

TARTU ÜLIKOOL
Arvutiteaduse instituut
Tarkvaratehnika õppekava

Karl Kuusik

**Ehitise vananemise simuleerimine Godot mängu-
mootoriga**

Magistritöö (30 EAP)

Juhendaja: Jaanus Jaggo, MSc

Tartu 2022

Ehitise vananemise simuleerimine Godot mängumootoriga

Lühikokkuvõte:

Käesolev magistritöö kirjeldab algoritmi, millega saab luua ajas vananevaid ehitisi, vastavalt kasutatud ehitusmaterjalides toimuvatele füüsikalistele protsessidele. Samuti antakse ülevaade eri materjalide vanandamisest arvutigraafikast ning selle jaoks kasutatud meetoditest. Lisaks kirjeldatakse Godot mängumootoris algoritmi implementatsiooni ja tuuakse välja selle käigus tekkinud probleemid ning nende probleemide lahendused. Saadud tulemust analüüsitakse ja diskuteeritakse võimalikke tuleviku edasiarendusi.

Võtmesõnad:

Arvutigraafika, protseduuriline genereerimine, puidu lagunemine, Godot

CERCS:

P170 Arvutiteadus, arvutusmeetodid, süsteemid, juhtimine (automaatjuhtimisteooria)

Simulating the aging of a building using the Godot game engine

Abstract:

This Master's thesis describes an algorithm for creating buildings which age according to the physical processes taking place inside the building materials. In addition, an overview of methods used in computer graphics to age different materials is provided. The algorithm is implemented with Godot game engine and associated problems and their solutions are described. The results are analyzed and possible future developments are discussed.

Keywords:

Computer graphics, procedural generation, wood decay, Godot

CERCS:

P170 Computer science, numerical analysis, systems, control

Sisukord

1	Sissejuhatus	5
2	Mõisted ja terminid	6
3	Erinevad vanandamise meetodid arvutigraafikas.....	7
3.1	Fallout 4.....	7
3.2	Rooste rakuautomaatidega.....	8
3.3	Sõlmepõhine geomeetria	9
4	Ehitise vananemise simuleerimine	10
4.1	Kasutatud ehitusmaterjalid	10
4.1.1	Puit	10
4.1.2	Õlekõrtest katused	10
4.1.3	Vatt ja tamp seinad.....	11
4.2	Simuleeritud protsessid	12
4.2.1	Niiskuse levimine puidus	13
4.2.2	Puidu paindumine ja murdumine	14
4.2.3	Puidu mädanemine	15
4.2.4	Õlekõrte lagunemine	16
4.2.5	Vatt ja tamp seinte lagunemine	16
5	Algoritmi implementeerimine Godot mängumootoris.....	18
5.1	Ehitiste konstrueerimine	18
5.1.1	Puitpalkide graafiline implementatsioon.....	18
5.1.2	Puitpalkide kokkupuudete süsteem	19
5.1.3	Puidu niiskuse ja mädaniku levimise implementeerimine	20
5.1.4	Puidu murdumise implementeerimine	21
5.1.5	Seinte graafiline implementatsioon.....	24
5.1.6	Õlekõrtest pundarde graafiline implementatsioon	24
5.2	Ajatelg	25

5.3	Godot füüsikamootoril tuginev algoritm	26
5.4	Area-objektidel tuginev algoritm	27
6	Tulemused	29
6.1	Inglismaa pronksiaegne ringmaja	29
7	Algoritmi hindamine	32
7.1	Visuaalne sarnasus reaalsusega	32
7.2	Jõudlus	33
7.3	Võrdlus varasema tööga	33
7.3.1	Blenderi generaator	33
7.3.2	Puu massi kadu	34
8	Tuleviku edasiarendused	36
8.1	Pikem ajaperiood	36
8.2	Paindumine ehitusmaterjalides	37
8.3	Rohkem ehitusmaterjale	37
8.4	Teised mängumootorid	37
9	Kokkuvõte	38
10	Viidatud kirjandus	39
	Lisad	41
I.	Rakenduse paigaldamine ja kasutamine	41
II.	Rakenduse lähtekood	42
III.	Litsents	43

1 Sissejuhatus

Inimesed on ehitanud ajaloo vältel mitmesuguseid erinevaid ehitisi, millest on tänapäevaks säilinud vaid üksikud. Aastatuhandete tagusest elust saame aga üha parema ülevaate tänu kiiresti arenevatele arheoloogilistele meetoditele, mis võimaldavad erinevaid asukohti välja kaevates avastada seniks saladuseks jäänud tõsiasju meie esivanemate igapäevaelu kohta. Näiteks oskavad tänapäeval arheoloogid ainult mullapinnaste erinevate värvide järgi tuvastada iidseid kraave, puidust palkidest jäänud auke ja lõkkeplatse. Kaevamistel leitud esemed ja säilmed aitavad luua täpsema pildi selle kohta, mis antud asukohas kunagi toimus ja mis ajaperioodil. Selle info inimestele edastamiseks kasutatakse muuseumides, telesarjades ja väljapanekutel erinevaid visualiseerimismeetodeid.

Üheks näiteks, kus taolisi visualiseeringuid kasutatakse, on Briti populaarteaduslik telesari "Time Team", mis on populariseerinud arheoloogiat ja sellest arusaamist. Iga sarja episood näitab kindlal asukohal 3-päevast väljakaevamiste protsessi ja kulmineerub kogutud informatsiooni põhjal visualiseeritud ehitise või asula näitamisega. Visualiseeringute tegemiseks kasutati saates esialgu joonistusi, aga hiljem kasutati selleks arvutigraafikat.

Arvutigraafikas visualiseeritakse vananemise protsessi peamiselt arvutimängudes. Hea näide ehitiste vananemisest on mängus „Valheim“, kus vihma mõjul saavad katuseeta objektid vigastada ning näevad selle tulemusena vanemad välja. Mänge, kus ehitised muutuvad üle aja arheoloogilisteks jäästeks, kahjuks ei leidu. Sellest on inspireeritud ka käesolev töö, kus on esindatud mõned puuduvad protsessid ehitise vananemisest.

Töö eesmärgiks on uurida ehitiste vananemise protsessi ja panna selle põhjal kokku mudel, mille abil ehitisi mängumootorites vanemaks muuta saaks. Mudel tuleb realiseerida Godot mängumootoris, kus seda testida saab. Vananemise protsessi lõppedes peab kogu protsess ajatelje abil edasi-tagasi keritav olema ning tulemus peab tulenema mehhanismidest päriselus.

Töö teine peatükk kirjeldab erinevaid meetodeid materjalidele ja ehitistele vananenud välimuse andmiseks. Töö kolmas peatükk kirjeldab algoritmi jaoks kasutatud ehitismaterjale ja nende omadusi. Samuti kirjeldatakse materjalides toimuvaid protsesse nende vananedes ning seda, kuidas need protsessid materjali omadusi mõjutavad. Töö neljas peatükk kirjeldab algoritmi implementatsiooni Godot mängumootoris. Töö viies peatükk kirjeldab algoritmi väljundit ning kuendas peatükis hinnatakse ja analüüsitakse saadud tulemust. Töö seitsmes peatükk kirjeldab võimalikke tuleviku edasiarendusi.

2 Mõisted ja terminid

Arheoloogiline jääne¹ - arheoloogilistel väljakaevamistel paljandunud paikne nähtus, ilming või objekt, mis ei ole hoone, rajatis ega üksikleid.

ringmaja - inglise k. *roundhouse*, levinud ehitis inglismaal pronksiajastul.

vatt ja tamp – inglise k. *wattle & daub*, iidne seinte ehitusmeetod.

difusiiivsus - tunnus, mis iseloomustab meediumi läbitavust mingi vedeliku poolt.

paindetugevus - maksimaalne jõud, mida materjal suudab enne murdumist taluda.

rakuautomaat – inglise k. *cellular automata*, diskreetne arvutusmudel automaaditeoorias

varjutaja – inglise k. *shader*, arvutiprogramm, mis määrab arvutigraafika kujutamiseviisi

mängumootor - rakendus mängude loomiseks ja jooksumiseks.

¹ <https://sonaveeb.ee/search/unif/dlall/arh/feature/4>

3 Erinevad vanandamise meetodid arvutigraafikas

Arvutite arvutusjõu suurenedes on arvutigraafikas kujutavate objektide ja keskkondade visuaalne kompetents tõusnud. Nende kvaliteeti võimendavad ka uued tehnoloogiaid, näiteks kiirte jälitamine (ingl. k. *raytracing*), mis võimaldab kujutada reaalses valguse peegeldumist eri pindadelt. Selliseid tehnoloogiaid kasutavad ka realistlikke keskkondi kujutavad arvutimängud, kus on eriti populaarseks muutunud nii iidset linnade kui ka post-apokalüptilised tühermaad.

3.1 Fallout 4

„Fallout 4“ on esimese isiku vaates rollimäng, mille tegevus toimub tuumasõjajärgses maailmas. Sellisele keskkonnale on iseloomulik, et enamus majadest ja objektidest on radiatsiooni ja inimtegevuse puudumise tõttu aastate jooksul lagunenu. Seetõttu on mängu keskkonna disainimisel jäljendatud 40-60ndate stiilis metallist ehitiste vananemist pärast tuumasõda.

Mängija alustab oma teekonda majas nimega „House of Tomorrow“, mida on joonis 1 peal kujutatud nii enne kui pärast tuumasõda [1]. See annab hea võrdluse aja jooksul toimunud muutustest. Need muutused on tehtud mängu kunstnike poolt nende enda äranägemise järgi ja ei pruugi vastata pärismaailma lagunemisprotsessidele.

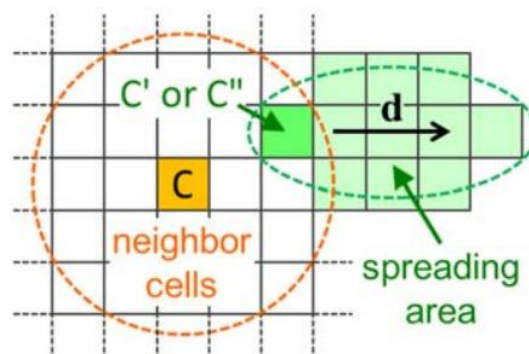


Joonis 1. „House of Freedom“ enne ja pärast tuumasõda [1].

3.2 Rooste rakuautomaatidega

Materjalide välimuse vanandamist on manuaalsete meetodite asemel ka algoritmiliselt tehtud. Eriti populaarsed selleks on metallid ja nende oksüdeerumine. Näiteks raua ja terase oksüdeerumise tulemuseks on rooste, mille kujunemine sõltub roostetava pinna geometriilistest omadustest. Eelkõige roostetavad nõgusad pinnad, kuhu niiskusel on gravitatsiooni mõjul lihtne koguneda.

Selle efekti loomiseks arvutigraafikas saab kasutada tõenäosuslike rakuautomaate. Rakuautomaatide implementeerimiseks jaotatakse terve objekti pind eri parameetritega ruutudeks ning iteratiivselt muudetakse nende ruutude väärtusi vastavalt rakuautomaadi reeglitele [2].



Joonis 2. Rooste levimine rakuautomaadis [2].

Kirjeldatud lahenduses eristatakse kahte erinevat tüüpi roostet, milleks on nähtav ja pinnaalune rooste. Algoritm kontrollib iga ruutu, vaadates kas tegemist on nähtava roostega või mitte. Kui rooste mõnes ruudus on nähtav, siis valitakse üks ruudu juhuslik naaberruut, kuhu pannakse levima pinnaalune rooste (vt Joonis 2). Pinnaalune rooste levib voolavalt mööda vektorit d (vt Joonis 2) ning muutub aja möödumisel nähtavaks [2]. Algoritmi mitmekordse jooksutamise tulemuseks on pind, millel olev rooste jätab kindlas suunas levimise mulje ja selle eri staadiume on kujutatud joonisel 3.



Joonis 3. Roostetav pool-silinder [2].

3.3 Sõlmepõhine geomeetria

3D tarkvaraga Blender² saab luua sõlmepõhist geomeetria, mis tähendab, et see pole mitte kunstniku poolt käsitsi modelleeritud, vaid algoritmiliselt erinevate reeglite põhjal genereeritud. Arvutigraafika kunstnik Sozap kasutas Blenderi sõlmepõhiseid funktsioone materjalide ja geomeetria genereerimiseks, et luua vananenud ehitiste generaator [3]. Tema lahendus suudab lisaks vananenud välimusele andmisele ka ehitise geomeetrilist kuju muuta, tekitades kokkuvarisenud sektsioone [3].

