

## ხელოვნური ინტელექტის არგუმენტაციის მექანიზმის გამოყენება ვიდეოთამაშებში

გიორგი აბელაშვილი, ზურაბ ბოსიკაშვილი  
საქართველოს ტექნიკური უნივერსიტეტი

### რეზიუმე

ვიდეო თამაშებში არსებობს მრავალი ამოცანა, რომელთა მთავარი მოქმედი სუბიექტებია აგენტთა გაერთიანებები, ჯგუფები. ისინი იღებენ როგორც ინდივიდუალურ, ასევე კოლექტიურ გადაწყვეტილებებს. კოლექტიური გადაწყვეტილებები დამოკიდებულია ინდივიდუალურზე ე.წ. არგუმენტაციის მექანიზმით. აგენტები ჯგუფში ყოველთვის ერთნაირად არ მსჯელობენ და მოქმედებენ, მათი მიზანია მიაღწიონ კონსენსუსს, რომელიც საბოლოოდ ამ კოლექტივის გადაწყვეტილება იქნება. აგენტთა ჯგუფის ერთიანი გადაწყვეტილება, კონსენსუსის მიღწევის ერთ-ერთი ცნობილი მეთოდია არგუმენტაციის მექანიზმი. სტატიაში განვიხილავთ არგუმენტაციის მექანიზმს კონკრეტულ მაგალითზე - ჭიანჭველების ამოცანის მოდელზე.

**საკანძო სიტყვები:** ინტელექტუალური აგენტი. არგუმენტის შეტევა. VAF. პათერნი.

### 1. შესავალი: ვიდეოთამაშებში აგენტების გადაწყვეტილებები

ნებისმიერ ვიდეოთამაშში, სადაც მოცემულია მულტიაგენტური ტექნოლოგია, ძალიან მნიშვნელოვანია აგენტთა გადაწყვეტილების გამომტანი მექანიზმი. სტანდარტულად, აგენტს განვიხილავთ როგორც ინდივიდუალურს, შესაბამისად, მისი ქცევა უნდა იყოს ინდივიდუალური გადაწყვეტილებების გამომტანი სისტემის ქვემდგომი. თუმცა არის შემთხვევები, როდესაც აგენტები გაერთიანებულია კოლექტივებად, ჯგუფებად. თუ გადაწყვეტილების გამოსავალი წერტილი არის ამ აგენტთა ჯგუფის მიერ მიღებული საერთო გადაწყვეტილება, მაშინ მნიშვნელოვანია აგენტები შეთანხმდნენ გარკვეულ კონსენსუსზე, რომელიც ყველაზე მეტად ახლოს მდგომია სასურველ გამომავალ შედეგთან. ბუნებრივია, ჩნდება კითხვა, თუ ყველა აგენტი დამოუკიდებლად მოქმედებს, როგორ შეიძლება ისინი რამეზე შეთანხმდნენ, ან რის მიხედვით მიანიჭონ ამა თუ იმ გადაწყვეტილებას პრიორიტეტი? ასეთ შემთხვევაში, არსებობს არგუმენტაციის მექანიზმი. ამ მექანიზმის საშუალებით აგენტები გამორიცხავენ და ადასტურებენ ერთმანეთის ლოგიკურ არგუმენტებს და ფაქტებს. საბოლოო შედეგი კი არის კოლექტივის მიერ მიღებული საერთო გადაწყვეტილება. განვიხილოთ არგუმენტაციის მექანიზმი ჭიანჭველების ამოცანის მაგალითზე.

