinnatud lahendused [9].

Kuna Blastronaut mängus saab mängija maailma hävitada ja seetõttu kiiresti edasi liikuda, seab see protseduurilise genereerimise süsteemile järgnevad nõuded:

1. Generaator peab suutma dünaamiliselt maailma suurendada, sest mängija saab väga kiiresti mööda maailma liikuda. Limiteeritud suurusega maailma läbimine oleks liiga kiire ning muudaks mängumaailma avastamise aspekti mängija jaoks igavaks.
2. Maailma suurendamine ei tohi liiga kaua aega võtta, sest kui mängija liiga kiiresti liigub ei jõuaks generaator talle järgi. Siis näeks mängija tühja maailma ja kuidas tükke ükshaaval juurde genereeritakse. See rikub mängija jaoks illusiooni ning on ebamugav. Kuna mäng sisaldab ka mitmikmängu võimalust, kus mänguserveri looja (ing. k. *host*) arvuti peab ka kõikide teiste mängijate jaoks maailma genereerima, siis on generaatori efektiivsus väga oluline.
3. Loodud maailm peab olema visuaalselt huvitav ja mitmekülgne. See tähendab, et generaator peab piisavalt malle paigutama nii, et mallid sulanduksid ühtlaselt maailma ja muudaksid mängija jaoks maailma huvitavamaks. Samas peab seda tegema nii, et liiga palju sarnaseid objekte lähestikku ei satuks, mis maailma liiga üksluiseks muudaksid.
4. Generaator peab olema deterministlik, ehk sama sisendi korral tuleb luua alati samasugune maailm. Sisendiks on kasutaja poolt määratud seeme (ing. k. *seed*), mida kasutatakse juhuarvude genereerimiseks. Antud juhuarvud genereeritakse koordinaadi põhjal. Ehk sama seemne korral genereeritakse sama koordinaadi puhul alati sama juhuarv. See on eriti oluline mitmikmängu puhul, kus mängijad võivad läheneda ühele punktile erinevatest suundadest. On väga oluline, et sellisel juhul genereeritakse mängijate jaoks tükid täpselt samas järjekorras.

Järgnevalt kirjeldatakse meetodeid, mida kasutatakse Blastronaut mängus: mürafunktsioon ja mallipõhine protseduuriline genereerimine.

## 2.1 Mürafunktsioon

Mürafunktsioon on n-kohaline deterministlik funktsioon, mis tagastab iga n-kohalise reaalarvude vektori puhul mingi juhusliku reaalarvu. Üksik mürakiht ei ole tihtipeale piisavalt mitmekesine ning selle jaoks liidetakse mitu kihti kokku, et saavutada huvitavam tulemus. Mürafunktsioone on palju erinevaid, näiteks väärtusmüra, Perlini müra ja Worley müra. Mürasid kasutatakse tihti mängudes, kuhu on tarvis luua loodusliku välimusega keskkondi. Näiteks on kasutatud mürafunktsioone maailma genereerimiseks ja ressursside paigutamiseks mängus Factorio<sup>3,4</sup>.

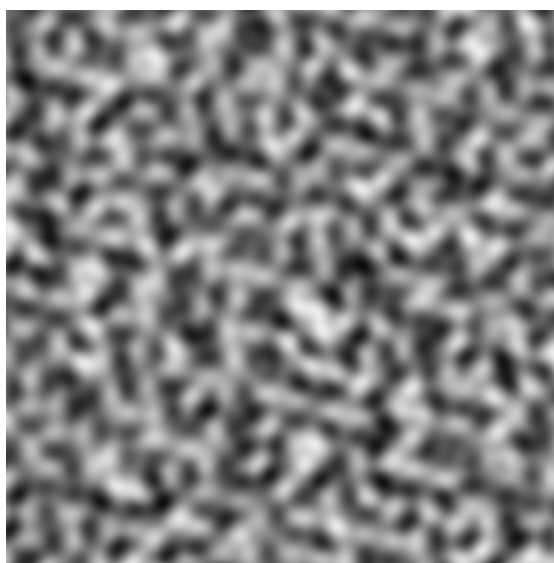
### 2.1.1 Simplex müra

Simplex müra on tuntud gradientmüra algoritm, millel on korrapäratu, kuid sujuv välimus ning mida on kujutatud joonisel 2. Selleks, et seda müra veel huvitavamaks muuta, saab kokku sujutada mitu erineva sagedusega kihti Simplex müra. Simplex müra autoriks on Ken Perlin, kes selle algoritmi esialgu patenteeris.

---

<sup>3</sup><https://factorio.com/blog/post/fff-112>

<sup>4</sup><https://factorio.com/blog/post/fff-258>



Joonis 2. Näidis Simplex mürast [10].

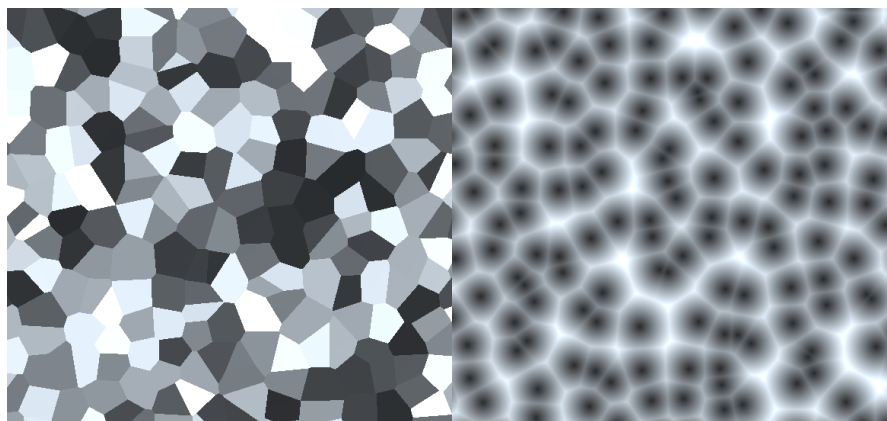
OpenSimplex müra on vabavaraline ja natuke muudetud versioon Ken Perlini Simplex mürast [11]. Godot mängumootoris on see teek sisse ehitatud ja seetõttu kasutatakse seda Blastronaut mängus. Selle müra abil määratakse, milline osa maailmast on plokkidega täidetud ja milline on tühi.

