**RĪGAS TEHNISKĀ UNIVERSITĀTE**

Datorzinātnes un informācijas tehnoloģijas fakultāte

Lietišķo datorsistēmu institūts

**Domeniks Delvers**

bakalaura akadēmisko, pirmā līmeņa studiju programmas

students, stud. apl. nr. 201RDB298

**PĀRKLĀJUMA CEĻA PLĀNOŠANAS ALGORITMU SALĪDZINĀJUMS**

**PUTEKĻSŪCĒJU ROBOTIEM**

**BAKALAURA DARBS**

Zinātniskais vadītājs Doktors, Profesors

Agris Ņikitenko

RĪGA 2022

**RĪGAS TEHNISKĀ UNIVERSITĀTE**

**DATORZINĀTNES UN INFORMĀCIJAS TEHNOLOĢIJAS FAKULTĀTE**

<institūta nosaukums>

<struktūrvienības nosaukums>

**<Noslēguma darba no saraksta: bakalaura darbs, maģistra darbs, diplomprojekts, kvalifikācijas darbs> izpildes lapa**

Noslēguma darba autors:

students(-e) <vārds, uzvārds>\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(paraksts, datums)

Noslēguma darbs ieteikts aizstāvēšanai:

Zinātniskais vadītājs:

<zinātniskais grāds, amats, vārds, uzvārds> \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(paraksts, datums)

Anotācija

Putekļsūcēj-robots, simulācija, ceļa plānošanas algoritms

Mūsdienās mājas tīrīšanas roboti paliek arvien izplatītāki, un īpaši – putekļsūcēj-roboti. Viena no īpašībām, kas raksturo robotu ir tā tīrīšanas ātrums, tāpēc ražotāji mēģina izstrādāt pēc iespējas efektīvākus algoritmus, lai robots spētu iztīrīt visu dzīvokļa platumu pēc iespējas mazākā laikā. Šajā darbā tiek apskatīti vairāki tādi algoritmi, kā arī simulēti virtuālajā vidē un salīdzināti dažādos apstākļos.

Darbā ir X lappušu, Y attēlu, Z tabulu, W pielikumu un Q izmantoto avotu skaits.

Abstract

Nowadays, home cleaning robots are becoming more and more popular, particularly vacuum-cleaning robots. One of the characteristics that defines a robot is its cleaning speed, so manufacturers are trying to develop as efficient algorithms as possible so that the robot can clean the entire width of the apartment in the shortest possible time. In this work, several such algorithms are examined and simulated in a virtual environment and compared under different conditions.

Satura rādītājs

[Anotācija 2](#_Toc130745051)

[Abstract 3](#_Toc130745052)

[Satura rādītājs 4](#_Toc130745053)

[Ievads 5](#_Toc130745054)

[1. Teorijas pārskats 7](#_Toc130745055)

[1.1. Tradicionālas pārklājuma meklēšanas tehnikas 7](#_Toc130745056)

[1.1.1. Šūnu dekompozīcija 7](#_Toc130745057)

[1.1.2. Bustrofedona dekompozīcija 7](#_Toc130745058)

[1.1.3. Režģa bāzētās metodes 8](#_Toc130745059)

[1.1.4. Aptveroša koka metodes 8](#_Toc130745060)

[1.2. Apmācība ar pastiprinājumu vienīga aģenta pārklājuma ceļa plānošanai 8](#_Toc130745061)

[1.2.1. Q – apmācība 9](#_Toc130745062)

[1.2.2. Dziļie Q – tīkli 9](#_Toc130745063)

[1.2.3. Aktiera – kritiķa metodes 10](#_Toc130745064)

[1.2.4. Monte Carlo koka pārmeklēšana 10](#_Toc130745065)

[1.3. Apmācība ar pastiprinājumu vairāku aģentu pārklājuma ceļa meklēšanai 10](#_Toc130745066)

[1.3.1. Neatkarīgi mācekļi 10](#_Toc130745067)

[1.3.2. Kopīgu darbību mācekļi 10](#_Toc130745068)

[1.3.3. Centralizēta apmācība ar decentralizētu izpildi 11](#_Toc130745069)

[1.3.4. Komunikācijas pamatotas pieejas 11](#_Toc130745070)