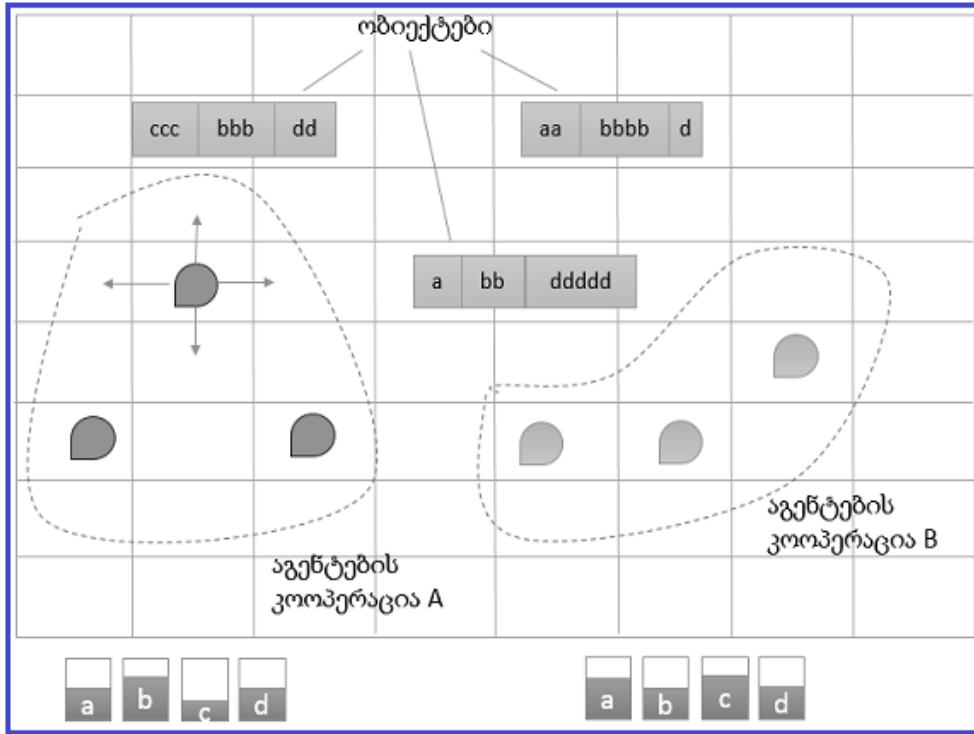
### 2. ჭიანჭველების ამოცანა

შემდგომში ყველგან ვიგულისხმებთ, რომ პროგრამული აგენტები დელიბერაციულია და მხოლოდ სიმბოლების სამყაროში მოქმედებს.

ჩამოვყალიბოთ ჭიანჭველა-აგენტების ამოცანა. ჭიანჭველა-აგენტები და ობიექტები ორ განზომილებიან, უჯრებად დაყოფილ, შემოფარგულ სიბრტყეზე განთავსებული და შეუძლია ამ სიბრტყეში მოძრაობა მხოლოდ ვერტიკალური და ჰორიზონტალური მიმართულებებით (ნახ.3).

აგენტები ასევე შეიძლება გარკვეულ კოორდინატაში იყოს გაერთიანებული. სიბრტყეზე შემთხვევითად ჩნდება ობიექტები (ფოთლები), რომლებიც წარმოადგენს შეზღუდული ანბანის შედგენილ მიმდევრობათა მიმდევრობას. აგენტს შეუძლია შემდეგი მოქმედებების განხორციელება:

- შეზღუდული ზომის ფოთლის გადაადგილება;
- ფოთლის დაჭრა იმ სიმბოლოზე რომელსაც იცნობს;



ნახ.3. ჭიანჭველების ამოცანის გრაფიკული პროტოტიპი

- გამოყოფილი სიმბოლოს შენახვა საცავში;
- თამაშის მიზანია გარკვეულ დროში კოლოპერაციაში მყოფმა ჭიანჭველმა - აგენტებმა, რაც შეიძლება მეტი სიმბოლოები დააგროვოს. აგენტები თავდაპირველად შემთხვევით პოზიციებში თავსდება და შეუძლია ერთდროულად მოძრაობა. ასევე კოლოპერაციაში მყოფმა აგენტებმა შეიძლება მოქმედებები ერთდროულად შეასრულოთ, მაგალითად, სამმა აგენტმა გადაიტანოს დიდი ზომის ფოთოლი.

