UDC 004.89 DOI https://doi.org/10.35546/kntu2078-4481.2025.3.2.32

M. M. KORABLYOV
Doctor of Technical Sciences, Professor,
Professor at the Department of Computer Intelligent Technologies and Systems
Kharkiv National University of Radio Electronics
ORCID: 0009-0005-2540-7741

O. O. FOMICHOV
Candidate of Technical Sciences,
Senior Lecturer at the Department of Electronic Computer Devices
Kharkiv National University of Radio Electronics
ORCID: 0000-0001-9273-9862

O. K. TKACHUK
Postgraduate Student at the Department of Computer Intelligent Technologies and Systems
Kharkiv National University of Radio Electronics
ORCID: 0009-0006-2943-9887

CONTROLLING BOTS' BEHAVIOR IN RPG GAMES USING THE IMMUNE MODEL OF CLONAL SELECTION

The development of game character control models that use artificial intelligence methods is extremely important. The issues of creating intelligent game bots that allow simulating the behavior of players in an RPG (Role-Playing Game) are considered. With the help of the game design created of the project, a futuristic world is simulated in which the player controls spaceships that compete with other similar spaceships. Various options for attacking or restoring ships are used according to the completed description of their capabilities, where the main attention is focused on the features of their action. The game provides for the implementation of several factions, which allow you to create different types of spaceships and give them specific properties. Each ship from the user's team, as well as from the enemy's team, can have different ranks, which significantly impact both the parameters and the cost of this ship. The game application uses the theory of artificial immune systems (AIS) for effective control of game bots, namely the immune model of clonal selection. For this purpose, a modified immune model of clonal selection, Clonalg-rpg, is proposed, in which the antibodies of the system are bots, and the antigens foreign to the system are ships from the project user team that can interact with antibodies. By executing recovery commands for both the bots of their team and for themselves, the bots have the opportunity to interact with each other. Using specific immune operators that allow performing appropriate actions with the population of antigens and antibodies, the work of the modified Clonalg-rpg algorithm is described. To analyze its effectiveness, several game sessions were conducted, in which different compositions of both user ship teams and bot teams were changed. During the game sessions, teams were formed using specific settings, and not randomly. At the same time, spaceships were divided into three classes according to their parameters and capabilities. The results of the experimental studies showed that the proposed immune model Clonalg-rpg is effective and simple to implement, which makes it possible to use it in other game genres.

Key words: *immune model, clonal selection, spaceship, combat, control, design, game bot.*

М. М. КОРАБЛЬОВ
доктор технічних наук, професор,
професор кафедри комп'ютерних інтелектуальних технологій та систем
Харківський національний університет радіоелектроніки
ORCID: 0009-0005-2540-7741

О. О. ФОМІЧОВ
кандидат технічних наук,
старший викладач кафедри електронних обчислювальних машин
Харківський національний університет радіоелектроніки
ORCID: 0000-0001-9273-9862

О. К. ТКАЧУК
аспірант кафедри комп'ютерних інтелектуальних технологій та систем
Харківський національний університет радіоелектроніки
ORCID: 0009-0006-2943-9887