[2. Iekštelpas vides procedurāla ģenerēšana 12](#_Toc130745071)

[1.1. Vides skeleta veidošana 12](#_Toc130745072)

[1.1. Durvju izvietojums 13](#_Toc130745073)

[1.2. Mēbeļu un šķēršļu ģenerēšanā 14](#_Toc130745074)

Ievads

Pēdējos gados ir palielinājusies interese pret pilna pārklājuma ceļa meklēšanas algoritmu attīstīšanu robotizēto sistēmu un mākslīgā intelekta nozarēs. Iespēja efektīvi izbraukt un pārklāt dažādas kompleksas, dinamiskas vides ir ļoti svarīga vairākiem uzdevumiem: glābšanas operācijām, drošības uzraudzībai, kartēšanai, vides monitoringam un īpaši tīrīšanai. Mājas tīrīšanas robotu tirgū jau no 2014 gada ir novērojama stabila izaugsme. Palielinoties interesei pēc gudrās mājas tehnoloģijas un automatizācijas, pieprasījums pēc mājas tīrīšanas robotiem arvien aug. Īpaša tirgus izaugsme tika novērota COVID – 19 pandēmijas laikā. Sakarā ar pandēmiju daudzi cilvēki bija spiesti pavadīt vairāk laika mājās, tāpēc palielinājās interese pēc produktiem, kas ļauj saturēt dzīvokļus tīrākus. Saskaņā ar “*Research and Markets*” pētījumu, starp 2021. gadu un 2026 gadu, var sagaidīt 20% salikto gada pieauguma tempu (*Compound Annual Growth Rate*). Zinātniskais raksts pamato tik strauju izaugsmi ar palielinātu pieprasījumu pēc inovatīviem mājas tīrīšanas veidiem, kā arī gudrās mājas tehnoloģijas izplatību. Mājas tīrīšanas robotu tirgū ir daudz konkurējošo ražotāju, piemēram: “*iRobot*”, “*Dyson*”, “*LG Electronics*”, “*Ecovacs* *Robotics*” un “*Samsung* *Electronics*”. Augot pieprasījumam pēc mājas tīrīšanas robotiem, ražotāji cenšas uzlabot savu robotu efektivitāti, lai piesaistītu vairāk klientu.

Īpaši būtiski robotu efektivitāti ietekmē tā programmatūra: navigācijas algoritms, kartēšanas algoritms un ceļa plānošanas algoritms. Lielākā daļā tieši no ceļa plānošanas algoritma ir atkarīgs, cik ātri un kvalitatīvi robots paveiks savu uzdevumu. Algoritmam ir jānodrošina pilns telpas pārklājums, šķēršļu apbraukšana, kā arī jāatrod ceļš ar minimālo garumu un mazāku pagriezienu skaitu. Lai veicinātu pilnu pārklājumu ražotāji izmanto dažādas metodes, kuras savukārt ir atkarīgas no vienlaicīgas lokalizācijas un kartēšanas (*SLAM*) sistēmas precizitātes. Mūsdienīgie tīrīšanas roboti izmanto heiristiskas metodes pārklājuma ceļa meklēšanai vairāku iemeslu dēļ. Pirmkārt *SLAM* sistēmu neprecizitātes dēļ ir grūti precīzi izbraukt iepriekš ieplānoto trajektoriju, kā arī nav iespējams pietiekami precīzi nokartēt vidi. Otrkārt visi mūsdienās pastāvošie pilna pārklājuma algoritmi ir nepiemēroti mājas apstākļiem.

Šodien neviens ražotājs nepiedāvā efektīvu risinājumu klientiem, kuri vēlās izmantot savās mājās vairākus tīrīšanas robotus vienlaicīgi. Mūsdienas tīrīšanas robotiem nav sistēmu, kas nodrošinātu efektīvu vairāku aģentu sadarbību kopīgas problēmas risināšanai.