### 2.1.2 Voronoi diagramm

Voronoi diagramm koosneb juhuslikult jaotatud alguspunktidest tasandil. Iga alguspunkt määrab enda ümber tasandi piirkonna, millesse kuuluvad punktid on vastavale algpunktile lähemal kui ühelegi teisele punktile. Neid piirkondi nimetatakse Voronoi rakkudeks. Lihtsustatult öeldes, tekib iga kahe alguspunkti vahele ristsirge, kusjuures selle sirge iga punkt asub võrdsel kaugusel kõikidest selle sirge algpunktidest. Igale rakule saab määrata juhusliku väärtuse [12]. Joonisel 3 on vasakul toodud näidis Voronoi diagrammist, kus rakud on värvitud erinevate halltoonidega vastavalt sellele, mis väärtus rakul on.

Worley müra puhul on samuti tasandil juhuslikult jaotatud alguspunktid ning iga

tasandi punkt kuulub kõige lähemal asuva alguspunkti ümbritsevasse piirkonda. Nüüd, aga ei määrata sellele piirkonnale juhuslikku väärtust vaid vaadatakse iga punkti kaugust lähimast alguspunktist [13]. Joonisel 3 paremal on toodud näidis Worley mürast, kus iga tasandi punkt on värvitud vastavalt sellele, kui kaugel ta lähimast alguspunktist asub.



Joonis 3. Vasakul on näidis Voronoi diagrammist, paremal on näidis Worley mürast [14].

Blastronaut mängus kasutatakse Voronoi diagrammi, et määrata, kus ressursid paiknevad. Selleks vaadatakse raku väärtust ning selle järgi valitakse ressurss, mida antud positsioonile paigutada. Worley müra kasutatakse selleks, et ressursid paigutada ainult teatud kaugusele raku keskmest, mitte terve raku ulatuses.

## 2.2 Mallipõhine genereerimine Spelunky mängu näitel

Mallipõhine genereerimine tähendab seda, et maailma paigutatakse inimeste poolt valmis tehtud tükke. Neid tükke saab maailma paigutada erinevatel viisidel ning järgnevalt on kirjeldatud, kuidas seda tehakse Spelunky mängus.

Spelunky on kahemõõtmeline kõrvalt vaates platvormimäng, kus mängija peab jõudma taseme lõppu ning tee peal on erinevad takistused ja vaenlased. Mängu tasemed koosnevad ruudustikest, kus iga ruut on mingi osa maastikust või mingi objekt. Joonisel 4 on välja toodud osa ühest võimalikust tasemest.



Joonis 4. Ekraanitõmmis mängust Spelunky [15].

Iga tase koosneb 16 erinevast ruumist, mis on esitatud 4 x 4 maatriksina. Maatriksi elementideks on täisarvud, kus iga täisarv määrab ära ruumi tüübi. Selleks tüübiks võib olla näiteks taseme algus, taseme lõpp, vasakule ja paremale suunduvate avadega ruum, kinnine ruum jne. Ruumid genereeritakse nii, et leidub tee algusest lõppu. Iga ruum koosneb omakorda veel 8 x 10 maatriksist, mis määrab täpselt, millised objektid selles ruumis on. Ruumid on käsitsi tehtud, kuid nende sisu ei ole üheselt määratud. Osad ruumide elemendid määratakse juhuslikult alles mängusiseselt, ehk sama ruum võib esineda mängus väikeste variatsioonidega.

Joonisel 5 on toodud näide ühest võimalikust ruumist. Antud esituses tähistab 0 tühja kohta, kus mängija saab liikuda, 1 tahket plokki ja 6 mingit juhuslikult valitud malli, mille suurus on 5 x 3 ruutu. Paremal on see ala tähistatud punasega. Number 2 tähistab ruutu, kus on võrdne võimalus, et genereeritakse tahke plokk või jäetakse see tühjaks. Paremal on need tähistatud valgete ruutudega. Kui sama tüübiga ruum uuesti genereerida,



Joonis 5. Spelunky näidisruum. Vasakul on näidatud ruumi andmed numbriliselt ning paremal sellele vastav ruum mängus [16].

siis see juhuslikult valitud mall ja tõenäosusega määratud plokid muutuvad, mis tagavad mitmekülgse taseme. Kõige viimase asjana paigutatakse genereeritud tasemes veel eraldi algoritmide järgi vaenlased [17].

Mallidel põhineva süsteemi eelis on see, et juhuslikult genereeritud maailma sisse on võimalik paigutada käsitsi tehtud ruume või struktuure. See muudab maailma huvitavamaks ning samuti saab sellega paremini anda edasi mängu teematikat, luues huvitavaid struktuure mallide näol. Samuti on selline süsteem hea, sest seda saab eksisteeriva generaatori külge lisada, ilma et peaks selle töökäiku muutma. Blastronaut mängus on maastik juba mürade abil genereeritud ning mallidel põhineva süsteemi lisamine ei vaja eelneva süsteemi ümbertegemist.

### 3 Blastronauti protseduuriline generaator

Blastronaut kasutab OpenSimplex müra maastiku genereerimiseks ning Voronoi diagrammi ja Worley müra ressursside paigutamiseks. Algne protseduuriline genereerimine on tehtud Jaanus Jaggo poolt. See genereerib maailma, paigutades sinna mürade abil erinevaid ressursse ja bioome. Käesoleva töö eesmärk on täiendada seda generaatorit selliselt, et see paigutaks sinna ka käsitsi loodud kohtasid. Selleks täiendati protseduurilist generaatorit mallipõhise süsteemiga. Esimene lähenemine mallide lisamiseks oli kasutada sarnast süsteemi nagu mängus Spelunky. Suureks erinevuseks nende mängude vahel on see, et Spelunky mängus on tasemed piiratud suurusega ja koosnevad ainult mallidest. Blastronaut mängus on aga lõpmatult suured ning mõned kohad on genereeritud mürade ja mõned mallide abil. Küll aga peavad mallid maailma hästi sulanduma, nagu see on kirjas eelnevalt püstitatud nõuetes.



Joonis 6. Maailma komponendid. Vasakult paremale plokk, tükk ja ülemtükk.

Mängumaailm koosneb kindla suurusega tükkidest (ing. k. chunk), mis omakorda koosnevad plokkidest. Üks tükk on 8 ploki pikkuse küljega ruut. Samuti on olemas ülemtükkid, mis koosnevad 8 x 8 tükkidest. Joonisel 6 on näidatud, kuidas plokkid asuvad tükkides ja tükid omakorda ülemtükkides.

