



**KAUNO TECHNOLOGIJOS UNIVERSITETAS**

**INFORMATIKOS FAKULTETAS**

**INTELEKTIKOS PAGRINDAI**

## **Projektinis darbas**

*Ataskaita*

### **Atliko**

IFF-6/10 grupės studentai

Margiris Burakauskas

Ramūnas Purtokas

Marius Teleiša

### **Priėmė**

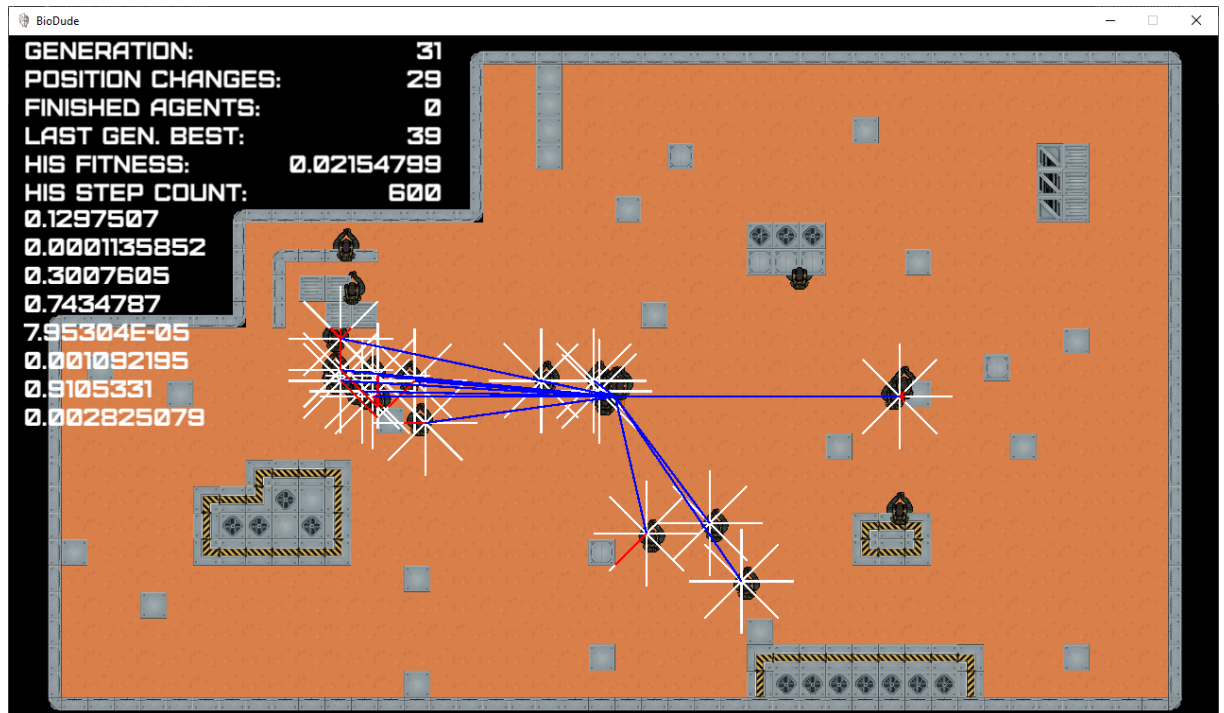
lekt. Audrius Nečiūnas

**Kaunas, 2019**

## Turinys

1. Žaidimas „BioDude“ .....	3
2. Duomenų surinkimas.....	5
3. Dimensijų sumažinimas .....	5
4. Duomenų normalizacija .....	5
5. Atsitiktinių komandų metodas, treniruojamas genetiniu algoritmu .....	5
6. Atsitiktinių komandų metodas, treniruojamas „Laimėtojas ima viską“ („Winner Takes All“) algoritmu .....	7
7. Neuroninio tinklo metodas, treniruojamas genetiniu algoritmu.....	8
8. Neuroninio tinklo metodas, treniruojamas „Laimėtojas ima viską“ („Winner Takes All“) algoritmu.....	10
9. Santrauka .....	11
10. Šaltiniai.....	11
11. Paveikslėliai.....	11

# 1. Žaidimas „BioDude“



1 pav. Modifikuoto žaidimo „BioDude“ vaizdas

„Unity“ žaidimų varikliu sukurtas 2D kompiuterinis žaidimas, kuriame galima vaikščioti, šaudyti, yra kliūčių. Žaidimas buvo pritaikytas mašininio mokymosi metodams ir neuroninio tinklo apmokymui atlikus tokius veiksmus:

- Sukurta grafinės sąsajos lentelė informacijai pateikti apie dirbtinio intelekto mokymąsi.
- Išimti grafinės sąsajos žaidimo informaciniai langai.
- Pašalinti priešai.
- Parašytas API veikėjo valdymui ir duomenų gavimui iš žaidimo.
- Sukurtas specialus žemėlapis mokymuisi.
- Modifikuotas mirties algoritmas: agentui atsitrenkus į sieną jis miršta ir nebegali atlikti tolimesniu veiksmu.