### 3. არგუმენტაციის მექანიზმი

არგუმენტაციის პათერნი არის მრავალჯერ გამოყენებადი პრობლემის გადაჭრის ხერხი არგუმენტაციის მექანიზმში. პათერნები წარმოდგენს შემდეგი სახის აღწერებს:

<პათერნი> ::= <<თვისება><მნიშვნელობა>?<ვალიდურობა>>\*

<<მიმართება><მნიშვნელობა>?<ვალიდურობა>>\*

<მიმართება> ::= <ISA მიმართება>|<კომპოზიციის მიმართება> | <ავრეგირების მიმართება>

<თვისება> ::= თვისების აღმნიშვნელი ტერმი

<მნიშვნელობა> ::=  $R^n$  | დისკრეტული სიმრავლე

<ვალიდურობა> ::= {true,false} | [0,1]C R

*<კომპოზიციის მიმართება> ::= მთელი-ნაწილი მიმართება*

*<აგრეგირების მიმართება> ::= ასოცირების მიმართება*

კონკრეტულ მაგალითში, ვთქვათ, ნებისმიერ ჭიანჭველას კოლონიაში გააჩნია თავისი როლი: *მზვერავი* - ჭიანჭველა, რომელიც ეძებს ვარგის ზომის ფოთლებს.

*ფოთლების მჭრელი* - ჭიანჭველები, რომლებსაც აქვთ საწყისი ცოდნა სიმბოლოებზე, ანუ ისინი ცნობენ მხოლოდ მარტივ სიმბოლოებს, თუ ფოთოლი შეიცავს ამ სიმბოლოს ეს ჭიანჭველები შესაბამისი ფოთლის ნაწილს მოაჭრიან.

*გადაშინდავები* - ჭიანჭველები, რომლებიც ეძებენ უკვე ფოთლებისგან ჩამოჭრილ სიმბოლოებს და მიაქვთ საცავში.

თუ ჭიანჭველების ამოცანას ცალკეული როლების მიხედვით განვიხილავთ, ამოცანა მარტივია, გამოსავალი შედეგი თვითოეული როლისათვის არის ცალსახა, ჭიანჭველა აკეთებს ინდივიდუალურ დასკვნებს და იქცევა ინდივიდუალურად.

განვიხილოთ შემთხვევა, როდესაც ჭიანჭველები უნდა გაერთიანდნენ ჯგუფებად და ისე იმოქმედონ, მაგალითად, მძებნელმა უნდა აიყვანოს ჯგუფში კარგი ფოთლის მჭრელი, შესაბამისად ფოთლის მჭრელმაც შეიძლება დაიმატოს გუნდში კარგი გადაშინდავი, ან ფოთლის მძებნელი. თუ ჩვენ ნებისმიერი ახალი გუნდის წევრის დამატებას კოლექტიურ გადაწყვეტილებად გავიტანთ, ამ შემთხვევაში უკვე საქმე გვაქვს არგუმენტაციის ამოცანასთან, გუნდში არსებული ჭიანჭველები უნდა შეთანხმდნენ დაიმატონ თუ არა ახალი წევრი მისი მონაცემების მიხედვით გუნდში. არსებობს რამდენიმე ტიპის არგუმენტაციის მექანიზმი, განვიხილოთ თვითოეული მათგანი [4].

### 3.1. კლასიკური არგუმენტაციის მექანიზმი

მოდელი, რომელიც წარმოდგენილია წყვილით  $AF = (AR, attacks)$ , სადაც  $AR$  არის არგუმენტების სია, ხოლო  $attacks$ ,  $AR$  - ის ბინარული მიმართება.  $attack(A, B)$  ნიშნავს, რომ არგუმენტი  $A$  უტევს არგუმენტ  $B$ -ს.  $S$  არგუმენტების სია უტევს არგუმენტ  $B$ -ს, თუ  $B$  არის შეტევის ქვეშ  $S$ -ში. ამ მოდელის არგუმენტაციის არსებითი კითხვაა, როდის შეიძლება იყოს მიღებული  $A$  არგუმენტი სადაც  $A, A \in AR$ . არგუმენტი შეიძლება იყოს მიღებული იმ შემთხვევაში, თუ მასზე შემტევი ყველა არგუმენტი არის მიღებული [1,2].