КЕРУВАННЯ ПОВЕДІНКОЮ БОТІВ У RPG-ІГРАХ З ВИКОРИСТАННЯ ІМУННОЇ МОДЕЛІ КЛОНАЛЬНОГО ВІДБОРУ

Розробка моделей керування ігровими персонажами, які використовують методи штучного інтелекту, є надзвичайно важливою. Розглядаються питання створення інтелектуальних ігрових ботів, які дозволяють імітувати поведінку гравців у RPG-грі (Role-Playing Game). За допомогою створеного ігрового дизайну проекту моделюється футуристичний світ, в якому гравець керує космічними кораблями, що змагаються із іншими подібними космічними кораблями. Використовуються різні варіанти атаки або відновлення кораблів згідно виконаного опису їх можливостей, де основна увага зосереджена на особливостях їх дії. У грі передбачається реалізація декількох фракцій, які дозволяють створювати різні типи космічних кораблів та надавати їм специфічні властивості. Кожен корабель з боку команди користувача, а також з боку команди супротивника, може мати різні ранги, які мають суттєвий вплив як на параметри, так і на вартість цього корабля. В ігровому додатку використана теорія штучних імунних систем (ШІС) для ефективного керування ігровими ботами, а саме імунна модель клонального відбору. З цією метою запропонована модифікована імунна модель клонального відбору Clonalg-rpg, в якій антитілами системи являються боти, а чужорідними для системи антигенами є кораблі з команди користувача проекту, які можуть взаємодіяти з антитілами. Виконуючи команди відновлення як для ботів своєї команди, так і для самих себе, боти мають можливість взаємодіяти між собою. З використанням специфічних імунних операторів, які дозволяють виконувати відповідні дії із популяцією антигенів та антитіл, описана робота модифікованого алгоритму Clonalg-rpg. Щоб проаналізувати його ефективність були проведені декілька ігрових сесій, в яких змінювалися різні склади як команд кораблів користувача, так і команд ботів. В ході ігрових сесій команди формувалися з використанням специфічних налаштувань, а не випадковим чином. При цьому космічні кораблі були поділені на три класи згідно своїх параметрів та можливостей. Результати проведених експериментальних досліджень показали, що запропонована імунна модель Clonalg-rpg є ефективною та простою в реалізації, що робить можливим її використання в інших ігрових жанрах.

Ключові слова: *імунна модель, клональний відбір, космічний корабель, бій, керування, дизайн, ігровий бот.*

Statement of the problem

The rapid development and spread of gaming applications have significantly influenced the behavior, skills, and habits of society. This has led to the spread of game mechanics and features in the organization of work of many mobile and web applications, which has been called gamification [1, 2]. Elements of gamification can now be observed in many areas of modern life – from food ordering and delivery services and distance learning systems to mobile banking applications [3]. Mastering the basic techniques is crucial to creating engaging and successful PC games. The foundation of every successful PC game is a well-thought-out game design that includes story, game mechanics, characters, and environments that work together to create an engaging gameplay experience. Game design is considered the most important aspect of computer game development.

Recent years have been characterized by the rapid development of artificial intelligence systems and the spread of free access to such systems by society for solving various everyday and creative tasks, in particular in gaming applications [4, 5]. The field of artificial intelligence has ceased to be exclusively research and scientific, and has become popular and accessible to everyone, similar to what happened on the computer market in the mid-1990s. Most existing systems of artificial intelligence organization are based on the biological principles of the human nervous or immune systems, as well as the principles of evolutionary research on the genome and behavior of animals and ants. Artificial intelligence has also significantly influenced the development of the modern gaming industry [6, 7].

The work sets the task of creating intelligent game bots that simulate the behavior of players in an RPG (Role-Playing Game) game when countering the actions of a team of user characters, using artificial immune systems, in particular, the immune model of clonal selection. Solving this task will allow us to create an intelligent system for controlling the behavior of bots in a computer game.

Game design project

The game application considered in the work simulates a futuristic world in the style of open space, where the player controls a group of spacecraft, the number of which ranges from 1 to 6 units, which compete with another group of similar spacecraft of a similar number [8]. The principle of game combat is borrowed from the popular RPG game Disciples II in the mid-2000s, but with a significantly simplified game design. The game provides for several game factions that can produce their own specific types of spacecraft with special capabilities and properties. At the same time, each spacecraft in the user's team or in the team of his opponent has a list of characteristics that determine its capabilities: cost, ship level, combat experience, strength, energy, attack, protection, speed, and recovery.

The game provides for several factions that create their own types of spaceships and give them specific properties. For ease of implementation, the factions are conventionally designated by colors, namely: yellow, blue, and red. The main feature of the ships of the yellow faction is an increase in the strength indicator by 10 %, compared to ships of other factions; the feature of the ships of the blue faction is an increase in the protection indicator by 10 %, compared to ships of other factions; and the main feature of the ships of the red faction is an increase in the recovery indicator by 10 %, compared to ships of other factions. Thanks to this, each faction can increase the significant characteristics of its ships uniquely compared to the ships of other factions. In addition, each ship, both from the user's team and from the opponent's team, can belong to one of the ranks. These ranks significantly affect the cost and parameters of this ship. The list of main ranks is defined as follows: support, scout, cruiser, frigate, destroyer, and dreadnought.