Sąsajos kodas:

```

float[] look()
{
    // Bit shift the index of the layer (17) to get a bit mask
    int layerMaskWall = 1 << 17;
    float[] vision = new float[9];

    Vector3 linePosStaticStart = transform.position;
    Vector3 linePosVisionStart = transform.position;
    Vector3 linePosVisionEnd = transform.position;
    linePosStaticStart.z = staticLinesZ;
    linePosVisionStart.z = visionLinesZ;

    for (int i = 0; i < 8; i++)
    {
        int angle = 45 * i;

        Vector3 direction = Quaternion.Euler(0, 0, angle) * Vector3.up;
        direction = new Vector2(direction.x, direction.y);

        RaycastHit2D hit = Physics2D.Raycast(transform.position, direction, visionDistance, layerMaskWall);

        Vector3 linePosStaticEnd = transform.position + direction * visionDistance;

        if (hit.collider != null)
            linePosVisionEnd = hit.point;

        linePosStaticEnd.z = staticLinesZ;
        linePosVisionEnd.z = visionLinesZ;
        staticLines[i].SetPositions(new[] {linePosStaticStart, linePosStaticEnd});
        visionLines[i].SetPositions(new[] {linePosVisionStart, linePosVisionEnd});

        //old version - gives 0 value if no hit and jumps to max value if hit at max range
        //vision[i] = visionDistance - hit.distance;

        //new version - returns the length of the raycast
        vision[i] = hit.distance == 0f ? visionDistance : hit.distance;
        //Debug.Log(i + ": " + vision[i]);
    }

    //vision[8] = Vector3.Angle(transform.position, posFinish);
    //vision[8] = Vector3.SignedAngle(transform.position, posFinish, Vector3.right);
    vision[8] = AngleInDeg(transform.position, posFinish);
    visionLines[8].SetPositions(new[] {linePosVisionStart, new Vector3(posFinish.x, posFinish.y, visionLinesZ)});

    return vision;
}

```

2 pav. Įvesties duomenų gavimo kodas

Atlikti testai:

- Mutacijos greičio įtaka.
  - 0,01
  - 0,05
  - 0,1
- Agentų skaičiaus įtaka.
  - 10
  - 50
  - 100
- Išvesčių kiekio įtaka (testas vykdytas tik 7 su skyriuje aprašytu metodu).
  - 4
  - 8
- Įvesčių normalizavimo įtaka (testas vykdytas tik 7 su skyriuje aprašytu metodu).
  - Normalizuota
  - Nenormalizuota

Numatytieji parametrai:

- Mutacijos greitis – 0,05.
- Agentų skaičius – 100.
- Išvesčių kiekis – 8.
- Įvesčių normalizavimas – nenormalizuotos.

## 2. Duomenų surinkimas

Duomenų šaltinis: žaidimas “BioDude”.

„Unity“ žaidime yra aprašytas API, atskiriantis žaidimo ir mašininio mokymosi programinius modulius, apribojantis sąsają tarp jų bet suteikiantis minimalius bei paruoštus duomenis.

1 lentelė. Įvesčių parametrai

Įvestis	Reikšmių diapazonas	Normalizuotos reikšmės *
Atstumai iki sienų visomis 8 kryptimis arba matymo atstumas	Nuo 0 iki 2 (matymo atstumas)	Nuo 0 iki 1
Tikslo kryptis laipsniais	Nuo -180 iki 180	Nuo -1 iki 1

\* – taikoma ne visuose metoduose.

2 lentelė. Išvesčių parametrai

Išvestys	Reikšmių diapazonas	Normalizuotos reikšmės
4 – 8 judėjimo kryptys judėjimo priklausomai nuo realizuojamo metodo	Nuo 0 iki 1	-

## 3. Dimensijų sumažinimas

Iš visų įmanomų krypčių (360 laipsnių) atrenkamos 8 kryptys kas 45 laipsniai bei vienas kampas nuo agento iki tikslo.