### 3.2. მნიშვნელობაზე დამყარებული არგუმენტაციის მექანიზმი

არსებობს არგუმენტაციის მექანიზმი, რომლის პრინციპი დამყარებულია მნიშვნელობაზე - (VAF) Value-based Argumentation Framework. [4] ეს მოდელი ადგენს რომელი არგუმენტები არის მისაღები სხვა აგენტების არგუმენტების გათვალისწინებით. როგორც ვიცით, აგენტებს აქვთ განსხვავებული მიდგომა დავალებებისადმი და ამ შემთხვევაში ისინი თანამშრომლობენ და ცვლიან თავიანთ ლოკალურ გამოძავალ შედეგებს.

VAF - ს შეუძლია განაცალკევოს შეტევები ( $attacks$ ) წარმატებული შეტევებისგან, იმ შეტევებისგან, რომლებიც ამარცხებს სხვა არგუმენტებს. ისინი ითვალისწინებს იმ მნიშვნელობათა მიმდევრობას, რომლებიც ასოცირებულია ამ არგუმენტებთან [3].

VAF გამოისახება როგორც:

$$VAF = (AR, attacks, V, val, P),$$

სადაც

$AR, attacks$  არის არგუმენტაციის მოდელი;

$V$  - არაცარიელ მნიშვნელობათა სიმრავლე;

$val$  - ფუნქცია, რომელიც ასახავს ელემენტებს  $AR$ -დან  $V$  და  $P$ ,

$P$  - ყველა შესაძლო შემთხვევის სია.

$$\text{ყოველი } A \in AR, val(A) \in V.$$

### 3.3. აგენტების მიერ თვისობრივი არგუმენტების გენერაცია

თვისობრივი არგუმენტაციის გენერაცია [4] იწყება მაშინ, როცა ობიექტის თვისებების შეთავსების აგენტებმა თავიანთი საქმე დაამთავრეს. არგუმენტაციის დასაგენერირებლად გამოიყენება  $\varphi(g_i^p(o), g_i^p(t))$  ფუნქცია და ღარდება მათი თვისებების სიმრავლეები.

1-ელ ცხრილში მოყვანილია  $c$  და  $h$ -ის მნიშვნელობები. თუ მოხდა პატერნთან სრული თანხვედრა მაშინ, ყველაფერი გარკვეულია და მისი კონფიდენციალობის დონე არის უდავო და სხვა აგენტები ვერ შეეკამათებიან ამ არგუმენტს. მაშინ, როცა ხდება ნაწილობრივი შეთავსება და შეთავსების ფუნქცია დადებითია მაშინ კონფიდენციალობა გაურკვეველია და შესაძლებელია ამ არგუმენტაციის გაპროტესტება.

ცხრ.1

C	+(h)
უდავოა	თუ ობიექტის ყველა თვისება 0 თანხვედრაშია პატერნ t-ს თვისებებთან და შესაბამისად, მნიშვნელობებიც ემთხვევა და $\varphi(g^p(o), g^p(t)) \geq 0$
სადავოა	თუ ობიექტის ყველა თვისება 0 მსგავსია პატერნ t-ს თვისებებთან, ან / და შესაბამისად, მნიშვნელობებიც არ ემთხვევა და $(g^p(o), g^p(t)) \geq 0$
C	-(h)
სადავოა	სხვა შემთხვევაში $\varphi(g^p(o), g^p(t)) < 0$

### 3.4. აგენტების მიერ მიმართებითი არგუმენტების გენერაცია

მიმართებითი არგუმენტაციის პროცესი [4] იწყება მაშინ, თუ თვისობრივი არგუმენტი გენერირებულია და ის განიცდის შეტევას, ამასთან ყველა გამოსაცნობი ობიექტის ყველა ქვეობიექტი უნდა იყოს პატერნებთან შეთავსებული.

მე-2 ცხრილში მოყვანილია არგუმენტების კონფიდენციალობის და კონტრარგუმენტების წამოყენების პირობები. აქ არგუმენტაციის გამოსათვლელად გამოიყენება  $\varphi(g^r(o), g^r(t))$  ფუნქცია.