To describe the capabilities of ships, various attack or recovery options were used, where attention was focused on the features of their action from the point of view of game design. First, two types of attack were used in the description of the capabilities of ships: close and long-range attacks. In this case, a close attack implies the ability of the ship to attack only those enemy ships that are on the first line of the battlefield in front of it, i.e. on positions 2, 4 and 6 of the game battle map (Fig. 1). A long-range attack implies the ability to attack enemy ships that are on the second line in the enemy team, i.e. on positions 1, 3 and 5 of the game battle map.

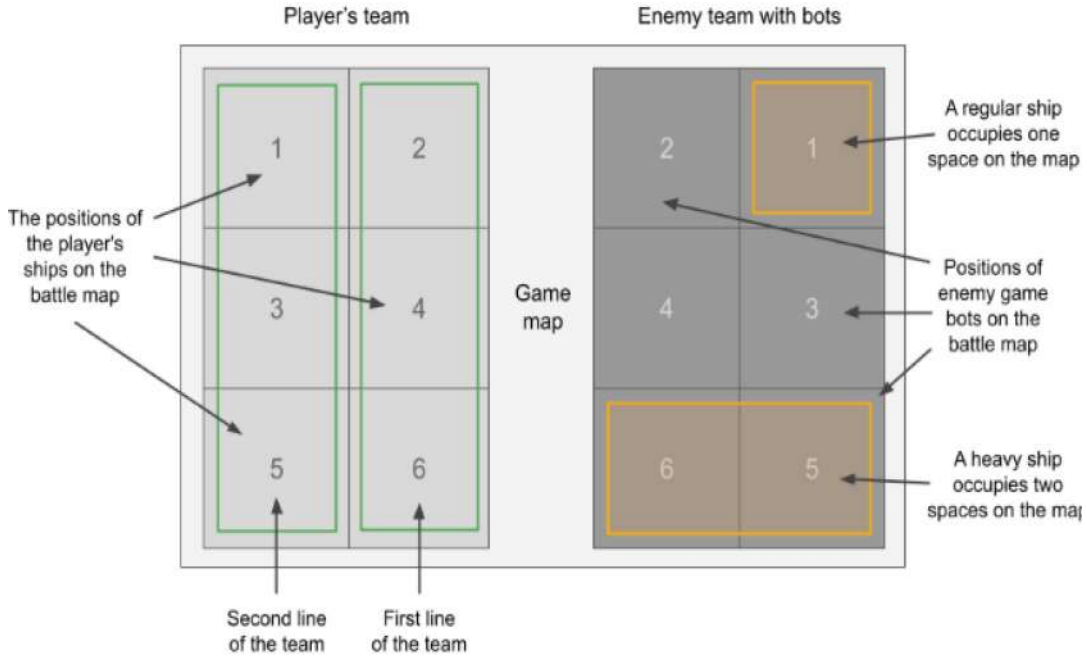

Fig. 1. General view of the game battle map

The definition of the method of attack between ships has a significant impact on the gameplay [9, 10]. Accordingly, and also taking into account the defined types of close and long-range attacks, it is necessary to describe the appearance of the playing field for the battle between ship teams. The entire playing field is conventionally divided into three parts, where the first part is occupied by the first player's ship team, the second part divides the teams among themselves, and the third part is occupied by the second player's team. It should be noted that in each team, the ships are lined up in two rows – the one closest to the enemy and the one far from the enemy, and only the ships that occupy two places on the game battle map occupy both rows, that is, they can be attacked by both close and long-range attacks. In addition, in the case of the destruction of all ships of the first line, all ships that are on the second line can be attacked by any method of attack.

In the game, a great influence on the capabilities of a ship is its level, namely, it increases the strength, defense, and attack indicators by 1 % of the initial values of a ship of a certain type of the first level, taking into account the characteristics of the faction. The level increase occurs due to the victory over the enemy team, where for every 100 units of strength of each destroyed ship of the enemy team, 10 units of experience are accrued, which are evenly distributed between all ships that participated in the battle. In addition, the method of assigning damage from attacks between ships, which is determined by the expression:

$$damage = \frac{e_1}{e_2} \cdot (A_1 - D_2), \tag{1}$$

where *damage* is the value of the adjusted damage that a ship inflicts on another during the attack; A_1 is the attack value of ship 1; e_1 is the energy value of ship 1, D_2 is the defense value of ship 2; e_2 is the energy value of ship 2.