Ceļu plānošanas uzdevums dzīvokļos un mājās ir izaicinošs vides nenoteiktības un šķēršļu klātbūtnes dēļ. Viens veids, kā efektīvi meklēt risinājumu šīm izaicinājumam, ir algoritma testēšanas laikā izmantot procedurāli ģenerētas vides, kurās vides plāns un šķēršļi tiek ģenerēti algoritmiski, nevis tiek iepriekš definēti. Tas ļauj izveidot ļoti dažādas un sarežģītas vides ar lielu variāciju. Šādās vidēs var testēt dažādu ceļu plānošanas algoritmu veiktspēju. Šajā darbā tiek salīdzināti dažādi pilna pārklājuma ceļa meklēšanas algoritmi procedurāli ģenerētās telpās, ar fokusu uz mājas tīrīšanas robotiem. Darbā tiek salīdzināti dažādi algoritmi pēc to ātruma, efektivitātes un izturības.

Darba mērķis ir simulēt vairāku aģentu pārklājuma problēmas risinošo algoritmu virtuālajā vidē, pie dažādiem apstākļiem un izpētīt piedāvātā risinājuma efektivitāti. Lai veiksmīgi sasniegtu šo mērķi tika izvirzīti sekojoši darba uzdevumi:

1. Sagatavot virtuālo vidi eksperimentu veikšanai. Šīm nolūkam tika izmantota “Unity 3D”, kur procedurāli tiek ģenerēti dažādi dzīvokli.
2. Īstenot istabas apiešanas algoritmu, ar kuru palīdzību var veiksmīgi apiet iepriekš ģenerētas vides.
3. Savākt datus un normalizēt tos, lai veiktu pilnvērtīgus secinājumus.

Darba pirmajā nodaļā ir aprakstīts iekštelpas vides procedurālas ģenerēšanā “Unity3D” vidē. Darba otrajā nodaļā tiek aprakstīti testētie pilnās apiešanas algoritmi, kā arī pamatota to izvēle. Trešajā nodaļā tiek aprakstīts testēšanas, datu savākšanas un secināšanas process.

1. Teorijas pārskats
   1. Tradicionālas pārklājuma meklēšanas tehnikas

Pārklājuma ceļa meklēšana (CPP) ir bieži sastopama problēma robotikā un automatizācijā, kur mērķis ir atrast optimālu vai tuvu optimālam ceļu, kas ļauj robotam vai vairākiem robotiem apmeklēt katru punktu dotās vides teritorijā, minimizējot zudumus, piemēram, laiku vai enerģijas patēriņu. Šī problēma ir nozīmīga dažādās jomās, tostarp vides monitoringā, inspekcijā, lauksaimniecībā, tīrīšanā, meklēšanā un glābšanā, kā arī uzraudzībā.

Tradicionālās pārklājuma ceļa plānošanas tehnoloģijas galvenokārt balstās uz ģeometriskām, grafu bāzētām vai heiristikām pieejām. Šīs metodes parasti tiešās uz algoritmu izstrādi, kas radītu efektīvu pārklājuma ceļu dotajai videi, ņemot vērā vairākus faktorus, piemēram, robota uztveres un darbības spējas, vides ģeometriju, struktūru un iespējamos šķēršļus. Daži no svarīgākajiem faktoriem, ko jāņem vērā tradicionālo CPP algoritmu izstrādē, ietver: pārklājuma pilnība, ceļa optimizēšana, skaitļošanas sarežģītība, mērogošana un izturība.

* + 1. Šūnu dekompozīcija

Šūnu dekompozīcija ir tehnika, kas sadala vidi vairākos nepārklājošās šūnās, kuras nākotnē varētu apceļot aģents. Viens no agrākajiem darbiem šajā nozarē pieder *Moravec* un *Elfes* kungiem (1985), kas pielietoja okupācijas režģa pamatā esošo pieeju robotu kartēšanai un navigācijai (High resolution maps from wide angle sonar). Vēlāk, zinātnieki *Choset* un *Pignon* (1997) piedāvāja bustrofedona dekompozīcijas metodi, kas sadala vidi šūnās, kas robotiem ir viegli šķērsojamas (Coverage Path Planning: The Boustrophedon Cellular Decomposition).

* + 1. Bustrofedona dekompozīcija

Bustrofedona dekompozīcija, ko ieviesa *Choset* un *Pignon* (1997), ir populāra šūnu dekompozīcijas pieeja ceļu plānošanai, kas nodrošina pilnīgu pārklājumu (Coverage Path Planning: The Boustrophedon Cellular Decomposition). Šī tehnika sadala vidi šūnās, kuras pēc tam aģents šķērso veicot “S” veida kustības. Pētnieki ir turpinājuši attīstīt un uzlabot šo pieeju, piemēram *Acar* (2002) darbā, kur tie piedāvāja pilnīgu ceļu plānošanas algoritmu robotiem ar ierobežotām sensoru spējam (Sensor-based Coverage of Unknown Environments: Incremental Construction of Morse Decompositions).