ცხრ.2

<b>უდავოა</b>	თუ ობიექტის ყველა თვისება 0 თანხვედრაშია პათერნ t-ს თვისებებთან და ყველა ქვე-ობიექტების მნიშვნელობაც თანხვედრაშია და შესაბამისად, თუ ყველა თვისების მნიშვნელობებიც ემთხვევა და $\varphi(g^r(o), g^r(t)) \geq 0$
<b>სადავოა</b>	თუ ობიექტის ყველა თვისება 0 თანხვედრაშია პათერნ t-ს თვისებებთან და ყველა ქვე-ობიექტების მნიშვნელობაც თანხვედრაშია და თუ ყველა თვისების მნიშვნელობებიც არ ემთხვევა და $\varphi(g^r(o), g^r(t)) \geq 0$
<b>C</b>	-(h)
<b>სადავოა</b>	სხვა შემთხვევაში $\varphi(g^r(o), g^r(t)) < 0$

**3.5. აგენტების მიერ სტრუქტურული არგუმენტაციის გენერაცია**

სტრუქტურული აგენტები არგუმენტების გენერაციას განიხილავს ცნებების ონტოლოგიური იერარქიის კონტექსტში [4], როცა მოცემული ობიექტი შეიძლება შეუთავსდეს იერარქიის ნებისმიერ ცნებას. რა თქმა უნდა არგუმენტების გენერირება იწყება ზევიდან ქვევით და ალგორითმი უნდა ჩავიდეს რაც შეიძლება ღრმად. თუ რომელიმე დონეზე ვერ მოხერხდა შეთავსება თვისობრივად და მიმართებრივად და არის გაურკვევლობა, მაშინ იერარქიის დაბალ საფეხურებზე მდგომი ცნებები უნდა მოინიშნოს როგორც არა გამოცნობადი, უდავო კონფიდენციალობით და მათთვის არგუმენტები აღარ უნდა დაგენერირდეს. თუ იერარქიის რომელიმე დონეზე გვაქვს განუზრვრელობის კონფიდენციალობა და კონტრ არგუმენტი არსებობს ამ დონეზე, მაშინ ეს ცნებაც უნდა მოინიშნოს როგორც არა გამოცნობადი, უდავო კონფიდენცი- ალობით და არგუმენტაცია უნდა გაგრძელდეს მშობელი ცნებისთვის (ცხრილი 3).

ცხრ.3

N	აგენტის დანიშნულება	კოდი	ტიპი	რაოდენობა
1	ობიექტების გამოყოფა	AO	Slave	m
2	თვისებების მნიშვნელობების გამოთვლა	AP	Slave	m+1
3	ობიექტის თვისებების შეთავსება პათერნის თვისებებთან	APM	Slave	m+n
4	ობიექტის მიმართების შეთავსება პათერნის მიმართებებთან	ARM	Slave	m+n
5	ობიექტის სტრუქტურის შეთავსება პათერნის	ASM	Slave	m+n
6	არგუმენტების გენერატორები	AA	Master	m+n
7	გადაწყვეტილების მიმღები	ADM	Guru	1

სადაც m-ქვეობიექტების კლასების რაოდენობაა, n-გამოსაცნობი კლასების რაოდენობა.

პირველ რიგში მუშაობას იწყებენ თვისებების გამოთვლელი და ობიექტების გამოყოფი აგენტები. თვისებების გამოთვლელი აგენტები მუშაობას იწყებს იმ შემთხვევაში, როგორც კი რაიმე ობიექტი არსებობს ან გამოიყოფა. თავის მხრივ, თვისების შეთავსების აგენტები მაშინ აქტიურდება, როდესაც თვისებების მნიშვნელობები გამოთვლილია. ამის შემდეგ აქტიურდება თვისებების მიხედვით არგუმენტაციის გენერატორი აგენტები.