## 4. Duomenų normalizacija

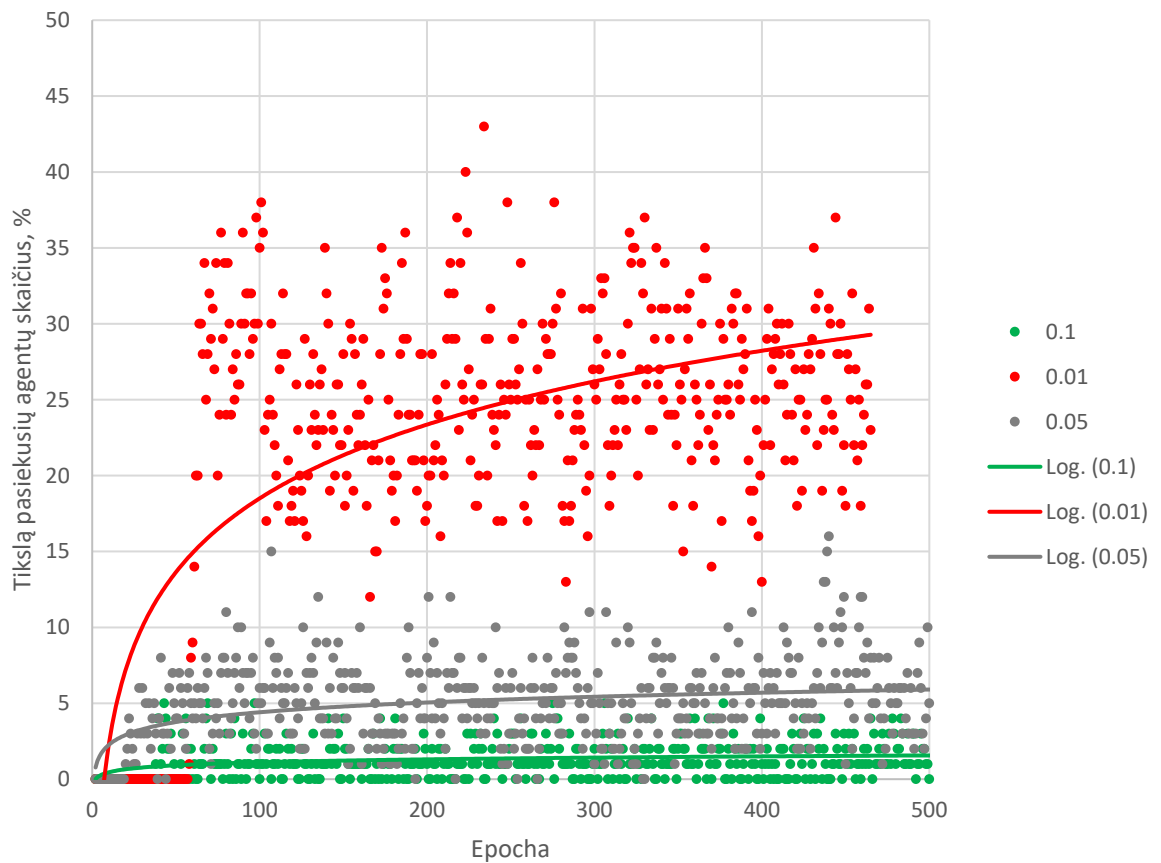
Viename metode (neuroninio tinklo su genetiniu algoritmu) buvo panaudota įvesčių normalizacija, tačiau atsižvelgus į gautus rezultatus nuspręsta jos nenaudoti.

## 5. Atsitiktinių komandų metodas, treniruojamas genetiniu algoritmu

Atsitiktinai sugeneruojama nustatyto ilgio komandų seka (į kairę, į dešinę, pirmyn, atgal), kurią agentas įvykdo žemėlapyje. Visiems agentams pabaigus vykdyti savo veiksmų sekas arba mirus, apskaičiuojamas kiekvieno agento tikslumas pagal likusi atstumą nuo jo iki tikslo (arba, jei agentas pasiekė tikslą, pagal atliktų veiksmų kiekį) ir genetiniu būdu (kryžminant du tėvus ir taip sukuriant vaiką bei mutuojuant sukurtus vaikus) sukuriami kiti agentų karta, kuri tęsia mokymąsi. Tėvai kryžminimui parenkami atsitiktinai, kiekvieno tėvo tikimybė būti parinktam yra proporcinga jo tikslumui.