Maailma hoiustamiseks kasutatakse Godot mootori poolt implementeeritud andmestruktuuri, milleks on rasterkaart (ing. k tilemap). Antud andmestruktuur on visualiseeritud kui suur ruudustik, kus igal ruudul on kindel suurus ja koordinaat. Iga ruudu sisuks on täisarv, mis määrab selle ruudu välimuse, mida mängija mängus näeb.

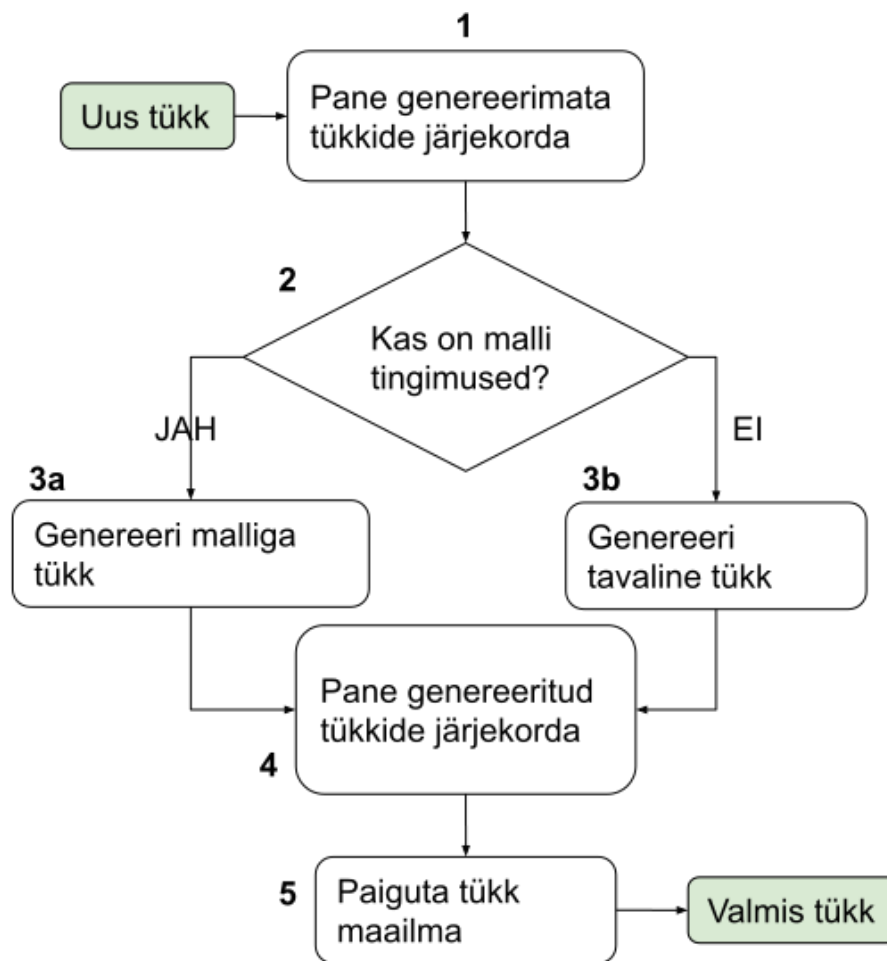
Blastronaut mängus on samaaegselt kasutusel mitu rasterkaarti. Järgnevalt on need rasterkaardid välja toodud:

- Tahkete plokkide rasterkaart. Siin hoitakse mängijale nähtavaid tahkeid plokkide.
- Tagumiste mitte-tahkete plokkide rasterkaart.
- Plokkide peal asuva taimkatte rasterkaart.
- Andmete rasterkaart, kus määratakse, kas antud koordinaatidel asub tahke plokk või mitte. Seda rasterkaarti mängija ei näe.

Järgnevalt kirjeldatakse maailma genereerimise protsessi.

### **3.1 Eksisteeriv protseduuriline generaator**

Maailma genereeritakse mängija ümber tüki kaupa. Tüki genereerimise protsess on kujutatud joonisel 7 ning see koosneb mitmest osast. Plokkide genereerimiseks kasutatakse OpenSimplex müra ja Voronoi diagrammi. OpenSimplex müra abil määratakse, millised tükid on tahked ja kus asub tühjus. Voronoi diagrammi abil määratakse tükkide bioomid. Iga tüki puhul vaadatakse, millises rakus ta asub ning antud tüki bioomiks määratakse selle raku bioom. Samuti kasutatakse Voronoi diagrammi ressursside maailma paigutamiseks.



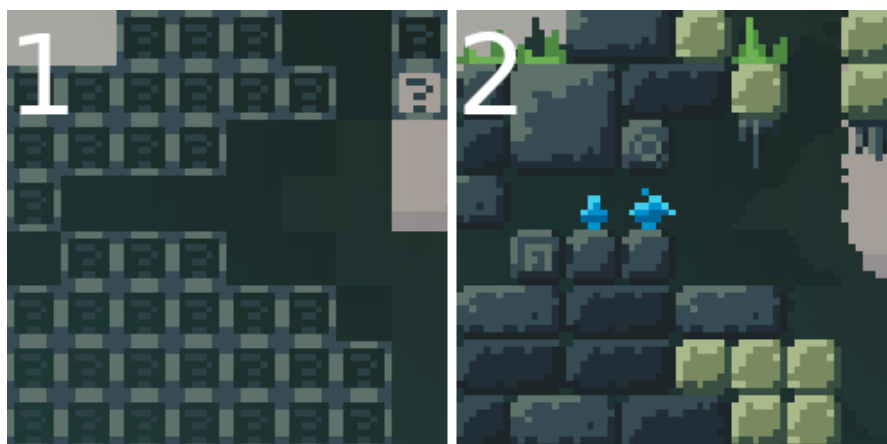
Joonis 7. Skeem maailma generaatori töö järjekorrast.

Järgnevalt on kirjeldatud joonisel 7 numbritega tähistatud tegevused:

1. Vaadatakse kõiki mängija ümber olevaid tükke, mis jäävad mängija ekraanile. Kui mõni tükkidest on genereerimata, pannakse see genereerimata tükide järjekorda.
2. Järjekorrast võetakse tükk ning leitakse ülemtüki koordinaat, mille sees antud tükk asub. Seejärel genereeritakse selle ülemtüki koordinaatide abil juhuslik number, mille abil otsustatakse, kas antud kohale genereeritakse malli järgi tükk või kasutatakse mürapõhist maastiku generaatorit.