* + 1. Režģa bāzētās metodes

Uz režģi bāzētās pieejas diskretizē vidi, lai iegūtu vides režģi un pēc tam izmanto grafu bāzētos algoritmus pārklājuma nodrošināšanai. Piemēram, *Zelinsky* (1993) ieviesa viļņa frontes bāzēto pārklājuma ceļa meklēšanas algoritmu, kas izmanto režģa rūtiņas, kā pamatelementus vides izmeklēšanai (Planning Paths of Complete Coverage of an Unstructured Environment by a Mobile Robot). Vēlāk *Huang* (2001) piedāvāja tā saucamo “optimālo nejaušo pastaigu” režģa bāzētu pārklājuma ceļa meklēšanas algoritmu (Optimal line-sweep-based decompositions for coverage algorithms).

* + 1. Aptveroša koka metodes

Tehnikas, kuras pamatojās uz aptveroša koka metodi, veido koku, kas aptver visu vidi, un kuram izseko aģents, lai paveiktu pilnu vides pārklāšanu. *Gabriely* un *Rimon* (2001) ieviesa aptveroša koka pārklājuma (STC) algoritmu, kas veido aptverošo koku par režģa pamatā esošo vidi un garantē pilnīgu pārklājumu (Spanning-tree based coverage of continuous areas by a mobile robot). Šo pieeju ir turpmāk izpētījusi un paplašinājuši citi pētnieki, piemēram *Hazon* un  *Kaminka* (2005) darbā, kurā tie piedāvāja tiešsaistes algoritmu STC metodei (Redundancy, Efficiency and Robustness in Multi-Robot Coverage).

* 1. Apmācība ar pastiprinājumu vienīga aģenta pārklājuma ceļa plānošanai

Viena aģenta pastiprinātā mācīšanās (RL) pārklājuma ceļa plānošanai ir pāreja no tradicionālajām noteiktajām un ģeometriskajām metodēm uz adaptīvāku un datu-definēto pieeju. Pastiprinātā mācīšanās ir mašīnmācīšanās nozare, kur aģents mācās pieņemt lēmumus, mijiedarbojoties ar vidi un saņemot atsauksmes veidā balvas vai soda. Aģenta galvenais mērķis ir maksimizēt kopējo atalgojumu laikā.

Viena aģenta RL piemērošana pārklājuma ceļa plānošanai nodrošina vairākas priekšrocības salīdzinājumā ar tradicionālajām metodēm:

* Pielāgojamība: RL algoritmi var mācīties pielāgoties dažādām vides situācijām, ieskaitot dinamiskas vai neskaidras situācijas. Viņi var modificēt savu rīcību, pamatojoties uz reāllaika atsauksmēm no vides, kas padara tos piemērotākus, lai rīkoties dinamiskās vidēs.
* Vispārīgums: Tā kā RL algoritmi mācās no datiem, tie var potenciāli labi vispārināties jaunām un neredzētām vides situācijām. Tas nozīmē, ka, pēc trenēšanās dažādos vides apstākļos, RL aģents varētu efektīvi darboties iepriekš neredzētās situācijās.
* Mērogojamība: RL algoritmi var potenciāli mērogot līdz lielām un sarežģītām vides situācijām, jo tie nebalstās uz skaidrām vides reprezentācijām. Tā vietā tie mācās pieņemt lēmumus, pamatojoties uz lokāliem novērojumiem.
* Pastāvīga mācīšanās: RL aģenti var turpināt mācīties un uzlabot savu veiktspēju pat pēc sākotnējās apmācības. Tas ļauj tiem precizēt savas politikas un pielāgoties jaunām situācijām vai vides izmaiņām laikā.