Tikslą pasiekusių agentų skaičius, %

Mutacijos greičio įtaka agentų mokymosi greičiui

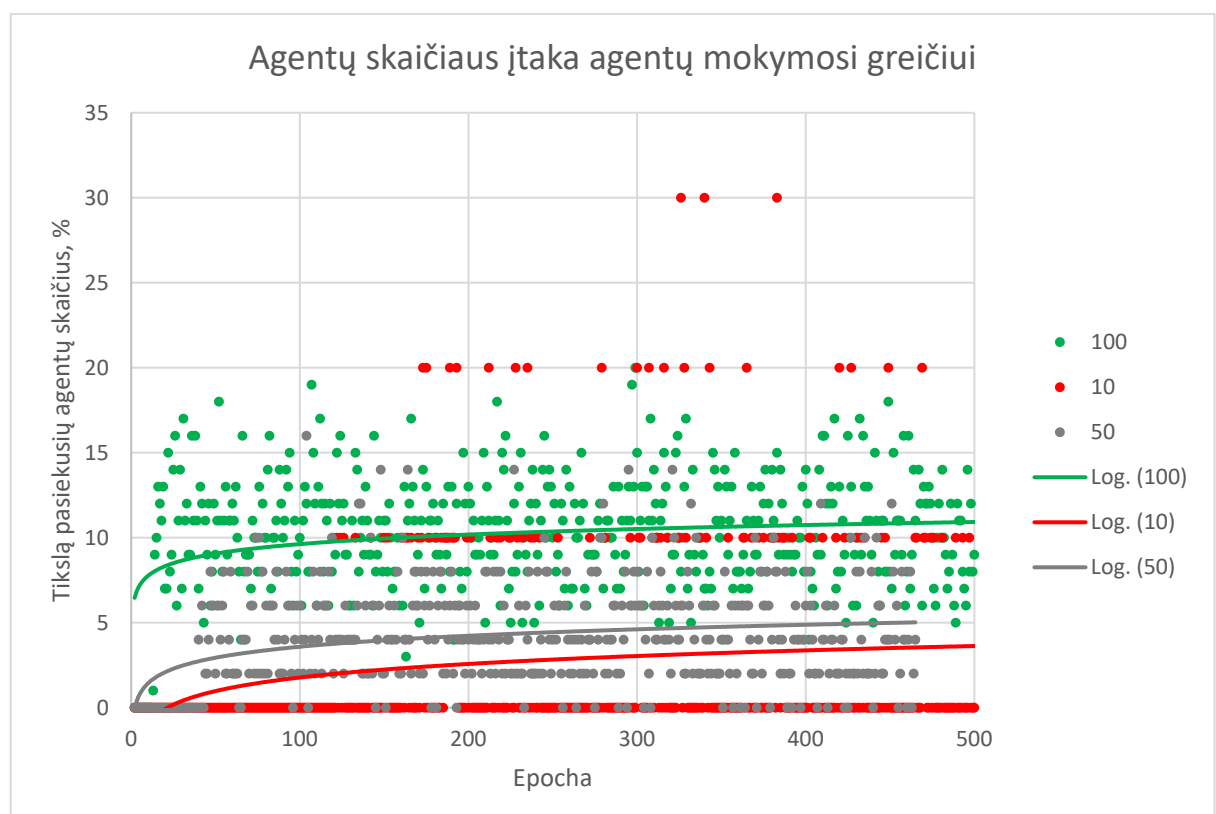
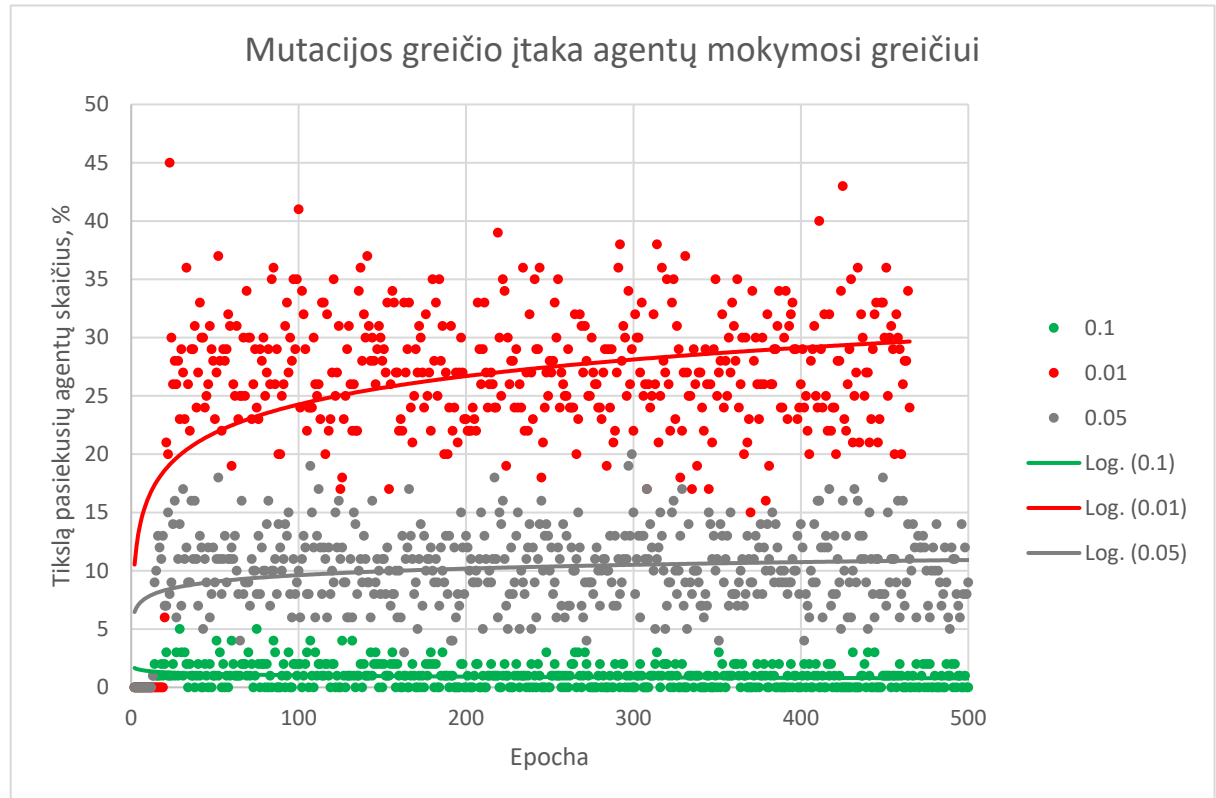