Siiski puudub loodud generaatoril täielik arusaamine kokkuvarisemise protsessist, sest kasutab funktsioone, mis üritavad seda vaid visuaalselt jäljendada. Seda on hästi näha olukordades, kus tulemuseks saadud ehitisel ilmnevad hõljuvad osad. Näiteid ühest generaatoriga tehtud ehitise eri staadiumitest näeb joonisel 4, kus on ühes ehitise staadiumis näha ka hõljuvaid puutükke.



Joonis 4. Blenderis vanandatud ehitis.

² <https://www.blender.org/>

4 Ehitise vananemise simuleerimine

Ehitiste vananemist mõjutavad mitmed tegurid. Simuleeritava protsessi keerukus sõltub suuresti sellest, kui palju tegureid kaasata. Samuti mõjutab keerukust see, kui süvitsi igasse protsessi minna. Peamiselt tuleb siiski ehitise vananemise juures arvestada kolme faktoriga:

- 1) ehitusmaterjalid,
- 2) materjalidega seotud protsessid ajas,
- 3) ning ehitise struktuurne tugevus.

4.1 Kasutatud ehitusmaterjalid

Töös kasutatud ehitusmaterjalide valikul arvestati nende ajaloolise taustaga. Valituks osutusid puit, õlekõrred ja savi.

4.1.1 Puit

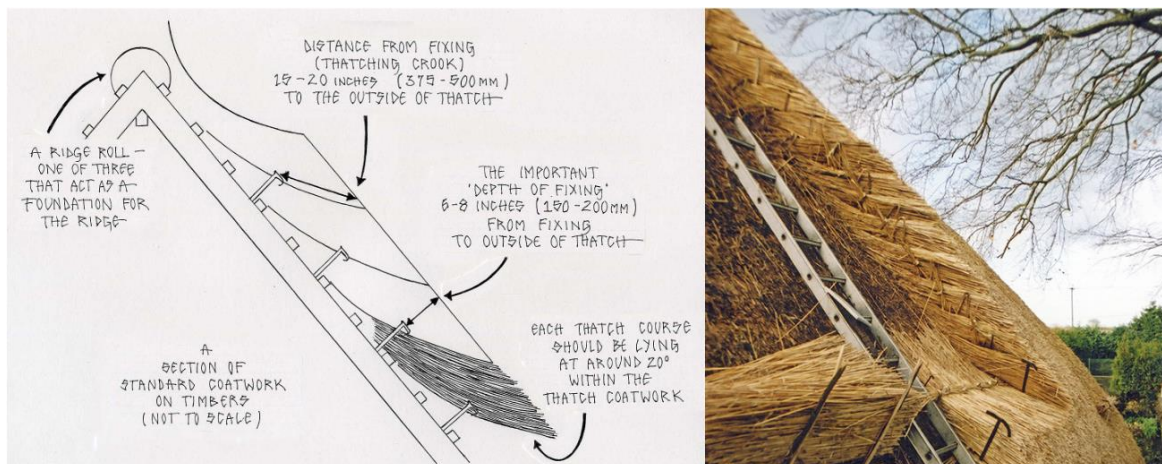
Puit on olnud läbi aastatuhandete inimeste poolt laialt kasutatud ehitusmaterjal ja on tänapäevalgi üks peamisi ehitusmaterjale. Pikast kasutamisajast tulenevalt on detailselt dokumenteeritud erinevate puiduliikide mehaanilisi omadusi nagu tihedus, painduvus, niiskuspaisumine ja tugevus. Peamiselt eristatakse ehituses leht- ja okaspuitu, ning seejuures on lehtpuu tihtipeale painduvam ja mitte nii tihe kui enamik okaspuid.

Samas ei saa ainult puidu tiheduse kaudu hinnata nii tema mehaanilisi omadusi kui ka nende muutumist ajas. Näiteks nii 1300 aastases tõmbiokkalises ebaküpressis (okaspuu liik) kui ka 650 aastases jaapani tselkva (lehtpuu liik) puidus oli vanuse tõttu niiskuspaisumise konfitsent kahanenud [4]. Samas 300-aastase kuusest palgi niiskuspaisumise konfitsent oli vastupidiselt suurenenud [4]. Seetõttu lähtutakse ehituses alati kindla puuliigi omadustest. Käesolevas töös on selleks valitud harilik tamm, sest see on euroopas laialt levinud puuliik ja sellest tulenevalt kasutatud erinevates ehitistes läbi Euroopa ajaloo.

4.1.2 Õlekõrtest katused

Õlekõrsi on kasutatud ehitusmaterjalina alates põllumajanduse laialdasest levimisest. Reeglina on õlekõrred ehitusmaterjaliks katuste konstruktsioonil, aga need sobivad ka seinte täitematerjaliks. Katuse ehitamise jaoks seotakse õlekõrred pundardesse, mis seejärel asetatakse puuokstest tehtud võrestikule. Ajalooliselt on kinnitatud õlekõrtest puntraid võrestiku külge kas nõõri või taimekiududega, aga kaasajal kasutatakse selleks peamiselt metalltraati.

Pärast kinnitamist lõigatakse puntrad katusega tasaseks, et anda sellele ühtlane välimus. Näidet kaasaja rookatuse konstruktsioonist on kujutatud joonisel 5.



Joonis 5. Läbilõige rookatusest [5].

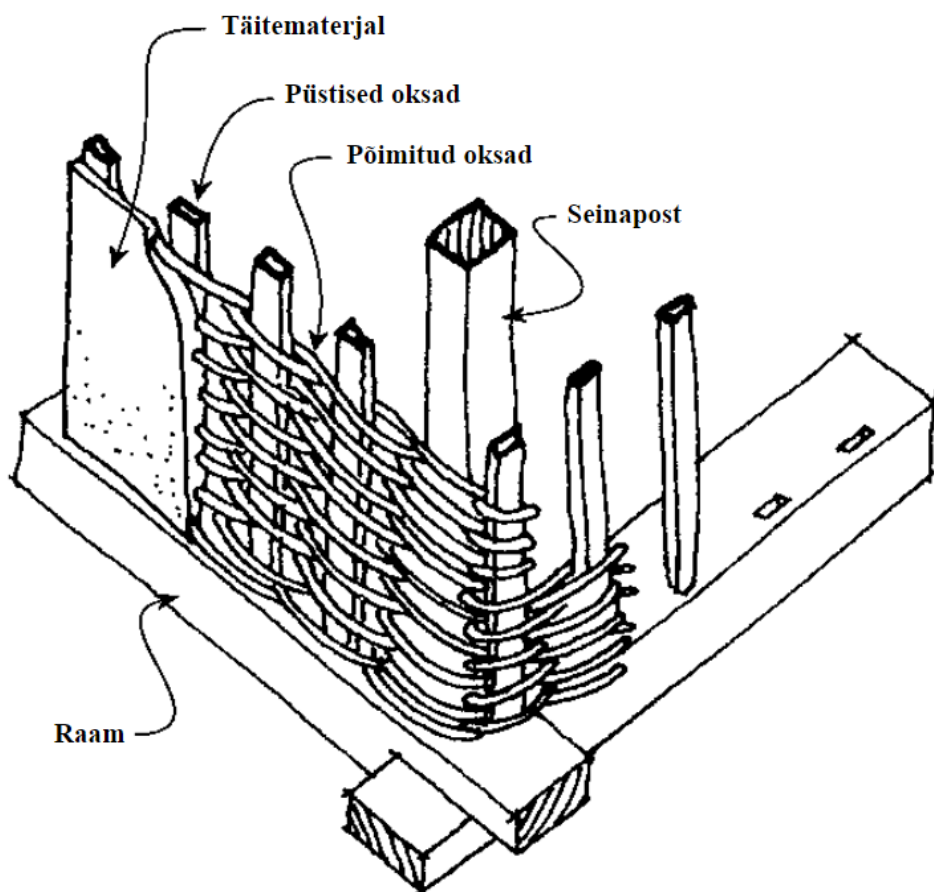
Õlekõrtest katuste funktsionaalne vastupidamine sõltub erinevatest faktoritest. Kaasaja õlekõrtest katused kestavad 40-50 aastat, sõltuvalt katuse kallakust [6]. Enamasti on rookatuste kallakuks 45 kraadi, kuna laugemad katused võivad sademeid läbi lasta [7]. Järsema kallakuga katused võivad kesta isegi pikemalt, kuna sademete korral voolab vesi kiiremini ära. Suurema kallaku miinuseks on hind, sest iga 10 kraadi kohta tekib umbes 20% katusepinda juurde [5]. Pikast eluajast hoolimata tuleb rookatuseid perioodiliselt hooldada, asendades peamiselt katuse harjas olevat roogu, kuna see kulub kõige kiiremini. Hoolduseid tehakse tavaliselt sõltuvalt katusest iga 7-12 aasta tagant [8].

Iidsete ehitiste rookatuste kohta arheoloogilist informatsiooni ei ole. Õlekõrred on kergesti biolagunevad, mistõttu ei jää neist aja möödudes mitte midagi alles [9].

4.1.3 Vatt ja tamp seinad

Joonisel 6 kujutatud kiviajast pärinev ehitusmeetod seinte jaoks kannab nimetust vatt ja tamp (ingl. k. *wattle & daub*). Selleks, et sellise meetodiga seinu ehitada, püstitatakse kõigepealt puupalgid, mille vahele luuakse väiksematest painduvatest okstest sõrestik. Järgnevalt segatakse valmis täiteaine, mis tambitakse eelnevalt püstitatud sõrestiku sisse. Selle täiteaine täpne koostis varieerub suuresti üle ajaloo ning sõltub tihti ehitise geograafilisest asukohast. Tihti on peamiseks aineks savi või muu savilaadne mass, millele lisatakse sideaineks juurde lubjakivist pulbrit. Lubjakivi lisamise eesmärk on loodud segu kivistumise soodustamine ning seda kasutatakse ka tänapäeval betooni valmistamisel.

Seinasisest okstest sõrestikku toetavate tugipalkide püstitamiseks on mitu võimalust. Varajases ehitusmeetodi kasutuses kinnitati palgid otse maapinnase sisse. Selle jaoks süvistati püstitatava palgi läbimõõdule vastav auk. Hilisemalt võisid seinte tugipalgid olla kinnitatud puidust alusraami sisse (vt Joonis 6) ja osadel juhtudel võis vastav puitraam omakorda olla toetatud kividest aluse peale. Selle eesmärgiks oli ehitise struktuurset tugevust ja seeläbi ka ehitise eluiga tõsta. Samuti kergitas kivivundament ehitise puitseksiooni kõrgemale maapinnast, vähendades niiskuse mõju ehitisele.



Joonis 6. Vatt ja tamp sein puitraamil [10].

4.2 Simuleeritud protsessid

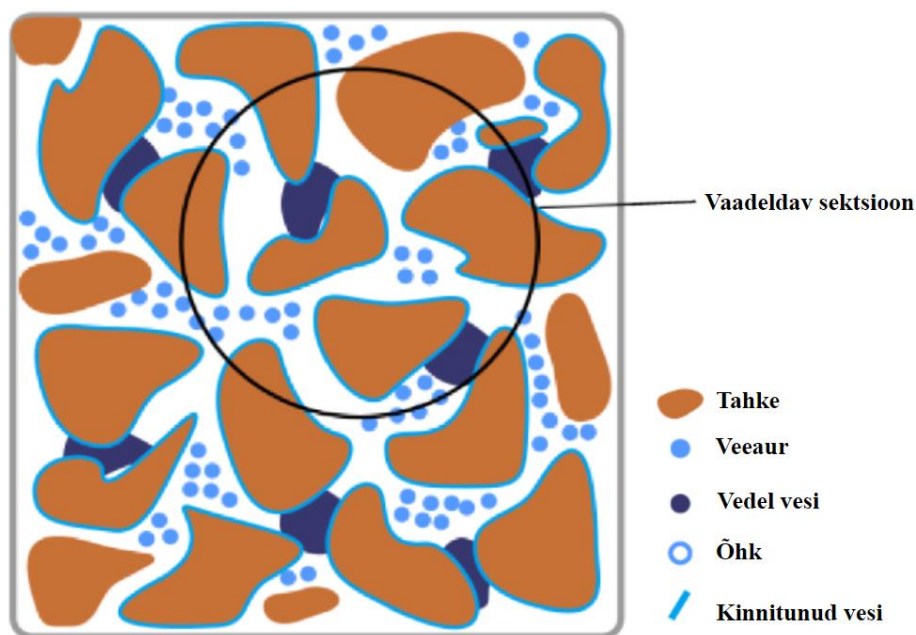
Ehitise vananemise simuleerimiseks tuleb arvestada sellega, et iga ehitises kasutatud ehitusmaterjal läbib ehitise eluea jooksul palju erinevaid protsesse. Puitraamiga ehitise puhul on peamisteks toimuvateks protsessideks puiduga seonduvad protsessid, mis mõjutavad nii ehitise välimust kui ka struktuurset tugevust. Nendest peamised on järgnevad:

- niiskuse liikumine mööda palki,
- palgi paindumine/deformeerumine

- puitstruktuuri kokkuvarisemine eelnevalt nimetatud tegurite poolt põhjustatud struktuursete nõrkuste mõjul.