3. (a) Siin kasutatakse autori poolt loodud malli põhist süsteemi tüki genereerimiseks, millest räägitakse täpsemalt peatükis 3.2.  
(b) Tavalise tüki genereerimisel arvutatakse müra väärtus kõikidele plokkidele selle tüki sees. Kui müra väärtus on suurem on etteantud piirist, siis paigutatakse antud kohale plokk, vastasel juhul jäetakse see tühjaks. Siinkohal ei valita veel täpselt milline plokk antud kohale paigutatakse, vaid märgitakse see plokk lihtsalt täidetuks. Samal viisil otsustatakse ka tagumisel rasterkaardil ehk taustal paikneva ploki sisu. Eesmised ehk tahked plokid pannakse hetkel andmete rasterkaardi sisse, aga tagumised plokid paigutatakse juba õigesse rasterkaarti. Joonisel 8, pildil 1 on antud andmete rasterkaart nähtavaks tehtud.
4. Genereeritud tükk pannakse uude järjekorda. See järjekord hoiab kõiki tükke, mis on genereeritud, aga mida ei ole veel maailma paigutatud. Samuti tagatakse, et enne sellesse järjekorda lisamist, on ka kõik selle tüki naabertükid genereeritud.
5. Antud sammus käiakse uuesti kõik tüki plokid läbi ning otsustatakse, milline on ploki lõplik välimus. Selleks tuleb arvestada ka kõrvalolevaid plokkide, sest mängus on kujutatud ka pinnase tükke, mis koosnevad mitmest plokkist ning need on vaja õigesti kokku panna (vt. joonis 8, pilt 2). Samuti silutakse selle sammu sees ka taustaplokkide välimust. See tähendab, et kahe erineva ploki vahele asetatakse sujuva üleminekuga plokid, nii et üleminek ei tunduks liiga järsk. Joonisel 8, pildil 2 on näha, kuidas taustaplokkid ei ole enam ruudukujulised, nagu nad olid joonisel 8, pildil 1. Viimasena paigutatakse mitte-tahked objektid plokkide alla või peale, näiteks seemed, kristallid, väädid jms. Joonisel 8, pildil 2 on toodud näide seentest, stalaktiididest ning murust.

Põhjus, miks genereerimist tehakse kahes etapis ja pannakse kaks korda järjekorda on see, et teises etapis võrreldakse ka kõrvalolevaid plokkide. Kuna osa nendest plokkidest



Joonis 8. Number 1 on visualiseeritud andmete rasterkaart. Number 2 on sama tükk peale plokkide paigutamist.

jäävad hoopis kõrval oleva tüki sisse siis on vaja enne tahkete plokkide välimuse määramist kõik ümbritsevad tükid ära genereerida, et saaks alati võrrelda ka kõrvalolevaid plokkide.

### 3.2 Mallipõhine süsteem

Autori poolt loodud generaator kasutab malle. See tähendab, et mängu arendaja loob mingid mallid, mida siis generaator maailma paigutama hakkab. Malle on kahte tüüpi - väikesed ja suured mallid. Generaatori loomisel otsustas autor malli põhise süsteemi kasuks, sest see võimaldab erinevaid struktuure kokku panna, ilma et oleks vaja tervet seda struktuuri eelnevalt käsitsi luua. Kuna kogu genereerimise süsteem toimib tükkhaaval, siis oli mõistlik määrata väikese malli suuruseks terve tükk. Suure malli suurus valiti nii, et oleks võimalik luua piisavalt huvitavaid struktuure.

Mängu sees hoitakse malle objektidena, mis sisaldavad andmevälju ning meetodeid nende andmete töötlemiseks. Mängu väliselt salvestatakse need mallid JSON failidena.

### 3.3 Väikesed mallid

Väikesed mallid on ühe tüki suurused ehk need koosnevad 8 x 8 plokkidest. Malli andmeteks on kaks massiivi, mis tähistavad eesmisi ja tagumisi ehk tausta plokke. Antud massiivid on 8\*8 ehk 64 elemendi pikkused ning need on täidetud tühjade sõnedega, kui antud positsioonil plokki ei ole. See tähendab, et massiivide pikkus on alati sama, mis võimaldab sealt indeksi järgi plokki leida. Veel on mallil nimi ja kategooria. Nime järgi on kasutajal lihtsam neid redaktoris töödelda ning samuti on nimi vajalik suurtes mallides. Kategooria abil on võimalik redaktoris malle filtreerida. Joonisel 9 piltidel 1 ja 2 on välja toodud üks väike mall redaktoris kui ka maailma paigutatuna.



Joonis 9. Väikese malli näidised. Pildil 1 on mall redaktoris, pildil 2 on sama mall maailmas. Pildil 3 on malli edasiarendus redaktoris, pildil 4 on sama mall maailmas.

Selliste mallide loomine ei ole aga kõige parem lahendus, sest antud mall ei pruugi ülejäänud maailmaga hästi sulanduda, mis muudab mängija jaoks maailma koledamaks.

Joonisel 9 piltidel 3 ja 4 on välja toodud edasiarendus joonisel 9 piltidel 1 ja 2 välja toodud mallist. Siin on osaliselt mall asendatud teiste plokkidega, mis võimaldavad sellel paremini maailma sulanduda. Kõige ülemised tumedad viirutatud plokid tähistavad juhuslikku tahket plokki. Läbipaistva taustaga viirutatud plokid tähistavad kohta, kus rakendatakse tavalist ploki genereerimise algoritmi, mida kirjeldati peatükis 3.1. See tähendab, et antud kohale võidakse paigutada juhuslik tahke plokk, aga see võib ka tühjaks jääda.

### **3.4 Suured mallid**

Suured mallid on ühe ülemtüki suurus. See tähendab, et need koosnevad 8 x 8 tükkidest ehk väikestest mallidest. Joonisel 10a on välja toodud osa suurest mallist JSON formaadis. Suurel mallil on nimi ning järjend nimega “templates”. Antud järjendis tähistab iga element ühte tükki ülemtüki sees. Igal tükil on lokaalne koordinaat, väikese malli nimi, mis sellel positsioonil asub ning kas antud tükki on vaja peegeldada või pöörata. Antud näidisele vastava malli kujutis redaktoris on välja toodud joonisel 10b.



(a) Väljavõte suurest mallist JSON formaadis.

(b) Suure malli näidis redaktoris.

Joonis 10. Suure malli näidis.