Agentų skaičiaus įtaka agentų mokymosi greičiui



## 6. Atsitiktinių komandų metodas, treniruojamas „Laimėtojas ima viską“ („Winner Takes All“) algoritmu

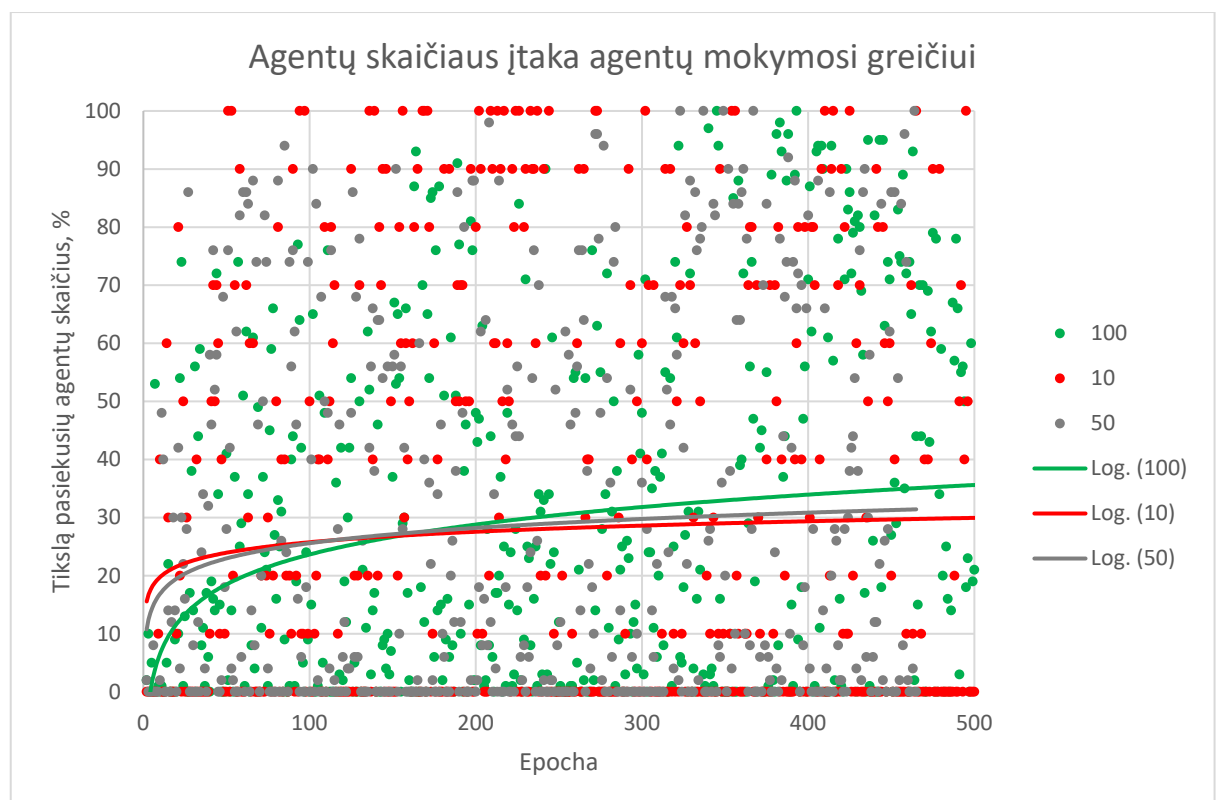
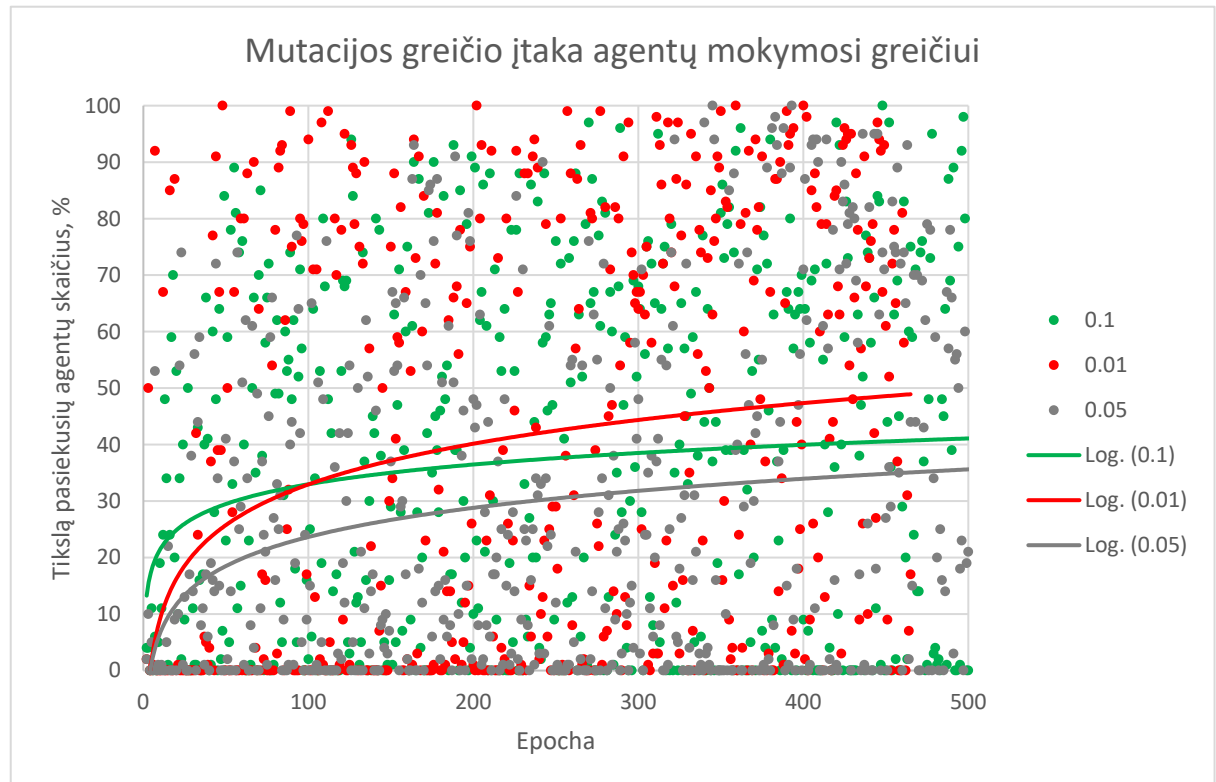
Atsitiktinai sugeneruojama nustatyto ilgio komandų seka (į kairę, į dešinę, pirmyn, atgal), kurią agentas įvykdo žemėlapyje. Visiems agentams pabaigus vykdyti savo veiksmų sekas arba mirus, surandamas geriausias agentas ir perkeliamas į kitą kartą. Likę agentai tampa jo klonais ir yra mutuojami. Po to agentai tęsia mokymąsi.