4.2.1 Niiskuse levimine puidus

Puit on pooriline ja hügrokoopne materjal, ehk vee molekulid saavad liikuda vabalt puidu rakkude ja otseselt ümbritseva keskkonna vahel. See tähendab, et puidu niiskustase on otseses seoses teda ümbritseva keskkonnaga ning vastavalt ümbruskonna niiskusele saab puidu niiskustase kas tõusta või langeda. Samuti on kõik puidu füüsilised omadused tema niiskustasemest mõjutatud.



Joonis 7. Niiskus puidus [11].

Puitpalgi niiskustase ei pruugi olla kogu pikkuse ulatuses ühtlaselt jaotunud, kuna see on mõjutatud puupalki koheselt ümbritseva meediumi niiskusest [11]. Näiteks märja maapinna vastu toetav puupalgi ots on oluliselt kõrgema niiskustasemega kui vabas õhus olev ots, kuna teda ümbritseb oluliselt väiksema niiskustasemega keskkond.

Samuti võib niiskus mööda palki liikuda. Selle juures on oluline ümbritseva keskkonna õhurõhk ning puitmaterjali difusiivsus. Difusiivsus ei ole puidusiseselt konstantne ning saab ainult kindla vaadeldava sektsiooni järgi täpselt mõõta. Näidet vaadeldavat sektsioonist puidu läbilõikes ja selles olevatest vee olekutest on kujutatud joonisel 7. Kui teada difusiivsete omaduste muutusi kindlas puidutükis ning ümbritsevat õhurõhku ja õhu

niiskust, siis jälgides Fick'i ja Darcy seaduseid saab ennustada puidusisest vee voolu [11]. Nende täpne implementeerimine jääb liigse keerukuse tõttu käesoleva töö skoobist välja.

Selle asemel lihtsustati eelnevalt kirjeldatud niiskuse levimise mudelit, sest mudel on mõeldud lühiajaliste muudatuste ennustamiseks. Lihtsustades lähtuti põhimõttest, et hügrokoopse materjalis üritab niiskus alati liikuda kuivema materjali suunas ning puidutüki tihedus kahandab ühelt lõigult teisele kantavat niiskuse hulka. Lisaks kiirendab seda protsessi suurem ümbritseva niiskuse tase ning difusiivsed omadused on üldjuhul vähem tihedamatel puuliikidel paremad. Lihtsustatud keskmine niiskuse kasv aastas on kirjeldatud võrrandis 1, kus u_0 on vaadeldava palgi lõigu algne niiskus (%), U ümbritsevate pindade kõrgeim niiskustase (%), p puiduliigi tihedus (t/m^3) ning l lõigu pikkus(m).

$$u = u_0 + 0.5 * U \frac{l}{p}$$

Võrrand 1. Niiskuse levimine palgilõigis ühes aastas.

4.2.2 Puidu paindumine ja murdumine

Ehitistes kasutatud puit peab taluma oma eluea jooksul palju erisuunalisi jõude. Sein ja katust toetavad palgid peavad suutma terve ehitise raskust taluda. Siiski võib tekkida olukord, kus mõnele palgile mõjuv jõud on liiga suur, näiteks juhul kui ligidal olev puu langeb ehitise peale. Sõltuvalt langenud puu massist võivad ehitist toetavad palgid painduda või murduda. Puidu puhul on palju muutujaid, mis mõjutavad selle paindumist ja murdumist, kuid kaks peamist on normaalelastsusmoodul (MOE) ja paindetugevus (MOR).

Normaalelastsusmoodul iseloomustab materjali jäikust, ehk keha võimet avaldada välisjõu deformeerimisele vastupanu ning seda iseloomustatakse Paskalites (Pa). Puitmaterjalide puhul kasutatakse seda terminit peamiselt juhul, kui paindumine toimub puidusüüga risti [12]. Normaalelastsusmoodul on võrdne materjali normaalpinge ja normaalmoonde jagatisega, nagu kirjeldatud võrrandis 2. Kui mõjuv välisjõud ületab normaalelastsusmoodulit, siis ei naase puutükk enam täielikult oma algele kujule ning põhjustatud deformatsioon on püsiv.

$$E = \frac{\text{normaalpinge}}{\text{normaalmoone}} = \frac{\sigma}{\varepsilon} = \frac{F/S}{\Delta l/l_0} = \frac{Fl_0}{S\Delta l}$$

Võrrand 2. Normaalelastsusmooduli arvutamine [13].

Paindetugevus iseloomustab maksimaalset jõudu, mis saab kehale mõjuda enne murde teket. Puidu paindetugevus arvutatakse võrrand 3 abil kehale vahetult enne murdumist mõjuva jõu F ning keha painduva osa läbilõike laiuse x , kõrguse y ja sügavuse z põhjal. Painduva osa pikkuseks loetakse kahe toetuspunkti vahelist pinda [14]. Murdumine võib olla osaline, mille juures jääb osa puutükist ühendatuks, aga vananemise protsessi implementatsiooni juures loetakse üleliigse keerukuse vältimiseks kõiki murdumisi täielikuks.

$$\delta_r = \frac{3Fx}{yz^2}$$

Võrrand 3. Paindetugevuse arvutamine.

4.2.3 Puidu mädanemine

Puidu mädanemine on põhjustatud puidu tselluliidist toituvatest seeneliikidest. Need seeneliigid vajavad puu sees kasvamiseks kindlaid tingimusi ning vajalikud tingimused erinevad liigiti, kusjuures peamiselt eristatakse pruun-, valge- ja pehmemädanikku [15]. Erinevusi esineb ka arvamustes, missuguse minimaalse puidu niiskustaseme juures seeneliigid levima hakkavad, kuid leitud tulemused varieeruvad 85% ja 98% vahel [15]. Üldjuhul nõustutakse siiski, et alla 20% suhtelisest maksimaalsest niiskustasemest ei teki ega progresseeru puidus mädanik [12]. Samuti ei progresseeru mädanemine miinuskraadidega, kuna seeneliigid ei suuda sellistes tingimustes paljuneda.

Siiski on proovitud luua mudeleid mädanikust põhjustatud lagunemise ennustamiseks. Üks selline mudel on kujutatud võrrandil 4, mille abil saab ennustada n päeva jooksul toimunud massi kadu D eelnevalt defineeritud D_T ja D_u funktsioonide abil, kus T_i on päeva keskmine temperatuur ja u_i on päeva keskmine niiskus [16]. Funktsioon D_T kirjeldab temperatuuri mõju mädanemisele ja D_u kirjeldab niiskuse mõju mädanemisele. Mudelit saab kasutada eeldusel, et mädanemine on juba alanud [16].

$$D(n) = \sum_1^n D_i = \sum_1^n (f(D_T(T_i), D_u(u_i)))$$

Võrrand 4. Niiskusest põhjustatud lagunemise arvutamine n -päeva kohta [16].

Kuna võrrand 4 on loodud päeva kohta ennustamiseks, siis tuli implementatsiooni jaoks laiendada seda võrrandit aasta peale. Selle jaoks tehti eeldusi, et veerand aastast

miinuskraadide tõttu mädanemist ei toimu ja varju all olemist ei arvestata liigse keerukuse vältimiseks. Järelikult on mädanemisest põhjustatud massikadu aastas 0.75 korda konstant x , mis on kao protsent aasta kohta ning vastavat valemit on kujutatud ka võrrandil 5. Massi kao protsendiks aastas valiti implementatsiooni juures 2%.

$$D_a(n) = \sum_1^n 0.75x$$

Võrrand 5. Niiskusest põhjustatud lagunemise arvutamine iga n aasta kohta.

Puidu mädanemine mõjutab oluliselt selle struktuurset tugevust. See on tingitud sellest, et mädanemisel toitub puuseen ligniinist rakuseintes. Tugevuse kaotust saab mõõta vastavalt sellele, kui palju on mass protsentuaalselt vähenenud [12]. Näiteks 5% kadumisel puidutüki kogumassist on struktuurne tugevus vähenenud 20% võrra, aga 10% kao puhul võib struktuurne tugevus langeda koguni 80% võrra [12].

4.2.4 Õlekõrte lagunemine

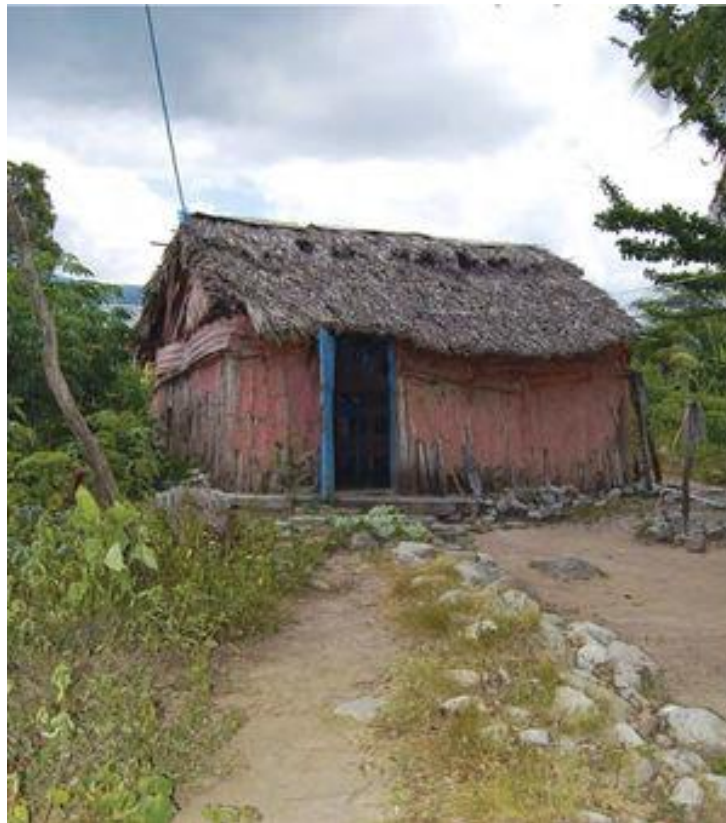
Õlekõrte mädanemine on sarnane puidule, kuna see saab nakatuda sama liiki mädanikega. Mädanikud puidus lagundavad rakuseintes olevat ligniini, mis on olemas ka õlekõrte rakuseintes, ning seetõttu suudavad nad ka õlekõrte siseses keskkonnas levida. Tavaliselt on nakkavaks liigiks valgemädanikud, aga pehmemädanik võib samuti tekkida [17]. Üldjuhul tekivad mädaniku kontsentratsioonid õlekõrskatuse siseruumipoolses osas seente kasvuks paremate tingimuste tõttu. Lisaks mädanikule mõjutab katuse lagunemist ka kliimast tingitud efektid. Päikese, tuule ja vihma mõjul muutuvad õlekõrre otsad hapraks ning õlekõrre värvus halliks [18].

4.2.5 Vatt ja tamp seinte lagunemine

Vatt ja tamp seinte lagunemise juures mängivad rolli peamiselt niiskus ja erosioon. Temperatuuri kõikumisega alla 0 kraadi paisub tambitud materjali poorides olev vesi, mis põhjustab pragusid seina pinnases [19]. Pragude täitmata jätmisel võib vesi sinna lõksu jääda, kiirendades sisemise puidust võrestiku lagunemise protsessi [19]. Praod võivad viia ka seina osalise delamineerumiseni, oluliselt võimendades niiskuse mõju seina sisemisele võrestikule [19]. Otsene kontakt märja maapinnaga mõjub samuti negatiivselt [19].

Kahjuks ei ole täpseid mudeleid ja numbreid selle kohta, kui kiiresti sein oma massi kaotab ja ära laguneb. Sellist mudelit ongi raske luua, sest lisaks niiskusele ja erosioonile on veel faktoreid, nagu taimeliikide kasv seintel ja näriliste poolt uuristatud augud [19].

Siiski saab kitsendusi tehes seina lagunemisprotsessi ette kujutada. Kui eeldada et ehitis on maha jäetud, siis ajas tekkinud pragusid seintes ei parandata ning lagunemise protsess on selle võrra kiirem. Lisaks laguneb maaga ühendatud osa seinast oluliselt kiiremini vihma-järgse otsese vee kontakti ja kõrgendatud niiskuse tõttu [19]. Näidet vastavate erosiooni tunnusmärkidega ehitisest on kujutatud joonisel 8.