1.2.1. Q – apmācība

*Engel* (2005) demonstrēja Q – apmācības potenciālu vienīga aģenta pārklājuma ceļa plānošanas uzdevuma risināšanā (Reinforcement learning with Gaussian processes). Pētnieks piemēroja Q – apmācības metodi lauksaimnieciskas vides autonomām izsmidzināšanas nolūkam un pieradīja tehnoloģijas efektivitāti.

1.2.2. Dziļie Q – tīkli

Dziļie Q – tīkli (DQN) paplašina tradicionālo Q – mācīšanas metodi, izmantojot dziļos neironu tīklus funkciju aproksimācijai. *Mnih* (2015) veiksmīgi pielietoja DQN “Atari” konsoles spēlēm (Human-level control through deep reinforcement learning). Lai gan DQN nav plaši pielietoti pārklājuma ceļa plānošanai, to veiksmīgs pielietojums komplicētās vidēs pierāda potenciālu pielietojamību.

1.2.3. Aktiera – kritiķa metodes

Aktiera – kritiķa metodes ir piemērotas pārklājuma ceļa plānošanai, kā redzams *Konda* un *Tsitsiklis* (2002) darbā, kas prezentēja aktiera – kritiķa algoritmu mācīšanas ar pastiprinājumu metodei (On actor-critic algorithms). Viņi pierādīja, ka algoritms spej mācīties un pielāgoties dažādām vidēm un apstākļiem.

1.2.4. Monte Carlo koka pārmeklēšana

Monte Carlo koka pārmeklēšana “MCTS” ir ceļa plānošanas algoritms, kas apvieno Monte Carlo simulācijas ar koka pārmeklēšanu. MCTS tika veiksmīgi pielietots vienīga aģenta pārklājuma ceļa plānošanai, kā to pierādījis *Coulom* (2006), kurš izmantoja MCTS spēlei “Go” (Efficient selectivity and backup operators in Monte-Carlo tree search).

* 1. Apmācība ar pastiprinājumu vairāku aģentu pārklājuma ceļa meklēšanai

Vairāku aģentu pastiprinātā mācīšanās (MARL) paplašina viena aģenta pastiprinātās mācīšanās, kur vairākiem aģentiem ir jākoordinējas un jāsadarbojas, lai sasniegtu kopējo mērķi. Pārklājuma ceļa plānošanas kontekstā vairāku aģentu pastiprinātā mācīšanās var tikt izmantota, lai izstrādātu efektīvus algoritmus, kas koordinētu robota komandu, lai kopīgi pētītu un pārklātu vidi.

Daudzaģentu pastiprinātās mācīšanās (MARL) pielietošana pārklājuma ceļa plānošanai sniedz vairākas priekšrocības:

* Sadalītā lēmumu pieņemšana: MARL algoritmi var atļaut sadalītu lēmumu pieņemšanu starp vairākiem robotiem, ļaujot tiem strādāt kopā un pielāgot savu rīcību, pamatojoties uz lokālo informāciju un sadarbību ar citiem aģentiem. Tas var novest pie uzlabotas efektivitātes un izturības salīdzinājumā ar centralizētām lēmumu pieņemšanas pieejām.
* Mērogojamība: Vairāku aģentu pastiprinātā mācīšanās var potenciāli labāk mērogoties uz lielām un sarežģītām vidēm, jo darba slodze ir sadalīta starp vairākiem aģentiem. Katrs aģents mācās pieņemt lēmumus, pamatojoties uz saviem vietējiem novērojumiem samazinot kopējo problēmas skaitļošanas sarežģītību.
* Izturība: MARL var novest pie izturīgākiem risinājumiem neskaidru situāciju un dinamisku vides apstākļu gadījumā. Izmantojot vairāku aģentu kolektīvo intelektu, sistēma var labāk tikt galā ar negaidītiem šķēršļiem, vides izmaiņām vai pat ar individuālo aģentu neveiksmēm.
* Uzdevumu specializācija: Vairāku aģentu sistēmas var nodrošināt uzdevumu specializāciju starp aģentiem, kur katrs aģents mācās izcili pārvaldīt konkrētus apakšuzdevumus pārklājuma ceļa plānošanas problēmā. Tas var novest pie efektīvākiem pārklājuma ceļa veidošanas stratēģijām.