თუ მოცემული კლასისთვის თვისებები გამოთვლილია და არსებობს კონტრ-არგუმენტი თვისებების მიხედვით ამ კლასის შესახებ, მაშინ აქტიურდება მიმართებათა შეთავსების აგენტები, რომელთა შემდეგაც იწყებენ მუშაობას მიმართებების მიხედვით არგუმენტაციის გენერატორი აგენტები. თუ მოცემული კლასისთვის კონტრ არგუმენტები მაინც ჩნდება მაშინ იწყებენ მუშაობას შესაბამისად სტრუქტურის შემთავსებელი და არგუმენტების გენერატორი აგენტები.

მას შემდეგ რაც თითოეული არგუმენტების გენერატორი აგენტი დააგენერირებს თავისი არგუმენტების სიმრავლეს, აგენტები ცვლიან თვიანთ არგუმენტებს. როცა ყველა აგენტისთვის ცნობილი გახდება სხვა აგენტების არგუმენტები მაშინ ისინი აგენერირებენ შეტევათა სიმრავლეს.

შეტევა ჩნდება მაშინ, როცა ობიექტის შეთავსების მნიშვნელობა პატერნთან უკეთესია ვიდრე სხვისა და  $h$ -ის მნიშვნელობა კონფლიქტურია, მაგალითად არგუმენტ  $(m_1(o, t_1), P, \text{certainty}, +)$  -ზე ხორციელდება შეტევა არგუმენტ  $(m_2(o, t_2), P, \text{certainty}, -)$  - ის მიერ თუ  $m_1$  და  $m_2$  ერთიდაიმავე ობიექტის სხვადასხვა პატერნთან შეთავსებაა, ან  $(m_2(o, t_1), R, \text{certainty}, -)$  -ს შემთხვევაში თუ გვაქვს ობიექტის ერთიდაიგივე პატერნთან შეთავსება.

როდესაც არგუმენტების და შეთავსების სიმრავლეები წარმოქმნილია, აგენტებმა უნდა გამოითვალონ უპირატესი გაფართოება, აუდენციის და კონფიდენციის მნიშვნელობიდან გამომდინარე. არგუმენტების სიმრავლე გლობალურად სუბიექტურად მისაღებია, თუ თითოეული ელემენტი ჩანს რომელიმე აგენტის სასურველ გაფართოების სიმრავლეში.

არგუმენტების სიმრავლე გლობალურად ობიექტურად მისაღებია, თუ თითოეული ელემენტი ჩანს ყველა აგენტის სასურველ გაფართოების სიმრავლეში. თუ არგუმენტი არის არც სუბიექტურად და არც ობიექტურად მისაღები მაშინ ის განიხილება როგორც უსაფუძვლო.

#### 4. ჭიანჭველების ამოცანა და ფორმალური არგუმენტაციის მოდელი

თუ ვიტყვით, რომ ნებისმიერი ჭიანჭველა დამოუკიდებლად მოქმედებს და მათ სურვილი აქვთ კოლექტივში გაერთიანების, მათ „კამათი“ მოუწევთ უკვე არსებულ გუნდის წევრებთან. თუ გუნდი თავდაპირველად არის მხოლოდ ერთი ჭიანჭველისგან შემდგარი, პირობითად შეიძლება მივიჩნიოთ, რომ ამ შემთხვევაში, არგუმენტაცია არ არის საჭირო რადგან აგენტი თავის თავს ვერ ეკამათება.

აღსანიშნავია ერთი ფაქტი, კოლექტივში შეიძლება გამოეყოს ერთი ჭიანჭველა, რომელიც შეიძლება მოვნიშნოთ როგორც პრიორიტეტული არგუმენტის მქონე, მის არგუმენტებს შეიძლება უფრო მეტი შეფასება ჰქონდეს და უფრო პრიორიტეტული იყოს, თუმცა ეს ვიდელთამაშის სპეციფიკაციიდან გამომდინარეობს, მოვიყვანოთ ფორმალური არგუმენტაციის მოდელი ჭიანჭველის ამოცანის მაგალითზე:

*ყველა ჭიანჭველისთვის*

*ყოველი ახალი ჭიანჭველა, რომელსაც კოლექტივში სურს გაერთიანება*

თვისებრივი არგუმენტების გენერაცია();

მიმართებითი არგუმენტების გენერაცია();

სტრუქტურული არგუმენტების გენერაცია();

ყველა ჭიანჭველისთვის

ყოველი ახალი ჭიანჭველა, რომელსაც კოლექტივში სურს გაერთიანება

თუ არსებობს მიმართება(ჭიანჭველა, პატერნი, არგუმენტი)

მაშინ ჩვენ გვაქვს დასტური.