Joonis 8. Vatt ja tamp seintega ehitis Dominikaani Vabariigis.

5 Algoritmi implementeerimine Godot mängumootoris

Algoritmi implementeerimiseks kasutati Godot³ mängumootorit. Godot kasuks otsustati mängumootori avaliku lähtekoodi ja mugavalt kasutatava dokumentatsiooni tõttu. Godot mootoris on peamiseks programmeerimiskeeleks GDScript⁴, mis on kergesti õpitav, kuna sarnaneb suuresti Python⁵-i programmeerimiskeelele.

5.1 Ehitiste konstrueerimine

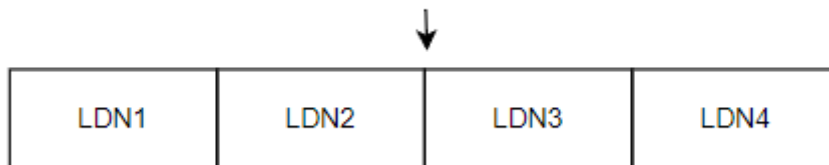
Algoritmi testimiseks on vaja implementeerida ehitiste monteerimiseks valitud ehitusmaterjalid. Nendeks on puidust palgid, vatt ja tamp seinad ning õlekõrtest puntrad katuse jaoks.

5.1.1 Puitpalkide graafiline implementatsioon

Puidust palkide Godot mängumootoris implementeerimiseks loodi puitpalgi protseduurilise genereerimise algoritm, mida on kujutatud joonisel 9. Algoritm koosnes kahest peamisest sammust. Kõigepealt luuakse puidutüki pikkuse ja raadiuse parameetrite põhjal massiiv LogDataNode (edaspidi LDN) klassi isenditest. Iga LDN isend kujutab üht lõiku puutükist ja sisaldab järgnevaid andmeid:

- asukoha koordinaadid,
- mõjuva stressi vektor,
- niiskustase,
- algne kuju ja reaalne kuju,
- kokkupuutepindade massiiv,
- ning sõnastik hetkel kokkupuutuvate objektide pindadest.

```
var nodeList = [ LDN1, LDN2, LDN3, LDN4 ]
```



Joonis 9. LDN objektide massiiv.

³ <https://godotengine.org/>

⁴ https://docs.godotengine.org/en/3.5/tutorials/scripting/gdscript/gdscript_basics.html

⁵ <https://www.python.org/>

Järgnevalt pannakse eelnevalt loodud LDN-objektide massiivi põhjal kokku visuaalselt nähtav puutükk. Selleks itereeritakse läbi objektide massiivi ja tehakse iga objekti kohta järgmised sammud:

- 1) Luuakse *MeshInstance* isend *logpiece*.
- 2) Väärtustatakse isendi *mesh* atribuut.
- 3) Pannakse isendi *translation* atribuudi väärtuseks LDN-i asukoht.

Esimese ja teise sammuga luuakse visuaalne puulõik ning kolmandas liigutatakse see vastavasse asukohta palgis. Kõigepealt antakse ette puulõigu jaoks kasutatav kujund, näiteks silinder, mis salvestatakse LDN-i algse kuju väärtuseks. Algsest kujust tehakse koopia, millele külge kinnitatakse puidu tekstuuriga varjutaja. Saadud koopiaga väärtustatakse teises sammus mainitud *mesh*-atribuut, mille tulemusel muutub puulõik nähtavaks. Koopia tegemise põhjuseks on see, et saaks muuta puulõigu varjutaja parameetreid eraldi teistest puulõikudest. Ilma koopiat tegemata muudetakse parameetreid kõikide puutükkide lõikude puhul, kuna iga puulõik viitaks samale objektile.

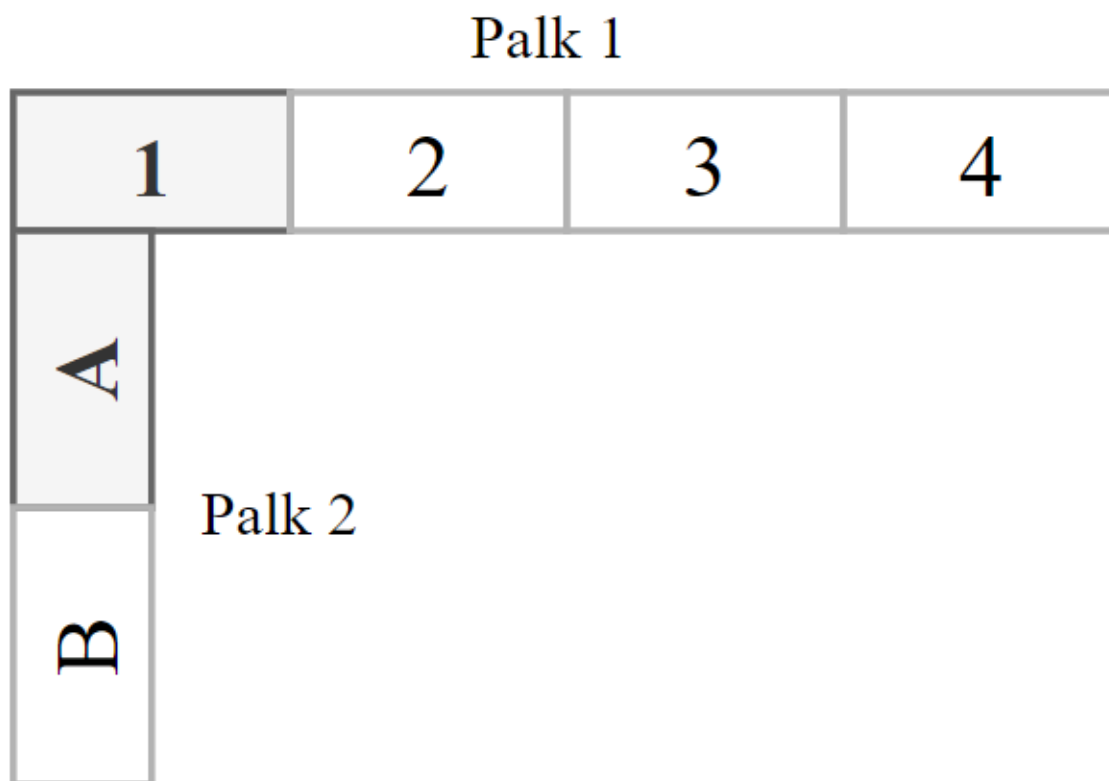
Põhjus LDN-klassi isenditest massiivi loomiseks seisneb selles, et puutükile mõjuvad jõud ja niiskus ei ole ühtlaselt jaotunud üle terve palgi pikkuse. Jaotades puupalgi eraldi lõikudeks saab iga lõigu keskmist niiskustaset ja mõjuva jõu vektorit lihtsasti pärida. Samuti saab simulatsiooni täpsust parandada, suurendades palgi lõikude arvu, kuid see nõuab ka rohkem arvutusjõudlust. Detailsematel ehitistel on simulatsiooni jõudlus oluline, kuna simuleeritavate objektide arv on suurem. Seetõttu on töös iga palgi meetri kohta kaks lõiku.

5.1.2 Puitpalkide kokkupuudete süsteem

Niiskuse ja murdumise implementeerimiseks peab puitpalk teadma, mis objektidega ta kokku põrkub ja mis on nende niiskustase. Selle jaoks täiendati eelnevalt kirjeldatud algoritmi selliselt, et lisati igale puupalgi lõigule kokkupuutekeha objekt (*CollisionShape*).

Kokkupuuteid registreerib puupalgi objekt erinevate signaalide abil, öeldes millise kokkupuutekehaga parasjagu põrkuti. Joonisel 10 kujutatud olukorras puutuvad kokku palk 1 ja palk 2, täpsemalt nende lõigud 1 ja A. Sellises olukorras on "body_shape_entered" signaali poolt tagastatav infoks viited palkide objektidele ning vastavad indeksid lõikude 1 ja A kokkupuutekehadele. Info puudutatud keha kohta salvestatakse signaali välja kutsunud palgi lõigu LDN-objekti sisse.

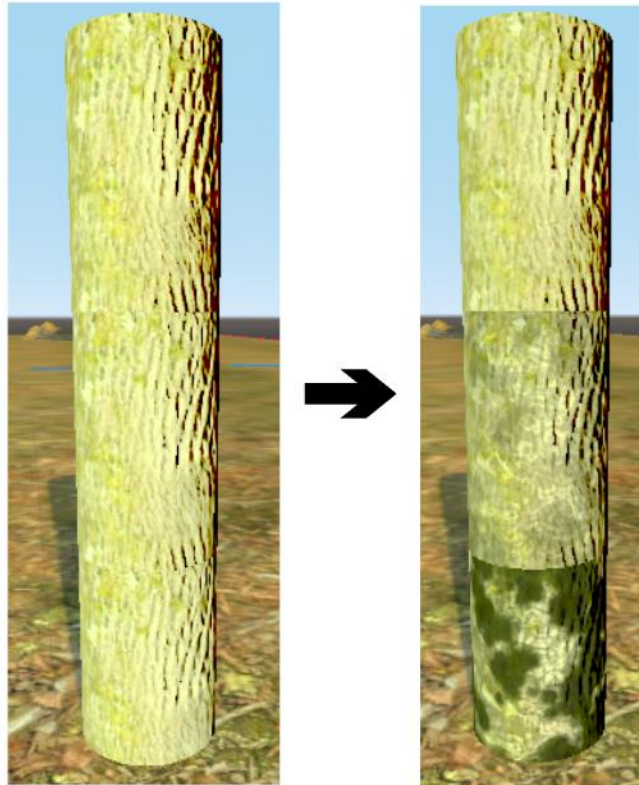
Lisaks puutumise signaalile on kasutusel ka vastupidine signaal põrkumise lõppemise tähistamiseks: “body_shape_exited”. Selle signaali väljakutsel tagastatakse sama info, mis kokkupuute toimumiselgi. Selle signaali abil kustutatakse lõigu LDN-objektis vastav kokkupuute info, sest puutumist enam ei toimu.



Joonis 10. Kokkupuutes puhul tagastatav info.

5.1.3 Puidu niiskuse ja mädaniku levimise implementeerimine

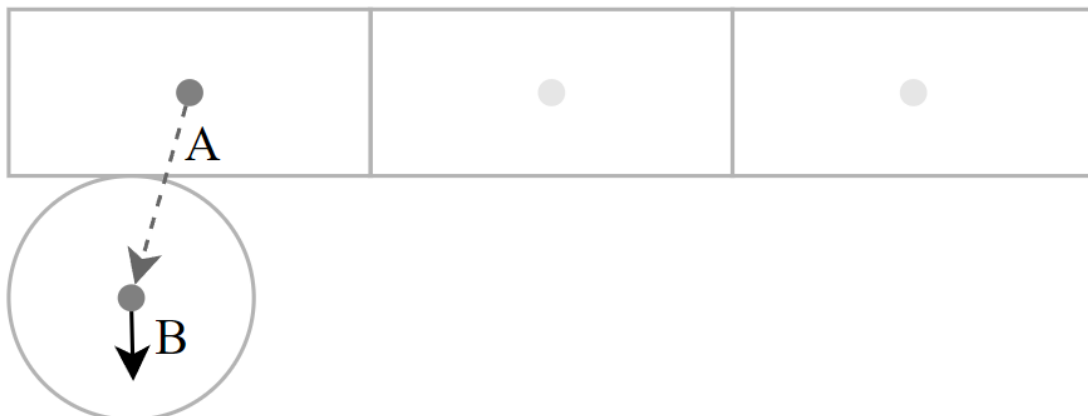
Niiskuse implementeerimiseks üle terve puupalgi kontrollitakse iga tema lõiku (LDN) ja vaadatakse lõiguga põrkuvaid pindu. Iga pinna kohta päritakse selle niiskustaset. Näiteks juhul kui puidulõiku puudutavaks pinnaks on muld, siis tagastatakse selle niiskuseks 100%. Juhul kui puutuvaks pinnaks on teine puulõik, siis vastava puulõigu niiskus. Leitud puutuvate pindade seast võetakse kõige kõrgem niiskustaseme väärtus, ning koos puulõigu hetkese niiskuse, pikkuse ja tiheduse järgi arvutatakse võrrand 1 abil puulõigu niiskustase. Seda protseduuri korratakse iga palgilõigu kohta. Joonisel 11 on kujutatud Godot mootoris implementeeritud niiskuse levimine.



Joonis 11. Niiskuse levimine palgis.