## 7. Neuroninio tinklo metodas, treniruojamas genetiniu algoritmu

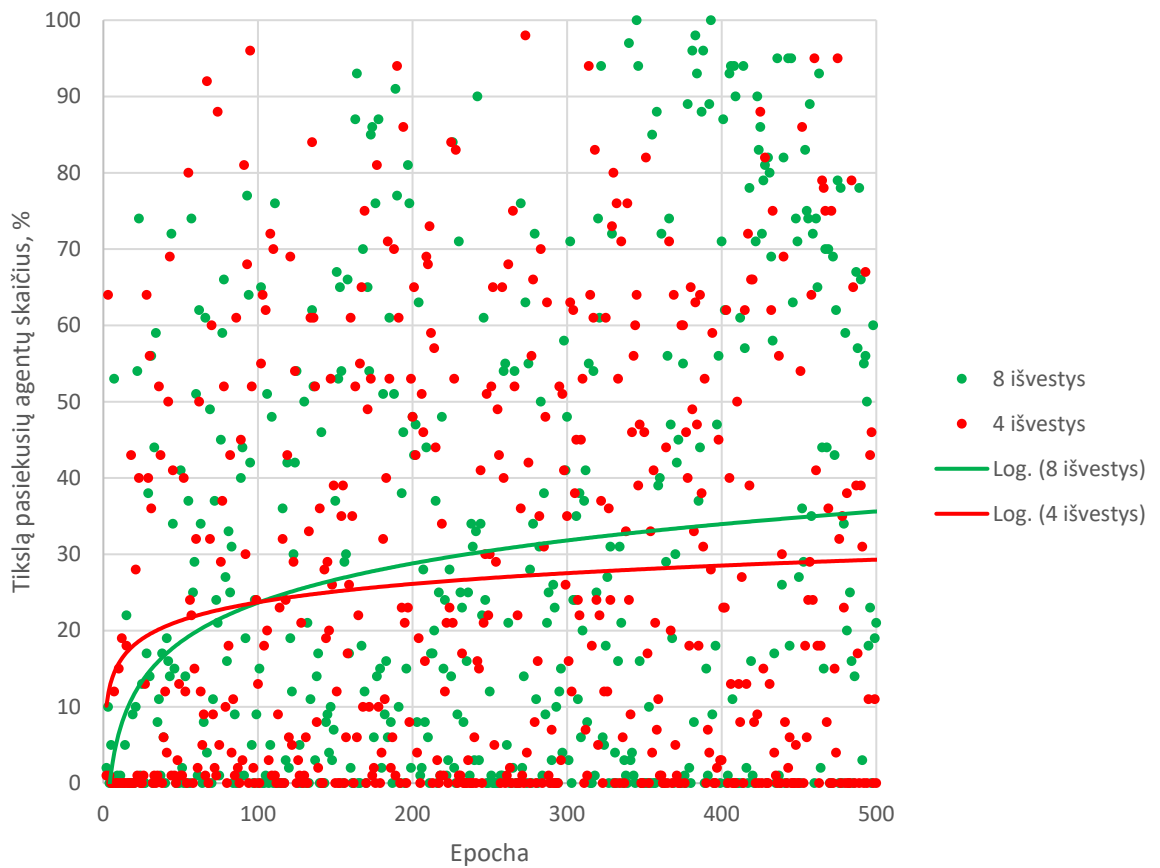
Neuroninio tinklo struktūra – 9 įvesties neuronai, 36 paslėpti neuronai bei 8 išvesties neuronai (8 kryptys). Tinklas pilnai sujungtas („fully connected“).

Atsitiktinai sugeneruojami jungčių svoriai, pagal kuriuos kiekvieno agento neuroninis tinklas kiekvieno kadro metu sprendžia veiksmą – eiti į vieną iš 8 (arba 4 tam tikruose testuose) krypčių – pagal gautus duomenis. Visiems agentams pabaigus vykdyti savo veiksmų sekas arba mirus, panaudojamas genetinis agentų tobulinimas, aprašytas 6 skyriuje.