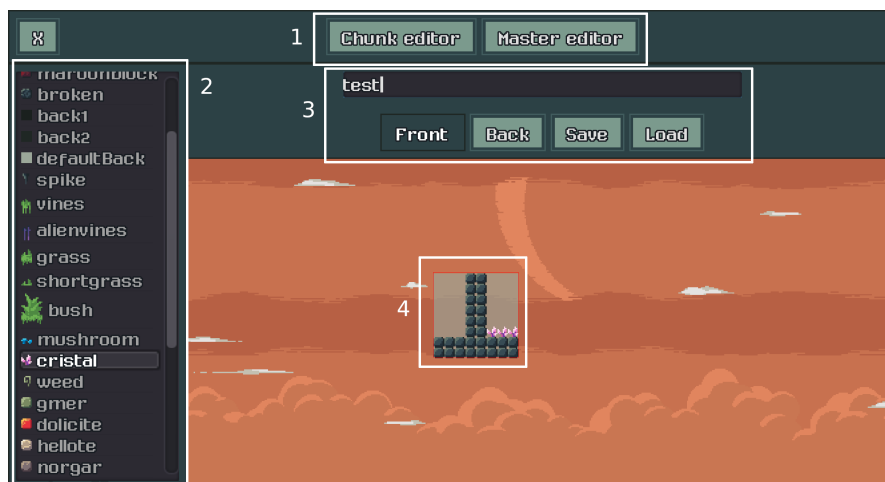
Võrreldes väikese malli struktuuriga, ei ole antud järjend alati täidetud, vaid seal hoitakse ainult neid tükke, kus on mingi väike mall sees - ülejäänud tükide puhul ei ole vaja neid eraldi suure malli sees hoida.

## 4 Mallide redaktor

Generaatorit luues oli süsteemi katsetamiseks vaja malle, mis tuli teha käsitsi. Kuna mallid on JSON failid, siis pidi need lihtsalt käsitsi valmis kirjutama. See aga oli väga tüütu ning ainult plokkide nimede põhjal oli raske ette kujutada, milline mall reaalset välja näeb. Oli vaja kasutajasõbralikku lahendust, mille abil malle luua. Selleks lõi autor mallide redaktori, kus saab vastavaid malle luua ning neid JSON formaadis salvestada. Redaktor võimaldab muuta ka teiste mänguelementide parameetreid, näiteks saab määrata iga bioomi jaoks eraldi plokkid, mis selles bioomis genereeritakse, aga see funktsionaalsus ei ole autori poolt implementeeritud. Järgnevalt räägitakse täpsemalt autori tehtud osadest ehk väikese ja suure malli vahelehest antud redaktoris. Esmalt kirjeldatakse lühidalt redaktori esimesest versiooni ning selle puudusi ja siis räägitakse redaktori teisest versioonist täpsemalt.

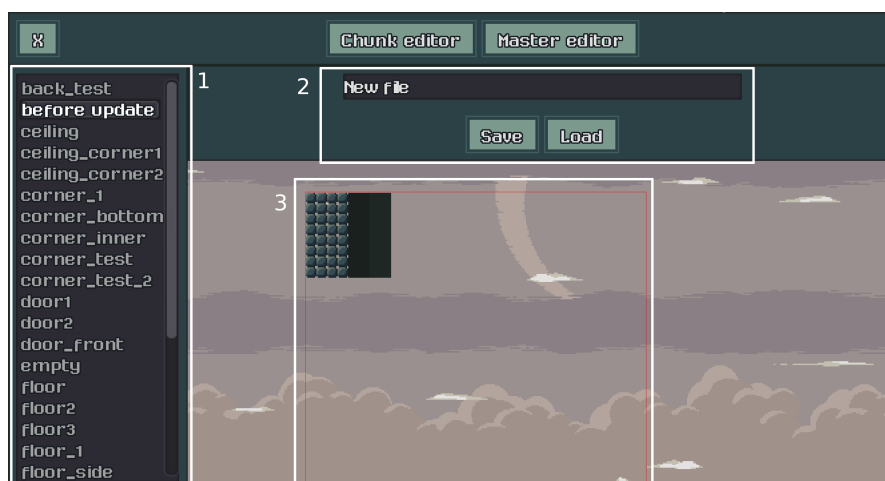
### 4.1 Esimene redaktor

Esimene redaktori versioon koosnes kahest vahelehest, kus oli võimalik luua väikeseid ja suuri malle. Joonisel 11 on välja toodud vaade väikese malli vahelehest redaktori esimeses versioonis koos nummerdatud elementidega. Number 1 tähistab vahelehtede valikut. Number 2 tähistab plokkide loetelu, mida kasutaja saab valida ja väikese malli sisse paigutada. Number 3 tähistab praeguse malli nime ning salvestamise ja laadimise nuppe. Samuti sai siin valida, kas plokk paigutatakse eesmisele või tagumisele rasterkaardile. Number 4 tähistab tükki, mille sisse kasutaja plokkide paigutab.



Joonis 11. Esimese redaktori väikese malli vaheleht.

Joonisel 12 on välja toodud vaade suure malli vahelehest redaktori esimeses versioonis koos nummerdatud elementidega. Number 1 tähistab tükide loetelu, mida kasutaja saab suurde malli paigutada. Number 2 tähistab praeguse malli nime koos salvestamise ja laadimise nuppudega. Number 3 tähistab suurt malli, mille sisse kasutaja tükke paigutab.



Joonis 12. Esimese redaktori suure malli vaheleht.

Redaktori loomisel ilmnusid mõned puudujäägid. Näiteks ei saanud plokkide ega tükke kuidagi filtreerida ning malli salvestamine ja laadimine oli kohmakas. Nende vigade parandamiseks ja teiste redigeeritavate elementide lisamiseks tehti redaktorist teine versioon. Järgnevalt kirjeldatakse uue redaktori väikese ja suure malli vahelehti.

## 4.2 Väikese malli redaktor

Üldine redaktori struktuur on kõikide vahelehtede jaoks sama. Joonisel 13 on näidatud väikese malli vaheleht koos nummerdatud elementidega. Number 1 tähistab failide või objektide loetelu. Number 2 tähistab nuppe, millega on antud objektidele võimalik luua, duplikeerida või kustutada. Number 3 tähistab loetelus olevate objektide kategooriaid, millega on loetelu võimalik filtreerida. Number 4 tähistab hetkel valitud objekti detailset vaadet, kus on võimalik selle objekti nime ja teisi parameetreid muuta. Ekraani keskosa kasutavad hetkel ainult mallide redigeerimise vahelehed.