5.1.4 Puidu murdumise implementeerimine

Puitpalgi murdumise implementeerimiseks peab palk suutma arvestada mitmete jõududega. Palgi orientatsioon võib olla mitmesugune, see võib olla maa suhtes vertikaalne, toetada teiste palkide peale või lebada risti maapinnal. Sellest orientatsioonist sõltub suuresti, kui suurt jõudu suudab palk enne murdumist taluda. Samuti sõltub sellest ka jõud, mida palk teistele objektidele omakorda avaldab.



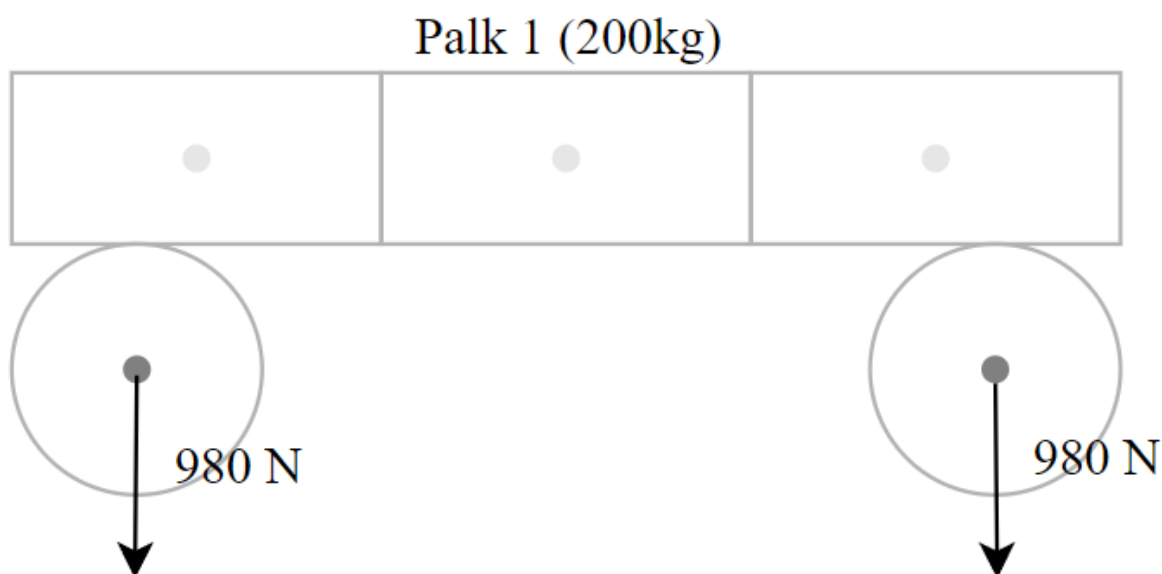
Joonis 12. Kokkupuutel leitavad vektorid A ja B.

Toetumise ja jõu avaldamise probleemi lahendamiseks laiendati eelnevalt mainitud kokkupuudete tuvastamise süsteemi. Joonisel 12 on näidatud kahte palki üksteise peal. Sellisel kokkupuutel leitakse kõigepealt kolmedimensionaalne vektor A, mis kirjeldab alumise ja ülemise palgi kokkupuutuvate lõikude asukohtade vahelist suhet. Vektorist A luuakse vektor B, mille jaoks jäetakse vektor A-st alles ainult y-komponent ning saadud vektor normaliseeritakse, mida on kujutatud joonisel 13 toodud näites. Vektor B põhjal saab järeldada, kuidas kokkupuutuv objekt puitpalki mõjutab. Kui vektori y-komponent on positiivne, siis puitpalk toetub kokkupuutuvale objektile, aga kui y-komponent on negatiivne, siis kokkupuutuv objekt toetub palgile. Saadud vektor B ehk toevektor salvestatakse lõigu LDN-objekti.

$$(-0.2523, -1.7624, 0.0) \Rightarrow (0.0, -1.7624, 0.0) \Rightarrow (0.0, -1.0, 0.0)$$

Joonis 13. Jõu suuna vektori arvutamine.

Lisaks jõu suunda kujutavale vektorile on vaja arvutada ka jõu enda suurus. Selle jaoks tehakse simulatsiooni ajal toevektorite põhjal struktuurarvutusi. Iga objekti poolt avaldatud jõud on seoses tema ja tema peale tuginevate objektide kaaluga. Näiteks, kahe tugipunkti peale toetatud palk mõjutab täpselt poole oma raskusega mõlemat tugipunkti, kuna selle mass on ühtlaselt jaotatud üle tema pikkuse. Joonisel 14 kujutatud 200kg kaaluva palgi puhul mõjub mõlemale tugipunktile pool tema kaalust, ehk 100kg. Mõjuva jõu leidmiseks tuleb kaal korrutada gravitatsiooni konstandiga ning tulemuseks on 980 njuutonit jõudu.



Joonis 14. Tugipunktile palk 1 poolt mõjuv jõud.

Saadud jõu, kasutatud puiduliigi paindetugevuse ja palgi dimensioonidega on võimalik tuuletada normaaltingimustes palgilõigule mõjuv stress. Stressi väärtuseks on protsentuaalne väärtus sellest, kui suur osa maksimaalsest talutavast jõust parasjagu avaldub. Maksimaalse jõu F saamiseks kasutati võrrandit 6, mis on avaldatud võrrandist 3.

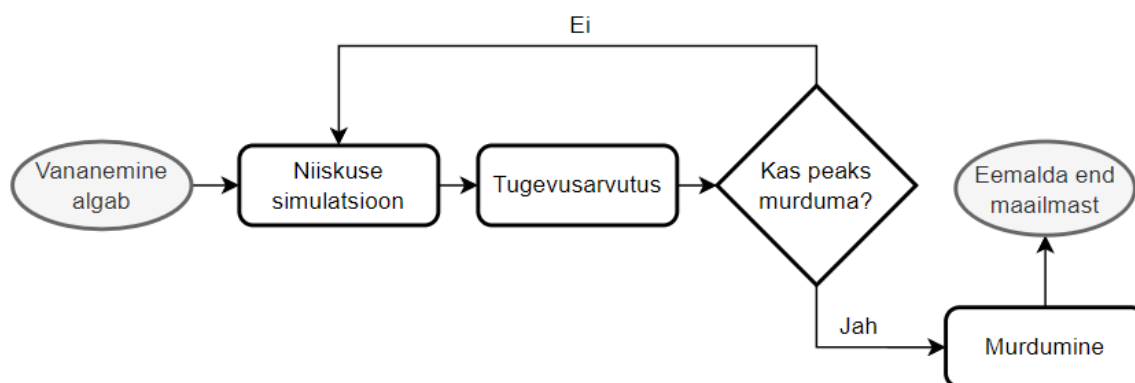
$$F_{\max} = \frac{\delta_r y z^2}{3x}$$

Võrrand 6. Maksimaalse palgile avaldatava jõu arvutamine.

Palgi lõigule mõjuv stress on mõjutatud ka mädanemisest. Kuna niiskuse simulatsioon tehakse enne tugevusarvutusi, saab vähendada maksimaalset avaldatavat jõudu F_{\max} vastavalt lõigu mädanemise väärtusele. Teades, et palgi maksimaalne talutav jõud on otseses seoses mädaniku poolt põhjustatavas massi kadumises, saame määrata palgilõigu niiskuse väärtuse järgi reeglid talutava jõu vähendamiseks:

- < 90% - mõju struktuursele tugevusele puudub.
- 90% < x < 95% - talutav jõud F väheneb proportsionaalselt 0% - 20%
- 95% - 100% - talutav jõud F väheneb proportsionaalselt 20% - 80%.

Kui palgile avalduv jõud on suurem kui maksimaalne talutav jõud, siis palk murdub. Niiskuse simulatsiooni, tugevusarvutuse ja murdumise protsessi omavaheline järgnevus puupalgis on kujutatud ka joonisel 15.

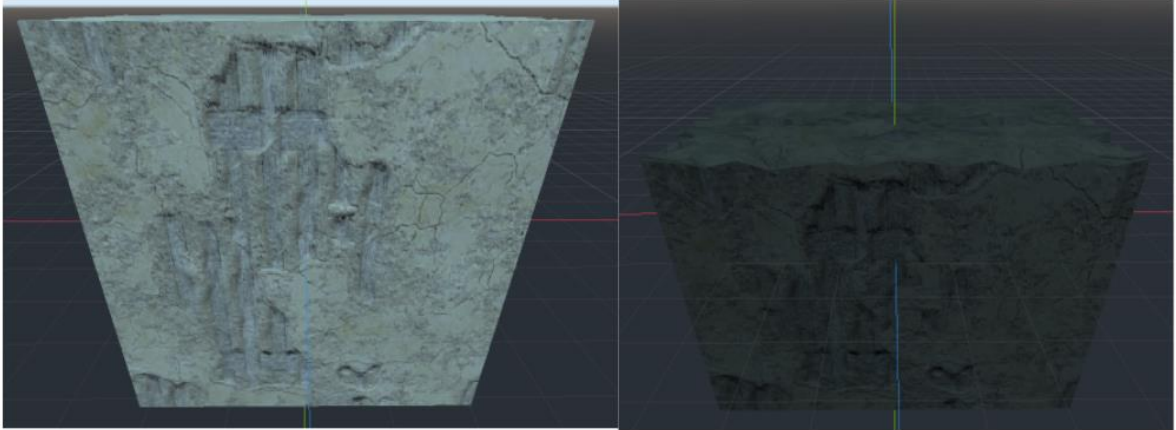


Joonis 15. Puidutüki isendi elutsükkel

Murdumisel luuakse kaks uut palki, mille pikkused vastavad murdunud palgi poolte pikkustele. Murdunud palk eemaldatakse maailmast nagu kujutatud joonisel 15, aga mitte täielikult. Murdunud palgi kokkupuutekehad kustutatakse ning `hide()` meetodiga peidetakse ka palgi visuaalne osa, et hiljem seda ajatelje juures utiliseerida saaks.

5.1.5 Seinte graafiline implementatsioon

Seinte Godot mängumootoris toomiseks loodi kuuptahukas, mille pindadele pandi vatt ja tamp seintele visuaalselt sobiv tekstuur (vt Joonis 16). Lisaks on materjalil pinnanormalide tekstuur, mida kasutatakse pinna varjutamisel, et jätta mulje ebataasest pinnast. Tulemuseks saadud uut ja vananenud sein on kujutatud joonisel 16.



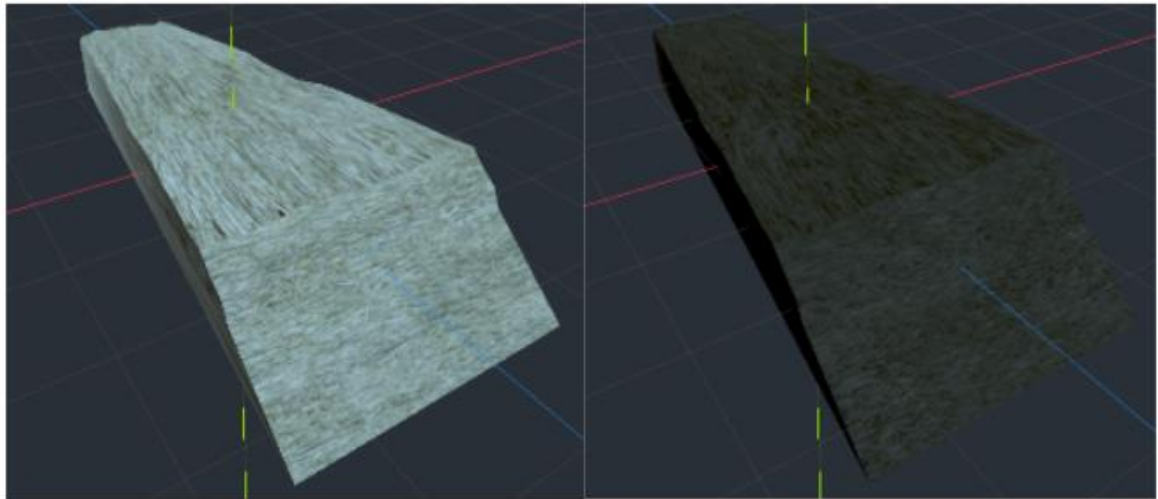
Joonis 16. Uus ja vananenud vatt ja tamp sein Godot mootoris.

Nagu eelnevalt kirjeldatud, ei ole täpset mudelit seina lagunemise protsessist. Sellest tulevalt üritatakse töös kujutada pigem visuaalselt, et sein on lagunenu. See kujutab endast seina kõrguse progressiivset kahanemist ja seina värvuse muutumist. Samuti on seinal võimetus katuse varisemisel ümber kukkuda.