1.3.1. Neatkarīgi mācekļi

Neatkarīga Q – mācīšanas metode tika pielietota vairāku aģentu pārklājuma ceļa plānošanai *Matignon* (2007) darbā, kas ieviesa sadarbības Q – mācīšanas algoritmu vairāku aģentu pārklājuma ceļa meklēšanai (Independent reinforcement learners in cooperative Markov games: A survey regarding coordination problems). Darbs pierāda, ka pieeja var efektīvi koordinēt vairākus robotus, risinot nenoteiktības dinamiskās vidēs.

1.3.2. Kopīgu darbību mācekļi

Kopīgu darbību mācības metode ņem vērā visu aģentu kopējās darbības mācību procesā. *Oliehoek* (2006) pielietoja kopīgu darbību mācības metodi vairāku aģentu pārklājuma ceļa meklēšanai, prezentējot decentralizētu algoritmu, kas ir balstīts uz vairāku aģentu Markova lēmumu procesiem (MMDP) (Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems). Pētnieks demonstrēja algoritma mērogojamību un izturību dažādos testēšanas scenārijos.

1.3.3. Centralizēta apmācība ar decentralizētu izpildi

Šī pieeja centralizēti apmāca visus aģentus un pēc tam ļauj tiem izpildīt savas politikas decentralizētā veidā. *Lowe* (2017) ieviesa vairāku aģentu aktiera – kritiķa metodi sadarbības un konkurējošās vidēs (Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments). Pētījums apskata šādas pieejas pielietojamību dažādiem sadarbības un konkurējošās vides uzdevumiem. Šajā darbā pieeja nav tiešā veidā pielietota pārklājuma ceļa meklēšanai, tomēr darbs demonstrē šīs pieejas potenciālu vairāku aģentu sistēmās.

1.3.4. Komunikācijas pamatotas pieejas

Komunikācijas iekļaušana starp aģentiem var ievērojami uzlabot koordināciju un veiktspēju vairāku aģentu mašīnmācības procesā. *Foerster* (2016) ieviesa diferencējamo starp-aģentu mācīšanas (DIAL) struktūru, kas ļauj aģentiem mācīties komunikācijas politikas, izmantojot *backpropagation* (Counterfactual Multi-Agent Policy Gradients)*.* Pētnieku ieviestā pieeja var būt piemērojama vairāku aģentu pārklājuma ceļa plānošanai, kur koordinācija un komunikācija starp aģentiem ir būtiska.

1. Iekštelpas vides procedurāla ģenerēšana

Procedurālā iekštelpu vides ģenerēšana ir tehnika, kas ļauj virtuāli veidot telpas un vides. Šī tehnika izmanto algoritmus un noteikumus, lai automātiski ģenerētu nejaušo telpas plānojumu, izmantojot programmatūru. Rezultējošās telpas var tikt izmantotas video spēlēs, simulācijās un citās interaktīvās lietojumprogrammās. Procedurāli ģenerēto vižu labums ceļa meklēšanas algoritmu testēšanai ir tas, ka procedurāli ģenerētas vides piedāvā lielāku variāciju telpu savstarpējam novietojumam, un šķēršļu novietojumam. Algoritmu testēšana telpās, kas pastāv reālajā dzīvē, ierobežo apstākļu variāciju, jo lielākoties dzīvokļu plānojumi balstās uz vienādiem arhitektūras un dizaina principiem, secīgi – tiem bieži ir līdzīgas īpašības. Savukārt robota efektivitāti neietekmē cik skaisti, vai ērti ir izvietotas mēbeles istabā, to ietekmē tikai iespēja precīzi un kvalitatīvi plānot savu ceļu jebkurā vidē. Šajā darbā, procedurālo vides ģenerēšanu var sadalīt trijos būtiskajos posmos: vides skeleta veidošana, durvju izvietošana un mēbeļu izvietošana.