Number 5 tähistab erinevate plokkide loetelu. Siit loetelust valib kasutaja ploki, mida ta soovib malli sisse paigutada. Seda loetelu on ka võimalik plokkide kategooriate abil filtreerida. Võrreldes redaktori esimese versiooniga, ei ole enam eraldi nuppe, mille abil



Joonis 13. Ekraanitõmmis väikese malli redaktori vaatest.

valida kummale rasterkaardile plokk paigutatakse. Selle asemel on nüüd automaatne süsteem, et kui kasutaja valib “back” kategooria, siis paigutatakse plokk tagumisele rasterkaardile. Number 6 tähistab tükki, mida kasutaja redigeerib.

Väikese malli loomiseks on vaja kõigepealt luua uus fail number 1. tähistatud nupu abil. Seejärel valib kasutaja number 4. tähistatud loetelust mõne ploki ning joonistab hiirega selle ekraani keskel asuva punase ruuduga tähistatud tüki sisse. Iga muudatuse järel salvestatakse automaatselt mall ära ning kasutaja ei pea eraldi salvestama oma tehtud tööd.

### 4.3 Suure malli redaktor

Joonisel 14 on toodud vaade suure malli vahelehest koos nummerdatud elementidega. Vasakul ja paremal pool ekraanist on samad elemendid nagu väikese malli vaate puhul. Number 1 tähistab nüüd tükide loetelu. Number 2 tähistab sätteid, millega on võimalik hetkel valitud tükki peegeldada nii x-telje kui ka y-telje suhtes või pöörata seda 90° kaupa vahemikus -180° kuni 180°. Peegeldamise operatsioon sooritatakse alati enne pööramist.

Nende sätete abil on võimalik ühte väikest malli mitmel erineval moel kasutada ja kasutaja peab vähem väikeseid malle joonistama. Number 3 tähistab suurt malli, mida kasutaja hetkel redigeerib.



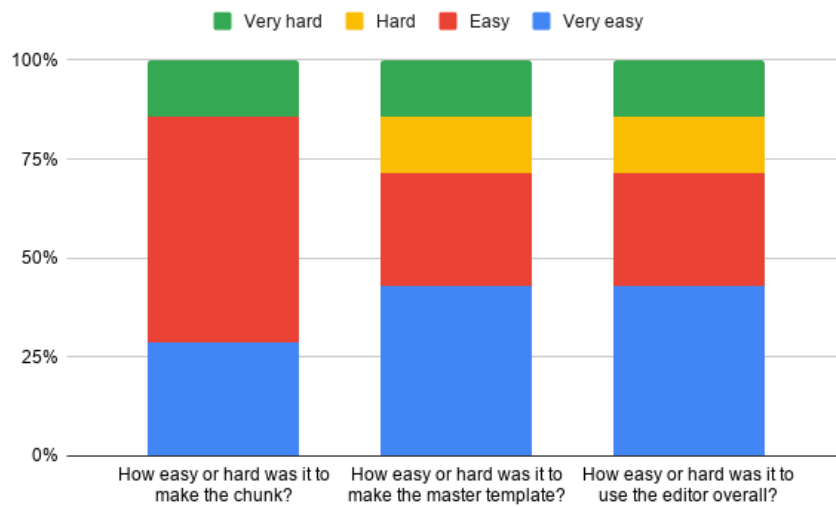
Joonis 14. Ekraanitõmmis suure malli redaktori vaatest.

Pegeldamise ja pööramise operatsioonid tehakse tüki joonistamise ajal ehk neid väikeseid malle, mida kasutaja joonistab, ei muudeta. Suure malli salvestamisel kirjutatakse malli nime kõrvale veel tõeväärtused selle kohta, millised peegeldused tehti ning pöörde operatsioon salvestatakse kraadidena. Kuna suur mall ei mahu täielikult ekraanile ära, siis on siin võimalik kaamerat liigutada.

## 5 Redaktori kasutajakogemuse testimine

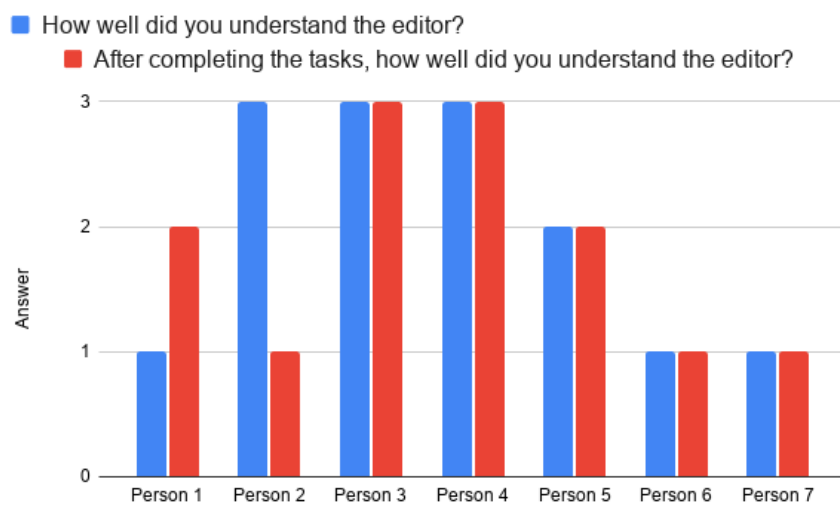
Töö käigus viidi läbi küsitlus, mille eesmärgiks oli hinnata mallide loomise protsessi kasutajakogemust. Küsitlus tehti inglise keeles, sest siis saavad ka mitte eesti keelt kõnelevad inimesed sellele vastata. Küsitluse ankeet on toodud lisa 2. Küsitlusele vastas seitse inimest, kelle seast viis õppisid informaatika erialal. Esmalt paluti kasutajal luua väike mall ning seejärel suur mall. Peale mõlemat ülesannet küsiti kui kerge või raske see oli. Sellele sai vastata nelja punkti skaalal, kus üks tähistas väga kerget ning neli väga rasket, samuti oli võimalik vastajal kirjutada pikemalt mõlema ülesande kohta. Küsitluse lõpus uuriti, kui hästi või halvasti kasutajad redaktorist aru said, kas nende loodud mall vastas nende ootustele ning kas nad jagaksid enda tehtud malle teistega.