5.1.6 Õlekõrtest pundarde graafiline implementatsioon

Õlekõrrepuntrad loodi sarnaselt seintele, aga kasutatav kujund modelleeriti Blenderit kasutades ning see on kujutatud joonisel 17. Üks õlekõrrepundarde objekt kujutab endast mitut, üksteise peale paigutatud õlekõrtest pundart, kiirendades nendega ehitiste konstrueerimist Godot mootoris.

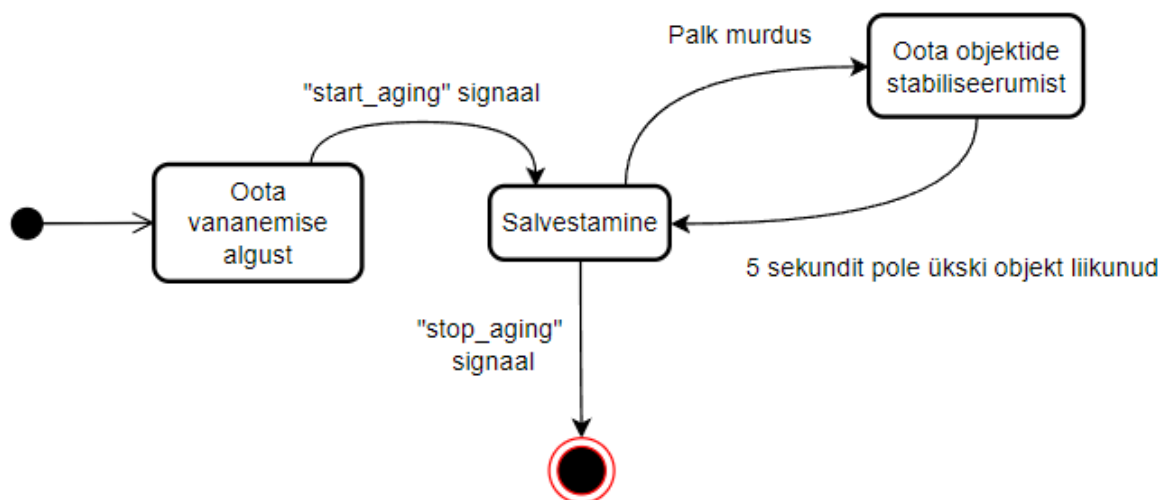
Õlekõrte vanandamisil muudetakse sarnaselt vatt ja tamp seintele ainult nende välimust. Nende värvus muutub ajas tumedamaks ning individuaalsete kõrte otsad näevad räsitud välja. Tulemuseks saadud värsket ja vananenud õlekõrrepundarde sektsiooni näeb üksteise kõrval joonisel 17.



Joonis 17. Uued ja vananenud õlekõrrepuntrad Godot mootoris.

5.2 Ajatelg

Joonisel 18 kujutatud ajatelje süsteemi implementeerimiseks jooksutatakse kõigepealt simulatsioon lõpuni, samal ajal perioodiliselt kogu stseeni andmeid salvestades. Juhul kui toimub lühiajaline tegevus, nagu puupalgi murdumine või katuse kokkuvarisemine, peatatakse andmete salvestamine ajutiselt. Peatamise põhjuseks on see, et ei tekiks ajateljel kerides olukorda, kus objektid kukuvad gravitatsiooni mõjul mitu aastat järjest. Salvestamine jätkub pärast ehitise olukorra stabiliseerumist. Lähenemisi nende andmete loomiseks on kaks: Godot füüsikamootoril tuginev ja manuaalne struktuurarvutus staatilistel objektidel.



Joonis 18. Salvestamise süsteemi olekumasin.

5.3 Godot füüsikamootoril tuginev algoritm

Esimene iteratsioon ajateljest kasutas Godot enda füüsikamootorit. Vähene kõikumine objektidevaheliste kokkupuutumiste registreerimises ei mänginud olulist rolli vananemise üldpildis, sest ka ajaloost ei ole täpseid kirjeldusi lagunemise kohta võtta.



Joonis 19. Puitraam Godot mootoris

Godot füüsikamootoril tuginev lähenemine töötas oma lihtsuse tõttu ainult lihtsakoelistemal ehitistel, mille juures ei olnud kasutatud ühendusi simuleeritavate materjalide vahel. Näiteks joonisel 19 on kujutatud sellist lihtsat ehitist, mis püsis koos vaid gravitatsiooni mõjul. Sellised ehitised aga ei peegelda reaalsust, eriti katuste puhul, kus on palgid omavahel kokku ühendatud. Selliste ühenduste loomiseks kasutati Godot mootori PinJoint objekte, millele pandi juurde ka katki minemise võimekus. Joonisel 20 on kujutatud sellist katusesektsiooni, mis kasutab PinJoint ühendusi.



Joonis 20. Katuse sektsioon. Palkidevahelised ühendused (PinJoint) tähistatud helesinises.

Objektide vahelised ühendused põhjustasid füüsikamootoriga seonduvaid probleeme. Esiteks tuli nende kasutamisel meeles pidada, et ühendust ei tohi luua parasjagu kokkupuutuvate objektide vahel, sest muidu üritavad seotud objektid end pidevalt lõhki rebida. Ei piisanud ka ühenduste ettevaatlikust ümber positsioneerimisest, sest lõpuks saadud struktuur võbeles konstantselt maapinnaga puutudes. See võbelus oli põhjustatud väikestest ebatäpsustest füüsikamootori enda jõuarvutuses. Lisaks põhjustas ehitise pidev võbelus puitstruktuurile toetava katuseroo kohese allakukkumise. Ainus viis olukorra lahendamiseks oli Godot füüsikamootori jõuarvutusest välistamine ning seetõttu mindi edasi teistsuguse lahendusega.

5.4 Area-objektidel tuginev algoritm

Eelnevalt kirjeldatud võbeluse välistamiseks loodi uus algoritm, kus Godot füüsikamootorit kasutati võimalikult vähe. Leiti, et ehitise eri sektsioonide kokku varisemine on ainus olukord, kus füüsikamootor asendamatu on, ning sellest tulenevalt kasutas uus lähenemine Godot füüsikamootorit ainult selles punktis. Struktuuri enda tugevuse arvutus tehti aga manuaalselt.

Manuaalse simuleerimise jaoks muudeti kõik stseenis asuvad objektid staatiliseks, kuid struktuurarvutuste jaoks tuli muuta kokkupuutumissignaali süsteemi. Nimelt ehitismaterjalide jaoks kasutatud füüsikalise keha (*RigidBody*) objekt, mis aktiivses olekus registreerib “body_shape_entered” signaale, ei tee seda staatilisena. Selle lahendamiseks loodi uus lahendus kokkupõrgete registreerimiseks Area-objektidega.

Area-objekti kasutati sellepärast, et ta on staatiline, aga samas registreerib samu signaale mida tavaline füüsikaline keha (*RigidBody*). Igale ehitismaterjalile tehti uus meetod, mis väljakutsumise järel tegi uue Area-tüüpi tütarobjekti iga materjalis oleva kokkupuutekeha kohta, pannes enda kujundiks koopia vastavast kehast. Samuti ühendati *connect()* meetodiga Area objekti poolt registreeritavad signaalid vastava ehitismaterjali kokkupuuteid registreerivate meetoditega, muutes struktuurarvutused taas funktsionaalseteks.

Põhjus tütarobjektide lisamises tulenes sellest, et kokkuvarisemisel peab objekt suutma käituda füüsikamootri põhiselt. Kui tahta staatilist objekti sellise lahenduse puhul tagasi füüsikaliseks teha, tuleb lihtsalt Area-objektid eemaldada ning füüsikaline keha seejärel aktiveerida. Kui eemaldada füüsikaline keha täielikult, siis poleks selline käitumine võimalik.

Staatilise puupalgi murdumise järgselt muudetakse murdunud objektile toetuvad objektid aktiivseteks. Aktiivseteks muutunud objektid tehakse uuesti staatiliseks siis, kui nad on lõpetanud liikumise. Liikumise lõpetamiseks kontrollitakse asukoha muutumist ning kui erinevus asukoha muutumises on piisavalt väike, käivitatakse objekti staatiliseks muutev kood.

6 Tulemused

Vananemise algoritmi vaatlemiseks ehitati Godot mängumootoris loodud materjalidega ehitis. Ehitise peal saab vananemise algoritmi jooksutada ning samal ajal vahetada ka nii tava-vaate kui ka ehitise struktuurvaate vahel. Pärast simulatsiooni lõpuni jooksmist saab ehitise erinevaid seise vaadata ajatelje peal edasi-tagasi kerides.

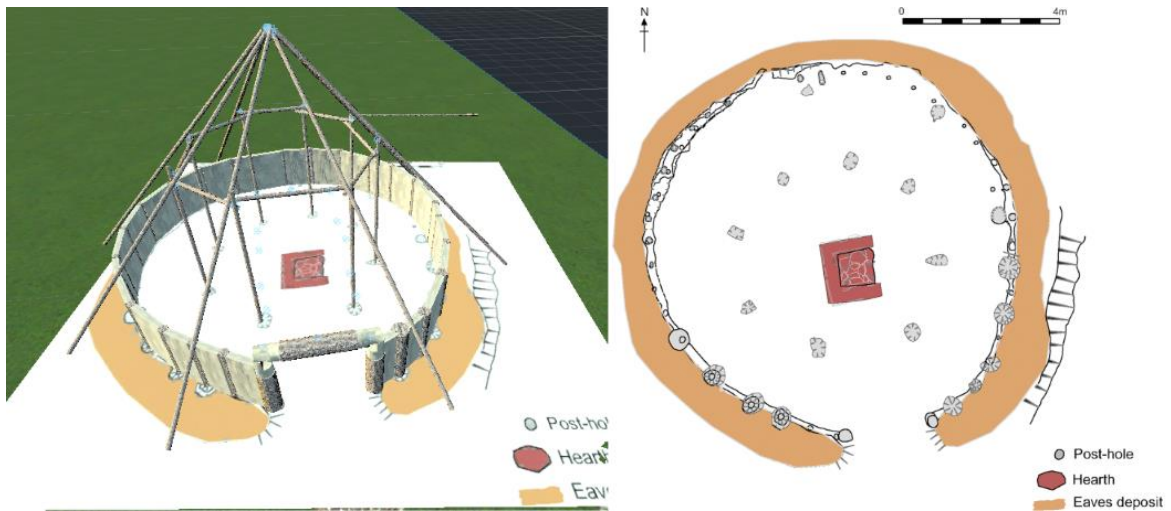
6.1 Inglismaa pronksiaegne ringmaja

Aluseks tulemuseks konstrueeritud ehitisele võeti inglismaal Castell Henlly'is rekonstrueeritud pronksiaja *roundhouse*, täpsemalt kokamaja [20]. Joonisel 21 kujutatud kokamaja valiti oma väiksema suuruse ja mitmete rekonstrueerimisfaaside tõttu, andes rohkem informatsiooni Godot mootoris ehitise rekonstrueerimiseks.



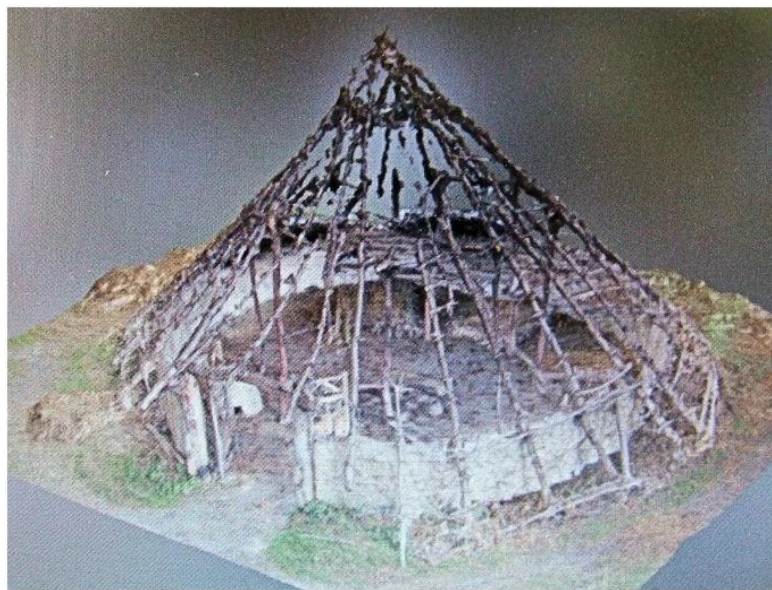
Joonis 21. Castell Henyll'is asuv pronksiaegne kokamaja. Pilt võetud 1984 [20].

Ringmajal oli kokku 4 faasi, kuid kasutamisele läks nendest kolmanda faasi arheoloogiliste leidude skeem. Kasutati just kolmanda faasi skeemi, kuna olid olemas sellele ehitise faasile vastavad pildid mulla pinnasesse tehtud aukudest. Need pildid aitasid kindlustada, et Godot sees jäljendatav versioon seintest oleks võimalikult lähedane päriselule.