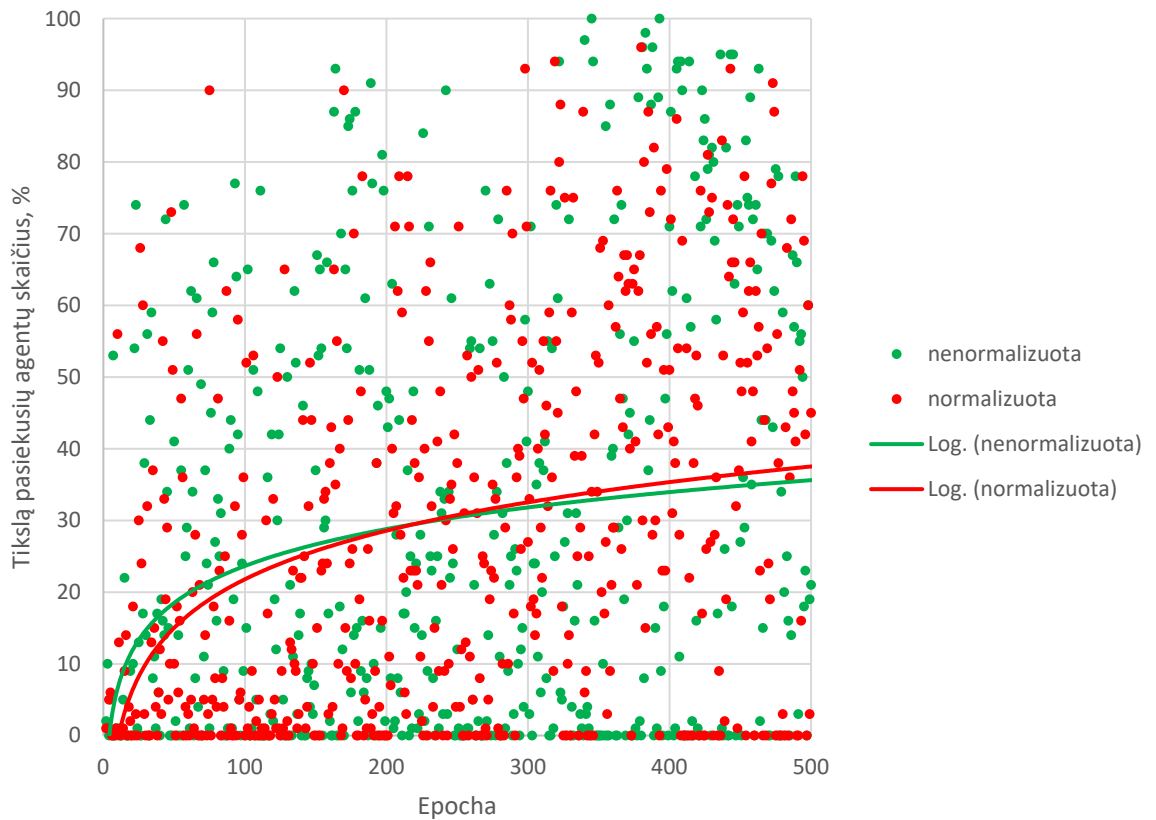




Išvesčių kiekio įtaka agentų mokymosi greičiui



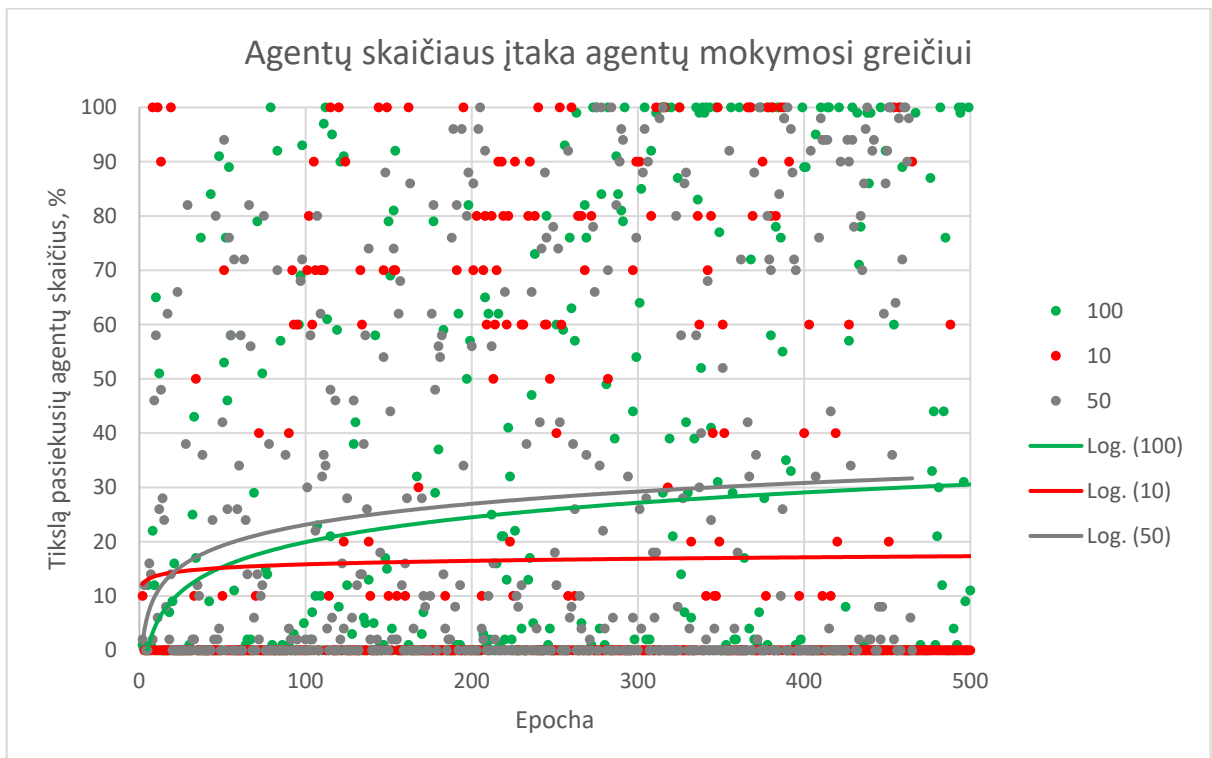
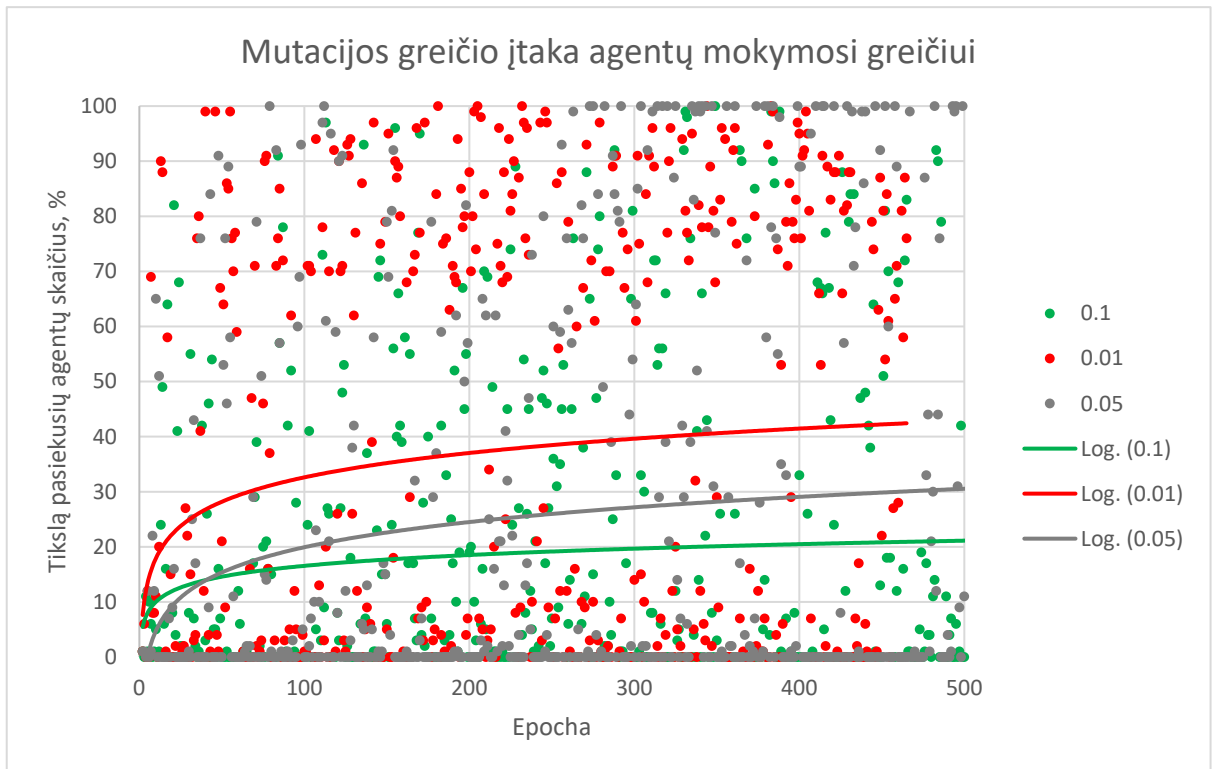
Išvesčių normalizavimo įtaka agentų mokymosi greičiui



## 8. Neuroninio tinklo metodas, treniruojamas „Laimėtojas ima viską“ („Winner Takes All“) algoritmu

Neuroninio tinklo struktūra tokia pat, kaip ir 8 skyriuje.

Atsitiktinai sugeneruojami jungčių svoriai, pagal kuriuos kiekvieno agento neuroninis tinklas kiekvieno kadro metu sprendžia veiksmą – eiti į vieną iš 8 (arba 4 tam tikruose testuose) krypčių – pagal gautus duomenis. Visiems agentams pabaigus vykdyti savo veiksmų sekas arba mirus, surandamas geriausias agentas ir perkeliamas į kitą kartą. Likę agentai tampa jo klonais ir yra mutuojami. Po to agentai tęsia mokymąsi.



## 9. Santrauka

Pagal gautus rezultatus galima spręsti, jog sukurtoje sistemoje panaudoti mašininio mokymosi metodai yra efektyvūs ir padeda pasiekti užsibrėžtą kelio radimo tikslą. Geriausiai sekėsi 7-ame skyriuje aprašytu metodu besimokusiems agentams. Su 0,01 mutacijos greičiu, 100 agentų, 8 išvestimis bei nenormalizuotais įvesčių duomenimis per 500 epochų pasiektas beveik 50% tikslumas.

## 10. Šaltiniai

Sukurtos sistemos „git“ repozitorija: <https://github.com/Ramunas9/gamedevexe>.

- <https://www.youtube.com/channel/UC0e3QhIYukixgh5VVpKHH9Q>
- [A Neural Network in 11 lines of Python \(Part 1\) - i am trask](#)
- [Code-Bullet/SnakeFusion: Using the genetic algorithm and neural networks I trained up 5 snakes who will then fuse to become the ultimate snake, this is how I did it](#)
- [Artificial Neural Network Unsupervised Learning](#)

## 11. Paveikslėliai

1 pav. Modifikuoto žaidimo „BioDude“ vaizdas .....	3
2 pav. Įvesties duomenų gavimo kodas .....	4