* 1. Vides skeleta veidošana

Vides skelets, šī darba kontekstā ir primitīva vide, kas sastāv no vairākām ortogonālām istabām, kurām ir noteikti izmēri, izvietojums telpā un tips. Vides skeleta būvēšana ir būtiskākais solis procedurālā ģenerēšanā, jo tas nosaka kopējo vides formu un vides kopējo platību. Šajā algoritmā katra istaba ir istabas (*Room*) klases objekts, kas glabā informāciju par istabas izmēriem, istabas centra koordinātēm, istabas stūru koordinātēm, istabas tipu un istabas identifikācijas numuru. Jaunas istabas būvēšana sākās ar nejaušo istabas tipa izvēli, piemēram “virtuve” (*kitchen*). Pēc tipa izvēlēšanas, algoritms nejauši izvēlās jaunas istabas izmērus, ar ierobežojumiem, kuri ir atkarīgi no istabas tipa. Zinot istabas izmērus, algoritms izmeklē visas potenciālas izvietošanas iespējas. Izvietošanai ir daži ierobežojumi: istaba nedrīkst pārklāt vai krustot citu istabu, neviens no istabas punktiem nedrīkst atrasties zem nulles koordinātes x vai z asī. Pēc visu potenciālu izvietošanas punktu atrašanas, uz nejauši izvēlēta punkta tiek būvēta jauna istaba. Process atkārtojās tikmēr nebūs izveidots nejauši izvēlētais istabu skaits.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| |  |  | | --- | --- | |  |  | |  |  | |
| 1.att. Procedurāli ģenerēto vides skeletu piemēri |

* 1. Durvju izvietojums

Nākamais solis vides ģenerēšanā ir durvju ģenerēšana. Darba mērķim nav vajadzīgas fiziskas atveramas durvis, tāpēc to vietā ir noteikta platuma caurumi sienās, kas ļaus aģentam brīvi pārvietoties starp visām istabām. Lai sasniegtu visu istabu pieejamību, tiek veidots plānojumu reprezentējošs grafs, kur katrs grafa mezgls reprezentē vienu no istabām. Iegūtam grafam tiek atrasts mazākais pārklājošais koks (*minimum spanning tree*), lai noteiktu minimālu durvju skaitu un izvietojumu, kas nodrošinās iespēju aģentam iziet cauri visām ģenerētām istabām. Turpmāk ar nelielu varbūtību tiek ģenerētas vel dažas durvis, ar priekšroku lielākām istabām, lai nodrošinātu lielāku variāciju..

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| 2.att. durvju izvietošanas algoritma darbības princips pa soļiem | | | |

|  |  |
| --- | --- |
|  |  |
| 3.att. Procedurāli ģenerēto vižu piemēri ar durvīm | |

* 1. Mēbeļu un šķēršļu ģenerēšanā

Reālajā dzīvē dzīvokļos un mājās vienmēr ir mēbeles. Mēbeles kalpo dažādiem nolūkiem, bet tīrīšanas robotiem tie pirmkārt ir šķērslī, kas ir jāņem vērā plānojot apiešanas maršrutu. Lai pilnās apiešanas algoritmu testēšanas rezultāti būtu pilnvērtīgi, arī simulācijā ir jāiekļauj mēbeļu ģenerēšana. Lai izvietot mēbeles uzticami, ir jāņem vērā vairāki noteikumi. Pirmkārt, katrā istabā var atrasties tikai konkrēta tipa mēbelēs, piemēram guļamistabā obligāti jābūt gultai, bet virtuvē nevajadzētu būt tualetei. Istabas tips tiek noteikts vel pirmajā solī, skeleta ģenerēšanas laikā. “Sēklas”, kas tika nejauši ģenerētas ir objekti, kas satur informāciju arī par istabas tipu. Ģenerējot mēbeles algoritms pirmkārt atsaucās uz to informāciju, lai veidotu zināšanu bāzi: kādām mēbelēm obligāti jābūt istabā, kādas mēbeles var būt istabā un kādas mēbeles ģenerēt nedrīkst noteiktajā istabā.

Otrkārt, ir jāņem vērā arī savstarpējo novietojumu mēbelēm. Piemēram mēbeles nedrīkst stāvēt durvīm priekšā, jo tādā gadījuma aģents nevarēs izbraukt tām cauri. Turpmāk, mēbeles ir jāizvieto tā, lai nerādītu pārāk šauras vietas arī istabas iekšā, pretējā gadījumā testēšanas vidē būs nesasniedzami apgabali, un testa rezultātus nevarēs uzskatīt par pilnvērtīgiem. Turklāt jāņem vērā arī mēbeļu izvietošanas loģiskumu. Piemēram ja dzīvojamā istabā ir televizors, tad dīvānam jābūt orientētam pret televizoru.