Rīgas 64. vidusskola

**Mākslīgā intelekta problēmu risināšana,**

**izmantojot klasisko meklēšanu**

Zinātniski pētnieciskais darbs datorzinātnēs

**Darba autori**:

Rīgas 64. vidusskolas 12. klases skolēni

Kristiāns Magons, Daniels Muļukins

**Darba vadītājs**:

Vadošais arhitekts, SAP SE

Krists Magons

**Darba konsultants**:

Programmēšanas skolotājs, Rīgas 64. vidusskola

Eduards Bukovskis

Rīga, 2025

**Ievads**

Pēdējo gadu laikā ir strauji attīstījušās datortehnoloģijas [1], kas līdzās sekmējušas arī mākslīgā intelekta attīstību, jo radās iespējas efektīvāk glabāt daudz datus mākoņos [2][3] un jaudīgās videokartes [4] ļāva ātrāk veikt skaitļošanu un apstrādāt vairāk datus. Šīs iespējas ļāva realizēt tādas tehnoloģijas kā TensorFlow, kas vienlaikus paralēli izmanto vairākus procesorus, kas būtiski paātrina matemātiskus aprēķinus [5]. Šādas skaitļošanas platformas brīvi pieejamas, pateicoties daudzkodolu videokaršu attīstībai. Pagaidām mākslīgais intelekts nav stingri definēts, jo atšķiras izpratnes par to. Vēsturiski ir nonākts pie četrām pamata definīcijām, kā izprast mākslīgā intelekta jēdzienu – domāšana vai rīkošanās cilvēciski, domāšana vai rīkošanās racionāli [6]. Mūsu pieeja būs racionāli rīkojošs, inteliģents aģents. Mēs izstrādāsim un izpētīsim putekļsūcēja modeli, kura mērķis, kā jau rīkojoties racionāli, ir visefektīvāk nokļūt no sākuma stāvokļa līdz rezultāta stāvoklim, proti, nonākt stāvoklī, kurā visas istabas ir tīras.

Mākslīgā intelekta rīki cilvēkiem jau ir kļuvuši par būtisku palīgu ikdienā, taču prognozes par tā tālāko nākotni ir dažādas. Parasti nepilnības rodas, kad spriedumus mēģina veikt, nevienojoties par mākslīgā intelekta definīciju, jo, kā jau tika minēts, mākslīgā intelekta sfēra ir ļoti plaša ar dažādām definīcijām. Sākot no šaha spēles risinājumiem līdz robotam, kas domā kā cilvēks. Mākslīgā intelekta problemātika ir pētīta jau vairākus desmitus gadus, un šī sfēra ir piedzīvojusi gan lielus uzplaukumus, gan arī kritumus [7]. Tāpēc mēs vēlamies apskatīt problēmjautājumu pamatus, lai gūtu skaidrību par mākslīgā intelekta tehnoloģiju iespējām un gūtu priekšstatu par tā limitācijām.

**Pētījuma mērķi**:

1. Iepazīties ar zema līmeņa programmēšanu un datu struktūrām.
2. Konstruēt racionālu aģentu, putekļsūcēja prototipu, kas darbojas mākslīgā, stingri definētā vidē un izmanto tā sensorus, lai veiktu stāvokļa pārejas, maksimizējot rezultātu (īsākais ceļš no sākuma stāvokļa līdz rezultāta stāvoklim, kad visas istabas ir iztīrītas).
3. Realizēt aģenta stāvokļu pārejas programmu, kā klasisku grafa meklēšanas algoritmu un novērtēt aģenta efektivitāti atkarībā no tā darba vides sarežģītības (istabu skaita).

**Darba uzdevumi**:

1. Valodā C++ realizēt zema līmeņa datu struktūras un pamatalgoritmus (saistītais saraksts, prioritāšu rinda, steks, grafs), kā arī iepazīties ar teksta failu apstrādi;
2. Iepazīties ar programmatūras versiju pārvaldības sistēmu GIT;
3. Izmantojot realizētās datu struktūras un OOP (Objektorientētā programmēšana) paradigmas, realizēt aģenta un vides implementāciju un darbības simulāciju;
4. Realizēt aģenta stāvokļa pārejas funkciju, izmantojot A\* vai citus meklēšanas algoritmus;
5. Novērtēt aģenta veiktspēju atkarībā no vides iestatījumiem;
6. Noskaidrot potenciālās vajadzības pēc atmiņas daudzuma un izšķirtspējām;
7. Izvērtēt mākslīgā intelekta potenciālās spējas nākotnē.

**Hipotēze**:

* Atrast teorētisku risinājumu mākslīgā intelekta problēmai nenozīmē, ka to ir viegli realizēt praktiski.

**Darbā izmantotās metodes**:

* Programma tiks vairākkārt palaista, mainot vides iestatījumus, aģenta īpašības un tā skaitu. Iegūtie rezultāti tiks analizēti (laiks, atmiņas patēriņš, utt.)

**Teorija**

1. Ieskats MI
   1. Kas ir MI? \**cilvēciska rīcība, cilvēciska domāšana, racionāla rīcība, racionāla domāšana*\* \**Alan Turing*\* \**MI ir plašs zinātņu lauks*\* \**Tipiskās MI problēmas un risinājumi*\*
   2. Izcelsme
   3. MI uzplaukums
2. Algoritmu sarežģītība
   1. Kas ir algoritmu sarežģītība?
   2. Kāpēc tas ir svarīgi?
   3. Kā tas saistās ar MI?
3. A\* algoritms
   1. Kas ir A\* algoritms?
   2. Kur izmanto?
4. Mūsu implementācija
   1. Sākumi. Paņēmienu izvēle, datu struktūru izmantošana
   2. Problēmas
   3. Kopskats

**Atsauces**

1. Moore, G. E. Cramming more components onto integrated circuits (1965). Pieejams tiešsaistē [šeit](http://cva.stanford.edu/classes/cs99s/papers/moore-crammingmorecomponents.pdf) [23/09/2024]
2. Russel, Stuart Jonathan; Norvig, Peter. (2009). Artificial Intelligence: A Modern Approach (3rd ed.), 27 lpp.
3. [Avots](https://www.oracle.com/artificial-intelligence/ai-cloud-computing/#:~:text=Cloud%20computing%20provides%20the%20necessary,data%20analytics%2C%20and%20improving%20cybersecurity) “What is Cloud Computing?”, “AI and Cloud Computing Explained”
4. Chellapilla, Kumar; Puri, Sidd; Simard, Patrice. (2006). [Avots](https://inria.hal.science/inria-00112631/document)
5. TensorFlow dokumentācija: [Avots](https://www.tensorflow.org/guide/migrate/mirrored_strategy)
6. Russel, Stuart Jonathan; Norvig, Peter. (2009). Artificial Intelligence: A Modern Approach (3rd ed.), 2 lpp.