თუ არა და არ გვაქვს დასტური.

ყველა დასტურისთვის

ყველა ჭიანჭველისთვის

თუ უდავობას შეუტია კოლონიის აგენტმა ჭიანჭველამ კონტრარგუმენტით

მაშინ უდავობა დადასტურებულია

თუ არა და

თუ უდავობას შეუტია კოლონიის აგენტმა ჭიანჭველამ და კონტრარგუმენტი არ არის

მაშინ უდავობა დაუდასტურებელია.

## 5. დასკვნა

ვიდეოთამაშებში ფართოდ არის გავრცელებული მრავალაგენტიანი სისტემები, აგენტთა გაერთიანებები, სადაც მნიშვნელოვანია ამ გაერთიანებათა საერთო გადაწყვეტილება, ხოლო საერთო გადაწყვეტილება ისეთი აგენტებისგან, რომლებიც ცალკე-ცალკე დამოუკიდებლად მოქმედებენ პირდაპირი გზით არ მიიღწევა, საჭიროა გარკვეული კამათი მათ შორის, რათა მიიღწეს კონსენსუსი და გამომავალი შედეგიც სასურველთან ახლოს იყოს, ამის მიღწევის ერთერთი ეფექტური მეთოდია არგუმენტაციის მექანიზმი, იგი დამოუკიდებელ აგენტთა მოქმედებებში ადგენს უპირატესს გამომავალ შედეგს პატერნების საშუალებით და საბოლოოდაც ამტკიცებს მას.

### ლიტერატურა:

1. Wang P. (2013). Non-Axiomatic Logic. A Model of Intelligent Reasoning. Copyright © 2013 by World Scientific Publishing Co. Pte. Ltd.
2. SmithR. (2012). Aristotle's logic. in *The Stanford Encyclopedia of Philosophy*, ed. Zalta, E. N. (Spring 2012 edn.).
3. MerrittD. (2004). Best Practices for Rule-Based Application. Development <http://msdn-microsoft.com/en-us/library/aa480020.aspx>

4. Archvadze G., Bosikashvili Z. (2015). Mathematic Model and Argumentation Theory with Multi Agent Based Architecture for Pattern Recognition. Transact. of Georgian Technical University. "ACS" No1(19), pp.187-196. [http://www.gtu.ge/Journals/mas/Referat/ASU-2015\(1\\_19\)/187\\_196.pdf](http://www.gtu.ge/Journals/mas/Referat/ASU-2015(1_19)/187_196.pdf)

## USING OF ARGUMENTATION MECHANISM OF ARTIFICIAL INTELLIGENCE IN VIDEO GAMES

Abelashvili Giorgi, Bosikasvili Zurab  
Georgian Technical University

### Summary

There are many problems in video games. Groups of agents have the main roles. They take a group and individual decisions. Group decisions are depended on individuals. Agents don't always think the same way, they wish to reach consensus, which will be their group's final decision. One of the known ways to reach consensus is the augmented mechanism. In this article, we will discuss argued mechanism as an example of Ants problem.

## ИСПОЛЬЗОВАНИЕ МЕХАНИЗМА АРГУМЕНТАЦИИ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА В ВИДЕО ИГРАХ

Абелашвили Г., Босикашвили З.  
Грузинский Технический Университет

### Резюме

В видео играх существуют много задач, в которых главным действующим субъектом являются объединение, группы агентов. Коллективные решения зависят от индивидуальных, от т.н. механизма аргументации. Агенты в группе не всегда рассуждают и действуют одинаково. Их задачей является достижение консенсуса, который является окончательным решением данной группы. Одним из методов достижения агентами группы единного решения, консенсуса является известный механизм аргументации. В статье рассматривается механизм аргументации для конкретного примера-задачи про муравьев.