Joonisel 15 on kajastatud osade küsimuste vastused. Saadud tulemustest kajastus, et enamiku kasutajate jaoks ei olnud väikese ega suure malli loomine väga keeruline. Üle poolte kasutajate jaoks oli üleüldine redaktori kasutamine lihtne või väga lihtne. Küll aga oli väikese malli loomine kergem kui suure malli loomine. Põhjus võib olla see, et suure malli loomisel on lisaks pööramise ja peegeldamise võimalused, mida väikese malli vahelehel ei ole. Samuti on vaja kaameraga ringi liikuda, et tervet malli näha, mis teeb kogu protsessi tülikamaks.



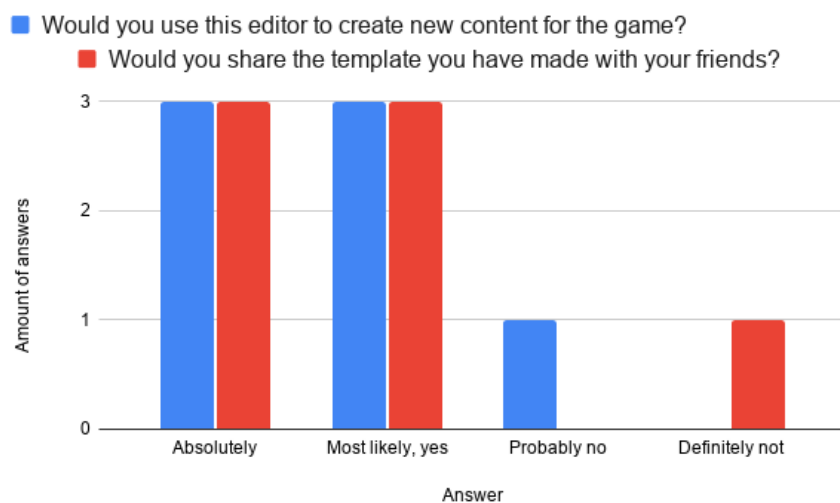
Joonis 15. Kasutajate tagasiside mallide loomise ja redaktori kasutamise kohta.

Joonisel 16 on toodud vastused küsimustele, kui hästi kasutaja redaktorist aru sai enne ja pärast testi läbimist. Number 1 tähendab, et kasutaja sai täielikult aru redaktorist ning number 4 tähendab, et kasutaja ei saanud üldse redaktorist aru. Tulemustelt on näha, et testi läbimine ei mõjutanud enamuse vastanute arusaama redaktorist. Ühe inimese puhul see paranes ning teise puhul halvenes ja kõigi ülejäänud vastanute puhul jäi see samaks. Seega testi läbimine ei aidanud kasutajatel redaktorist paremini aru saada.



Joonis 16. Kasutajate arusaamine redaktorist enne ja pärast testi sooritamist.

Joonisel 17 on toodud kahe küsimuse vastused. Esimeses küsimuses uuriti, kas kasutajad kasutaksid seda redaktorit, et mängule sisu juurde luua ning teises küsimuses uuriti, kas kasutajad jagaksid enda tehtud malle oma sõpradega. Mõlema küsimuse puhul oli enamuse kasutajate vastus jah või kindel jah.



Joonis 17. Kasutajate arvamus mallide loomise ja nende jagamise kohta.

Mitme küsimuse puhul oli vastajatel võimalik oma vastust täpsustada ning lõpus küsiti kasutajatelt, mida nende arvates oleks vaja redaktori juures veel muuta. Järgnevalt on need märkused ja ettepanekud välja toodud.

Suure malli vahelehe kohta arvati, et seal peaks olema võimalus sisse ja välja suumida ning liigutada kaamerat ka hiire abil. Samuti võiks olla võimalik juba paigutatud tükke keerata. Tükkide keeramisel toodi välja, et plokkide ei keerata, kuigi mõnel juhul oleks see kasulik, näiteks trepi plokkide puhul. Vastajaid häiris ka see, et pööramise ja peegeldamise nupud ei liigu kaameraga kaasa ehk kasutaja peab tihti edasi-tagasi liikuma, et tükke peegeldada või pöörata enne nende paigutamist. Samuti soovitati lisada kiirklahvid nende operatsioonide sooritamiseks. Kasutajaliidese kohta arvati, et peaks olema rohkem kirjeldusi nuppudel ning plokkide loetelu võiks olla laiem ja sisaldada plokkide nimesid.

## 6 Kokkuvõte

Käesolevas töös tutvustati arenduses olevat mängu Blastronaut. Seejärel kirjeldati erinevaid protseduurilise genereerimise meetodeid ja vaadeldi lähemalt, kuidas on maailm genereeritud Blastronaut mängus. Töö eesmärgiks oli täiendada seda protseduurilist generaatorit mallipõhise süsteemiga. Loodud süsteem võimaldab protseduuriliselt genereeritud maailma paigutada käsitsi loodud kohtasid, mis muudavad maailma avastamise mängija jaoks huvitavamaks. Selliste kohtade defineerimiseks saab kasutaja luua malle, mida on kahte tüüpi: väikesed ja suured mallid. Väikesed mallid koosnevad plokkidest ja suured mallid koosnevad omakorda väikestest mallidest. Need mallid paigutatakse deterministlikult maailma nii, et nad sulanduvad ülejäänud keskkonnaga.

Samuti loodi redaktor, mis võimaldab neid malle lihtsasti mängule lisada. Malli redaktoris saab kasutaja koostada nii väikeseid kui ka suuri malle. Sellest malli redaktorist tehti kaks versiooni. Esimese redaktori versiooni puhul ilmsid puudujäägid, mis parandati teises versioonis. Teise malli redaktori kasutajakogemuse hindamiseks viidi läbi testimine, milles osalejad pidid omaenda malli kokku panema. Testimise tulemusena selgus, et loodud malli redaktorit on enamiku vastajate jaoks pigem lihtne kui raske kasutada. Samuti arvasid peaaegu kõik vastanutest, et nad kasutaksid redaktorit mängule sisu loomiseks ja jagaksid ka enda loodud malle oma sõpradega.

Kasutajakogemuse testimise käigus saadi ka head tagasisidet redaktori puudujääkide kohta ning soovitusi selle paremaks muutmiseks. Näiteks, soovitati täiustada kaamera liikumist redaktoris. Samuti soovitati lisada kiirklahvid peegeldamise ja pööramise jaoks ning lubada tükke pöörata ka pärast nende paigutamist.