Joonis 22. Kokamaja 3. faasi skeem ja Godot mootoris olev ehitis ilma katusega [20].

Ehitise konstrueerimiseks paigutati kõigepealt skeemi legendi peale palk, et pärismaailma ja Godot mõõtesüsteemide ühtlus tagada. Skeemi suurendati ühtlaselt, kuni sellel olev legend saavutas palgile vastava pikkuse. Järgnevalt püstitati skeemi põhjal seinad ja sissekäik, nagu on kujutatud joonisel 22. Samuti püstitati ehitise keskel asuv ring katust toetavatest palkidest.



Joonis 23. Katuse skaneeritud mudel [20].

Katuse ehitamiseks oli vaja kombineerida informatsiooni rekonstrueerimistöödest ja baas-teadmisi ehitusinseneeriast. Joonisel 23 on kujutatud 3D skaneerimise käigus saadud informatsioon rekonstrueerimisel kasutatud katusestruktuuri kohta [20]. Kujutatud katuse struktuur on väga paljude väiksemate okstega, mida täpselt joonis 23 järgi paigutada Godot

mootoris ei ole mõistlik. Suur okste arv mõjutaks negatiivselt simulatsiooni jõudlust. Seetõttu püstitati ainult katuse peamised diagonaalsed puupalgid. Pikkused nende palkide jaoks tuletati lähtudes sellest, et kõige optimaalsem sobiv katuse kallak ringmaja puhul on 45 kraadi. Saadud struktuurile asetati õlekõrtest puntrad, saades tulemuseks joonis 24 peal kujutatud ehitise.



Joonis 24. Inglise ringmaja Godot mootoris.

7 Algoritmi hindamine

Algoritmi hinnati mitmes aspektis. Hinnati ehitise lagunemise visuaalset sarnasust päriselu ehitistega ning jõudlust detailsemate ehitiste juures. Samuti hinnati algoritmi varasemate lahendustega ning puidu lagunemise mudeliga päriselust.

7.1 Visuaalne sarnasus reaalsusega

Kuna tulemuses esindatud ehitis on inspireeritud päriselus eksisteerivast ehitisest, on loogiline võrrelda nende vananenud välimusi omavahel. Selle jaoks võeti aluseks pilt nelja aasta vanusest kokamajast ning võrreldi simuleeritud tulemusega pärast 4 aastat vananemist. Mõlemaid ehitisi on kujutatud joonisel 25. Pilt aastast 1984, mida joonis 21 peal samuti kujutatud on, sobib võrdluseks hästi, sest esialgselt konstruktsioonist alates ei ole ringmajale parandusi tehtud.



Joonis 25. 4-aastane ringmaja Godot mootoris (vasakul) ja päriselus (paremal).

Esimese asjana jääb silma, et katuse värvus päriselu ehitisel ei ole ühtlane. Õlekõrred katuse tipu poole on heledamad ja rohkem kulunud, kui maapinnale lähedasemad. Algoritmi ehitis on aga ühtlase tooniga üle terve katuse, ilma et arvestaks ilmastikust tingitud ebahühtlast kulutamist.

Siiski on ringmaja sissepääse väga sarnane päriselu ehitisele pärast lühiajalist vananemist, aga leidub sealgi erinevusi. Algoritmi poolt kasutatud varjutaja tekitab liiga palju sammalt palgi allosasse, ning seinte värvused ei sobi päris täpselt kokku. Seinte värvuse erinevus võib olla põhjustatud võrreldava ehitise pildistamiseajast ehk selle aja kaamerate kvaliteedist.

Üldpildis näevad ehitised sarnased välja ning täpsustades vananedes toimuvaid protsesse algoritmis oleks võimalik saavutada ka sarnasem, realistlikum pilt.

7.2 Jõudlus

Algoritmi jõudlust hinnati stseenis olevate objektide arvu, nende omavaheliste kokkupõrgete arvu ning simulatsiooni jooksutamiseks kuluva aja põhjal. Tulemuseks saadud ehitises kasutati kokku 364 objekti, nendest 282 on katuse jaoks kasutatud õlekõrrepuntrad. Testimise jaoks kasutatud arvuti protsessor oli Ryzen 9 5900X ning graafikakaart Nvidia 1070Ti. Tulemuses oleva ehitise peal katsetati, kui mitu korda sekundis võib terve simulatsioon sekundis uueneda, enne kui liiga kiire uuendamine jõudlusele mõjub. Selgus, et vahemikus 0.6 - 1 korda sekundis simulatsiooni jooksutamine on kõige optimaalsem, kuna esines vähem maha jäänud kaadreid kui tihedama sagedusega.

Samuti avastati simulatsiooni kiirust muutes, et kokkupõrgete arv kokkuvarisemisel mõjutab oluliselt rakenduse kiirust. Ringmaja katuse sektsioonide kokkuvarisemisel tekib palju õlekõrrepundarde kokkupuutesignaale, mis aeglustavad oluliselt rakenduse kiirust. See probleem on pigem tingitud Godot kokkupuutumisi tuvastavast süsteemist, sest õlekõrrepuntras kokkupõrkel jooksutatav koodi keerukus ei ole suur.

Kokkuvõttes on rakenduse jõudlus alla soovitud tulemusest. See on tingitud lagunemisel tekkivast aeglusest ning selle parandamiseks tuleks vähendada kokku puutuvaid objekte. Katuse puhul saaks luua ühe suure katuse objekti, mille kuju saaks dünaamiliselt muuta ning lagunemise karakteristikuid eelnevalt defineerida, vähenedes selle juures signaalide arvu.

7.3 Võrdlus varasema tööga

Ei eksisteeri palju simulaatoreid, millega antud töö tulemusi võrrelda saaks. Eelnevalt mainitud Blenderi generaator mahajäetud ehitiste jaoks on ainus kandidaat visuaalse ja funktsionaalse kvaliteedi võrdlemiseks. Lisaks saab võrrelda materjalide lagunemist päriselus toimuvaga, peamiselt puidu lagunemist.

7.3.1 Blenderi generaator

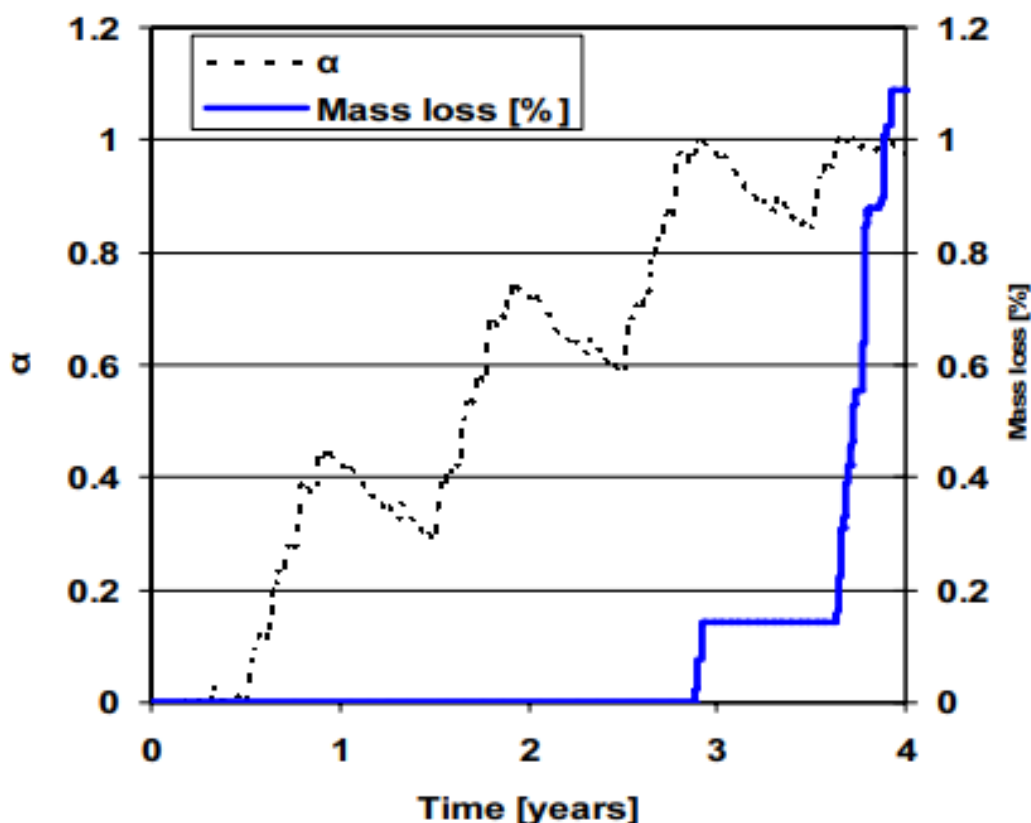
Võrreldes Sozapi valmistatud Blenderi generaatorit käesoleva töö tulemusega, on kerge näha, et Blenderist saadud mudel, mis on kujutatud joonisel 4, on parema välimusega. Generaator suudab puitu painutada, kivitükke tolmuks purustada ning kasutatud materjalid näevad üldpildis kvaliteetsemad välja. Samuti on generaatori esitlemiseks näideteks võetud tänapäevased ehitised, tehes tavainimesel saadud tulemuse reaalsuse hindamise lihtsamaks. Siiski teeb käesolev töö tasa visuaalsed puudujäägid oma võimekusega füüsilikas. Käesoleva

töö algoritmi implementatsioonis ei teki olukordi, kus objektid hõljuvad. Samuti on võetud lähenemine materjalide lagunemisele realistlikum, baseerudes materjalide füüsikalistel omadustel ja neis üle aja toimuvates protsessidel.

Samuti on käesoleva lahenduse plussiks asjaolu, et loodud algoritmi saab soovi korral implementeerida mistahes mängumootoris. Sozapi loodud generaator aga tugineb Blenderi sõlmepõhistele geomeetria funktsioonidele, limiteerides loodud lahenduse vaid Blenderis implementeerimiseks [3].

7.3.2 Puu massi kadu

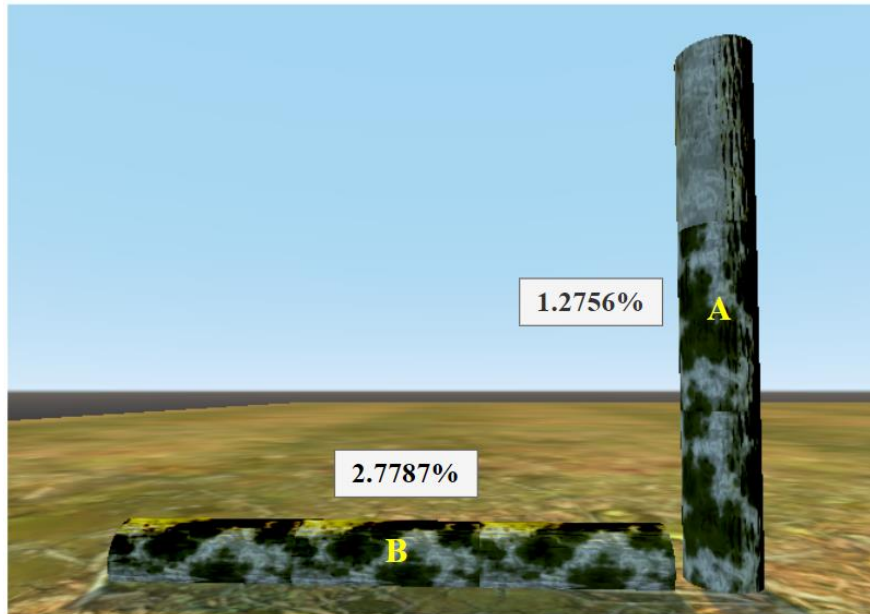
Tulemuse täpsust saab kontrollida ka päriselu põhjal. Algoritm on simuleeritud ka puu massi kadu üle aja, kasutades erinevaid kirjeldusi ning neid omavahel kombineerides [21]. Samuti on ka päriselus simuleeritud puitpalgis massi kadu nelja aasta jooksul, mida on kujutatud pideva sinise joonena joonisel 26.



Joonis 26. Massi kadu puitpalgis 4 aasta jooksul [21].

Sarnasuse kontrollimiseks jooksutati algoritmi kahe palgi peal: püsti seisva ja lamava palgi peal. Simuleeriti nelja aastat vananemist, mille tulemusena saadi püsti seisva palgi puhul

1.2756% massikadu ja lamava palgi puhul 2.7787%, mis on piisavalt lähedal joonisel 26 kujutatud andmetele. Palkide visuaalset välimust on kujutatud joonisel 27.



Joonis 27. Mädanenud puitpalkid Godot mootoris.

8 Tuleviku edasiarendused

Selles peatükis kirjeldatakse võimalikke algoritmi edasiarendusi ning teisi ehitiste vananemise nähtuseid, mida tulevikus algoritmide abil simuleerida. Esmalt räägitakse vananemise protsessi ajaperioodi pikendamisest ning sellega seonduvatest protsessidest.

8.1 Pikem ajaperiood

Loogiliseks edasiarenduseks vananemise algoritmile oleks pikendada simuleeritavat ajaperioodi. Selle jaoks tuleks uurida ja lisada simulatsiooni mõned protsessid. Näiteks pärast orgaanilise materjali mädanemist algab ka kõdunemisprotsess ning selle tulemusel jääb alles ainult muld. Kõdunemisprotsessi mõjutab palju faktoreid, aga protsessi lõpus saadud tulemused võivad olla ajaloolaste jaoks huvitavad.

Samuti saaks simuleerida ka maapinna kerkimist. Maapinna kerkimisel on samuti mitmeid faktoreid, aga huvitavaim neist on taimematerjali lagunemisel tekkinud mullapinna ladestumine ehitise jäätmetele. Selle järel tekiks ka arheoloogilised jäised, mille vaatlemiseks saaks luua eraldi vaate. Selles vaates võiks saada valida ekskaveeritava sektsiooni pinnasest, sarnaselt joonisel 28 kujutatud ristkülikukujulisele läbilõikele ekskaveeritud kraavist.



Joonis 28. Inglise ringmaja kraav ekskaveerituna [20].

8.2 Painumine ehitusmaterjalides

Mehaaniline surve paneb erinevaid ehitusmaterjale painduma ning piisava surve korral on tekkinud deformatsioon osaliselt jääv. Materjalide paindumise täpne implementeerimine on arvutuslikult väga kulukas protsess, mida on näha ka tänapäeva modelleerimistarkvarades, kus materjali moonet ning surve jaotust simuleerida saab. Nendes tarkvarades sõltub simulatsiooni pikkus objekti keerukusest, kuid enamus simulatsioone kestavad siiski minuteid. Paindumise implementeerimiseks mängumootoris tuleks vastavaid simulatsioone lihtsustada. Käesoleva töö raames uuriti ja katsetati puitpalkide deformatsiooni lisamist esimese ajatelje versiooni puhul, kuid Godot füüsikamootoris kokkupuudete registreerimiste süsteemi eripärasuste tõttu lõplikku tulemusse seda ei tulnud. Peamiseks takistuseks selle juures oli täpse kokkupuutepunkti puudumine kokkupuutel tagastatavas informatsioonis.

8.3 Rohkem ehitusmaterjale

Käesolevas töös realiseeriti ainult 3 ehitusmaterjali, aga keskaegsetes ehitistes ja tänapäeva ehitistes kasutatakse oluliselt rohkem erinevaid ehitusmaterjale. Kivist seinad ja metallist kinnituselemendid on samuti harilikult esinevad ehitusmaterjalid, kuid nende vananemisprotsess on keerulisem. Raua puhul on mulla sees vananemine mõjutatud suuresti mulla happesuse tasemest, mida käesolevas töös ei arvestatud, kuna see ei mõjuta oluliselt puidu vananemist maapinna peal. Kivi puhul mängivad rolli ka kivi omadused ja erinevad sambaliigid, mis kivi pinnal pragusid ja erosiooni põhjustavad.

8.4 Teised mängumootorid

Saadud algoritmi võiks katsetada erinevates mängumootorites, võrreldes nende jõudlust ja tulemuse visuaalset kvaliteeti. Samuti tuleks uurida võimalust, et kas mõnes teises mängumootoris saaks rohkem füüsikamootorile tugineda. See eemaldaks vajaduse manuaalselt mõjuvaid jõude arvutada ning muudaks ka algoritmi jõudluse paremaks. Sealhulgas võib proovida ka materjalide paindumist implementeerida.

9 Kokkuvõte

Käesolevas töös koostati vananemise algoritm Godot mängumootoris, sealhulgas uurides eri ehitusmaterjalide vananemisprotsesse. Samuti ehitati mängumootoris simuleeritud materjalidega inglismaa pronksiaegne ringmaja, mida saab vastavalt algoritmile vanandada. Vananemise käigus väheneb ehitise struktuurne tugevus ehitusmaterjalides toimuvate protsesside tõttu, põhjustades ehitise kokkuvarisemist.

Loodud algoritm toetab kolme ehitusmaterjali: puit, õlekõrred ning vatt ja tamp seinad. Iga materjali puhul uuriti selle mehaanilisi omadusi ning ajas toimuvaid muutusi. Puidu puhul uuriti selle paindetugevust mõjutavaid tegureid. Samuti uuriti mädaniku levimist nii puidus kui ka õlekõrtes ning selle mõju nende omadustele.

Algoritm implementeeriti Godot mängumootoris, selleks katsetati kahte lähenemist: Godot füüsikakehadel põhinevat ja manuaalsetel struktuurarvutustel põhinevat. Lõpptulemus kasutati manuaalsete struktuurarvutustega lähenemist, kuna esimese puhul tekkisid füüsikamootori käitumisest tingitud probleemid.

Samuti ehitati Godot mängumootoris inglise pronksiaegne ringmaja, mille peal vananemise algoritmi jooksutada saab. Simulatsiooni ajal saab kasutaja vahetada tavavaate ja struktuurvaate vahel, mis kirjeldab puupalkide stressi tasemeid antud hetkel. Pärast simulatsiooni lõpuni jooksmist saab ajatelje abil läbitud protsessi edasi-tagasi kerida.

Simulatsiooni täpsust analüüsiti mitmes aspektis, sealhulgas selle jõudlust, visuaalset ja füüsilist lähedust reaalsusele. Samuti uuriti võimalikke tuleviku edasiarenduste võimalusi.

Töö saavutas oma eesmärgid uurida vananemisel toimuvaid protsesse ehitistes. Implementeerimiseks tehti nendes protsessides mitmeid mööndusi, kuid lõplik tulemus esindab suuremat osa uuritud protsessidest rahuldava piirini.

10 Viidatud kirjandus

- [1] “The House of Tomorrow,” *Fallout Wiki*. https://fallout.fandom.com/wiki/The_House_of_Tomorrow (accessed May 17, 2022).
- [2] K. Kanazawa, R. Tanabe, T. Moriya, and T. Takahashi, “Rust aging simulation considering object’s geometries,” *SIGGRAPH Posters*, 2015, doi: 10.1145/2787626.2792641.
- [3] “An Abandoned House Generator Made in Blender.” <https://80.lv/articles/an-abandoned-house-generator-made-in-blender/> (accessed May 17, 2022).
- [4] K. Kránitz, W. Sonderegger, C.-T. Bues, and P. Niemz, “Effects of aging on wood: a literature review,” *Wood Sci. Technol.*, vol. 50, no. 1, pp. 7–22, Jan. 2016, doi: 10.1007/s00226-015-0766-0.
- [5] “A Beginner’s Guide | Thatching Info.com.” <https://thatchinginfo.com/beginners-guide/> (accessed May 17, 2022).
- [6] “Maintaining and repairing a thatched roof,” *Insurance Choice*, Nov. 15, 2019. <https://www.insurancechoice.co.uk/blog/2019/11/maintaining-and-repairing-a-thatched-roof> (accessed May 17, 2022).
- [7] “Rookatus,” *Vikipeedia*. Aug. 20, 2020. Accessed: May 17, 2022. [Online]. Available: <https://et.wikipedia.org/w/index.php?title=Rookatus&oldid=5699791>
- [8] “Maintaining a Thatched Roof | Thatching Info.com.” <https://thatchinginfo.com/maintaining-a-thatched-roof/> (accessed May 17, 2022).
- [9] D. E. Friesem, G. Tsartsidou, P. Karkanis, and R. Shahack-Gross, “Where are the roofs? A geo-ethnoarchaeological study of mud brick structures and their collapse processes, focusing on the identification of roofs,” *Archaeol. Anthropol. Sci.*, vol. 6, no. 1, pp. 73–92, Mar. 2014, doi: 10.1007/s12520-013-0146-3.
- [10] C. Udawatthe, R. Arooz, and R. Halwatura, “NEW EARTH WALLING MATERIAL: INTEGRATING MODERN TECHNOLOGY INTO ANCIENT MUD WALL,” Dec. 2016.
- [11] T. Busser, J. Berger, A. Piot, M. Pailha, and M. Woloszyn, “Comparative Study of Three Models for Moisture Transfer in Hygroscopic Materials,” *Transp. Porous Media*, vol. 126, no. 2, pp. 379–410, Jan. 2019, doi: 10.1007/s11242-018-1161-8.
- [12] D. W. Green, J. E. Winandy, and D. E. Kretschmann, “Mechanical Properties of Wood,” p. 46.
- [13] “Normaalelastsusmoodul,” *Vikipeedia*. Dec. 10, 2018. Accessed: May 17, 2022. [Online]. Available: <https://et.wikipedia.org/w/index.php?title=Normaalelastsusmoodul&oldid=5166497>
- [14] “How to Determine the Modulus of Rupture,” *Sciencing*. <https://sciencing.com/determine-modulus-rupture-12046932.html> (accessed May 17, 2022).
- [15] C. Brischke and G. Alfredsen, “Wood-water relationships and their role for wood susceptibility to fungal decay,” *Appl. Microbiol. Biotechnol.*, vol. 104, no. 9, pp. 3781–3795, May 2020, doi: 10.1007/s00253-020-10479-1.
- [16] J. Wang, X. Cao, and H. Liu, “A review of the long-term effects of humidity on the mechanical properties of wood and wood-based products,” *Eur. J. Wood Wood Prod.*, vol. 79, no. 2, pp. 245–259, Mar. 2021, doi: 10.1007/s00107-020-01623-9.
- [17] J. J. H. Kirby and A. D. M. Rayner, “The deterioration of thatched roofs,” *Int. Biodeterior.*, vol. 25, no. 1, pp. 21–26, Jan. 1989, doi: 10.1016/0265-3036(89)90025-0.
- [18] M. T. M. Bosman, “Some Effects of Decay and Weathering on the Anatomical Structure of the Stem of *Phragmites Australis* Trin. Ex Steud.,” *IAWA J.*, vol. 6, no. 2, pp. 165–170, Jan. 1985, doi: 10.1163/22941932-90000929.

- [19] “5.2.1 Decay of Daub and Plaster.” http://www.tonygraham.co.uk/house_repair/wattle_daub/WD-5_2_1.html#fnB78 (accessed May 17, 2022).
- [20] H. Mytum and J. Meek, “Experimental archaeology and roundhouse excavated signatures: the investigation of two reconstructed Iron Age buildings at Castell Henllys, Wales,” *Archaeol. Anthropol. Sci.*, vol. 12, no. 3, p. 78, Feb. 2020, doi: 10.1007/s12520-020-01028-y.
- [21] H. Viitanen *et al.*, “Towards modelling of decay risk of wooden materials,” *Eur. J. Wood Wood Prod.*, vol. 68, no. 3, pp. 303–313, Aug. 2010, doi: 10.1007/s00107-010-0450-x.

Lisad

I. Rakenduse paigaldamine ja kasutamine

Algoritmi implementatsiooni käigus valminud rakendus on tööga kaasas olevas zip-failis. Rakendus on ehitatud Windowsi operatsioonisüsteemi peal olevatele arvutitele. Rakenduse kasutamiseks tuleb alla tõmmatud zip-fail lahti pakkida ning käivitada fail nimega „Building aging sim.exe“.

Rakenduses saab simulatsiooni alustada vajutades nuppu „Start aging“. Simulatsiooni ajal saab struktuurvaadet ehitise palkidele mõjuvast jõust näha „Structure view“ kasti sisse-välja lülitades. Simulatsiooni saab lõpetada vajutades nuppu „Stop aging“. Simulatsiooni lõppes saab ajatelje abil vaadelda ehitise eri staadiume, kasutades akna alumises osas asuvat liugurit.

II. Rakenduse lähtekood

Rakenduse lähtekood on kättesaadav samast GitHubi repositooriumist:

<https://github.com/towrig/building-aging-sim>

III. Litsents

Lihlitsents lõputöö reprodutseerimiseks ja üldsusele kättesaadavaks tegemiseks

Mina, _____ Karl Kuusik _____,

(autori nimi)

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

_____ Ehitise vananemise simuleerimine Godot mängumootoriga _____,

(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 lihlitsentsi andmisega ei riku ma teiste isikute intellektuaalomandi ega isikuandmete kaitse õigusaktidest tulenevaid õigusi.

Karl Kuusik

17.05.2022