Tulevikus peaks mallidele lisama võimaluse mallidele defineerida mallide tükki ja plokkidele esinemistõenäosusi, mis võimaldaks samast mallist erinevaid variatsioone luua. Kuna mängule on plaanis lisada ka vastased ja masinad, siis neid peaks samuti saama mallidesse paigutada.

Kokkuvõttes on valminud süsteem praeguseks juba mänguga liidestatud ja on juba

praegu kasutusel. Blastronauti mäng vajab veel paar kuud arendamist ning pärast seda on plaan see Steam poes mängijatele kättesaadavaks teha.

## Kasutatud materjalid

- [1] A. Baldwin *et al.* “Mixed-initiative procedural generation of dungeons using game design patterns”. Teoses: *2017 IEEE Conference on Computational Intelligence and Games (CIG)*. ISSN: 2325-4289. August 2017, lk. 25–32.
- [2] J. Freiknecht ja W. Effelsberg. “Procedural Generation of Multistory Buildings With Interior”. *IEEE Transactions on Games* 12.3 (september 2020). Conference Name: IEEE Transactions on Games, lk. 323–336.
- [3] Alexander Gellel ja Penny Sweetser. “A Hybrid Approach to Procedural Generation of Roguelike Video Game Levels”. Teoses: *International Conference on the Foundations of Digital Games*. FDG '20. New York, NY, USA: Association for Computing Machinery, september 2020, lk. 1–10.
- [4] S. Snodgrass ja S. Ontañón. “Learning to Generate Video Game Maps Using Markov Models”. *IEEE Transactions on Computational Intelligence and AI in Games* 9.4 (detsember 2017). Conference Name: IEEE Transactions on Computational Intelligence and AI in Games, lk. 410–422.
- [5] Francesco Berto ja Jacopo Tagliabue. “Cellular Automata”. Teoses: *The Stanford Encyclopedia of Philosophy*. Fall 2017. Metaphysics Research Lab, Stanford University, 2017.
- [6] Petri Purho. *Exploring the Tech and Design of 'Noita'*. URL: <https://www.gdcvault.com/play/1025695/Exploring-the-Tech-and-Design> (vaadatud 03.05.2021).
- [7] Jonathon Doran ja Ian Parberry. “A prototype quest generator based on a structural analysis of quests from four MMORPGs”. Teoses: *Proceedings of the 2nd International Workshop on Procedural Content Generation in Games*. PCGames '11. New York, NY, USA: Association for Computing Machinery, juuni 2011, lk. 1–8.

- [8] Adam Summerville ja Michael Mateas. “Sampling Hyrule: Multi-Technique Probabilistic Level Generation for Action Role Playing Games”. *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment* 11.1 (september 2015). Number: 1.
- [9] Pablo García-Sánchez *et al.* “Optimizing Hearthstone agents using an evolutionary algorithm”. *Knowledge-Based Systems* 188 (jaanuar 2020).
- [10] Jaan Janno. “Java teek simplex müra paralleelseks arvutamiseks”. Thesis. Tartu: Tartu Ülikool, 2015.
- [11] *Noise!* URL: <https://uniblock.tumblr.com/post/97868843242/noise> (vaadatud 10.01.2021).
- [12] Franz Aurenhammer, Rolf Klein ja Der-tsai Lee. *Voronoi Diagrams And Delaunay Triangulations*. World Scientific Publishing Company, juuni 2013.
- [13] Patrick Cozzi ja Christophe Riccio. *OpenGL Insights*. CRC Press, juuli 2012.
- [14] *Voronoi Noise*. September 2018. URL: <https://www.ronja-tutorials.com/post/028-voronoi-noise/> (vaadatud 06.05.2021).
- [15] Michael Cripe. *For Derek Yu, Game Development Is a Life-Long Passion, and Spelunky 2 Is Just the Start*. Detsember 2020. URL: <https://www.escapistmagazine.com/v2/spelunky-2-interview-derek-yu-mossmouth/> (vaadatud 10.12.2020).
- [16] Darius Kazemi. *Spelunky Generator Lessons*. URL: <http://tinysubversions.com/spelunkyGen2/> (vaadatud 06.05.2021).
- [17] Noor Shaker *et al.* “Constructive generation methods for dungeons and levels”. Teoses: *Procedural Content Generation in Games*. Oktoober 2016, lk. 31–55.

## **Lisad**

### **I. GitLab repositoorium**

Mängu GitLab repositoorium asub lingil <https://gitlab.com/perfoon/blastronaut>. Ligipääsu saamiseks palun saata email aadressil [silverspitson42@gmail.com](mailto:silverspitson42@gmail.com). Testimise käigus kasutatud rakendus ning PDF vormingus küsitlus on pandud tööle lisadena kaasa.

## II. Küsitus

1. How well did you understand the editor?

Completely	Mostly	Barely	Not at all

2. How easy or hard was it to make the chunk?

Very easy	Easy	Hard	Very hard

3. Explain your previous answer. Which problems did you encounter during the process?

4. How easy or hard was it to make the master template?

Very easy	Easy	Hard	Very hard

5. Explain your previous answer. Which problems did you encounter during the process?

6. How easy or hard was it to use the editor overall?

Very easy	Easy	Hard	Very hard

7. Explain your previous answer. Which problems did you encounter during the process?
8. Did you achieve the desired result? Were you able to create the template you envisioned?

Exceeded my expectations	Looked exactly how I imagined it	Mostly resembled my idea, but not exactly	Barely resembled my idea	Did not look similar at all	Wasn't able to create a template

9. Would you use this editor to create new content for the game?

Absolutely	Most likely, yes	Probably no	Definitely not

10. Would you share the template you have made with your friends?

Absolutely	Most likely, yes	Probably no	Definitely not

11. After completing the tasks, how well did you understand the editor?

Completely	Mostly	Barely	Not at all

12. Do you have any other comments?

13. What would you improve about the editor or the template system?

### **III. Litsents**

#### **Lihtlitsents lõputöö reprodutseerimiseks ja üldsusele kättesaadavaks tegemiseks**

Mina, **Silver Spitsõn**,

(autori nimi)

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

**Mallide lisamine Blastronaut mängu protseduurilisele generaatorile,**

(lõputöö pealkiri)

mille juhendaja on Jaanus Jaggo,

(juhendaja nimi)

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

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

Silver Spitsõn

**07.05.2021**