An important influence on the gameplay is the way a game ship can be restored, which can be done during its turn instead of performing an attack. This indicator is determined as follows:

$$rp = R \cdot ke, \tag{2}$$

where rp – value of the restored strength of the ship; R – characteristic of ship recovery; e – ship energy value, a k – recovery factor, which is chosen randomly from the range [1.1; 1.25].

An important feature of the gameplay is the Player versus Enemy (PvE) mode, i.e., a game where the user plays against a team controlled by the game itself.

Controlling gaming bots based on the immune model of clonal selection

The task of controlling game bots in a battle with a team of user ships is reduced to modeling the behavior of each ship of the enemy team, taking into account the capabilities and characteristics of each specific ship of the player's team. Accordingly, game bots are only ships of the game team/system, with which the user's team is fighting. For effective control of game bots, it is currently advisable to use artificial intelligence systems, among which, along with neural networks, fuzzy logic, genetic algorithms, etc., the theory of artificial immune systems (AIS) occupies an important place. The most common AIS models used to solve various practical problems are the immune model of clonal selection and the artificial immune network model [11].

Controlling game bots can be considered as a special case of the optimization problem, for the solution of which the immune model of clonal selection can be used [12, 13]. According to this model, bots are considered as antibodies of the system that interact with foreign antigens to the system, which for bots are ships from the project user team. In addition, bots can interact with each other, executing the recovery command for themselves and for the bots of their team.

The ClonAlg method is one of the most common methods that operates on the basis of the clonal selection model. This method can be used to solve the problem of controlling game bots that oppose the user's ship team, which controls them personally. In this method, the method that will control the bots plays a key role. The classic version of the ClonAlg immune algorithm provides for the possibility of changing antibody parameters during their interaction with a population of foreign antigens by cloning, mutation, and self-regulation of the antibody population through clonal selection and aging. In this case, changes in antigen parameters were either not foreseen at all or were insignificant. In addition, the possibility of antigen disappearance in the process of forming an immune response by the system was not foreseen as such. But in our case, during the confrontation of the user's ship team with the system's game bot team, such a possibility is foreseen.

With this method of operation of the immune algorithm, the destruction of all antigens that represent the user's ship team is the only thing that can be used to solve this problem without a number of changes. Accordingly, the modified version of this algorithm, Clonalg-rpg, has several important features. First, the algorithm cannot directly change the values of the antibody features during its operation, but can only make decisions about the appropriateness of executing a particular command for each specific bot or its clone. Secondly, the choice of a solution for executing a particular bot command should be made from the point of view of the maximum efficiency of this action of destroying the entire population of antigens, or from the point of view of restoring the strength indicator of the entire population of antibodies. Thirdly, the condition for terminating the operation of the algorithm Clonalg-rpg is the destruction of the entire population of antigens, represented by the set of ships of the game user, before the population of antibodies, represented by game bots, ceases to exist. Fourth, dynamic changes in the features of antibody bots and antigen ships occur constantly at the user's command. Thus, the implementation of the modified version of the immune algorithm Clonalg-rpg should be based on four defined principles as the basis of the algorithm's work. In addition, an important feature of the clonal selection model is that in the process of work, antibodies focus on interaction with antigens, and then use some opportunities for interaction with other antibodies in the process of self-regulation of the antibody population.

The operation of the modified Clonalg-rpg method, created to solve the problem of controlling game bots, can be conditionally described at the level of immune operators as follows:

$$Clonalg - rpg = \begin{bmatrix} Presentation(AB, AG) \rightarrow \\ Cloning(AB, CL) \rightarrow \\ Mutation(AB, CL) \rightarrow \\ ClonSelection(AG, CL) \rightarrow \\ SelfRegulation(AB, CL) \rightarrow \\ Termination(AB, AG) \end{bmatrix}^{AG>0}, \tag{3}$$

where *Presentation*(*AB*, *AG*) is the operator for presenting the set of antigens *AG* of the antibody population *AB*; *Cloning*(*AB*, *CL*) is the operator for cloning the antibody population; *Mutation*(*AB*, *CL*) is the operator for mutation of the formed clones; *ClonSelection*(*AG*, *CL*) is the operator for selecting clones; *SelfRegulation*(*AB*, *CL*) is the operator for self-regulation of the antibody population; *Termination*(*AB*, *AG*) is the operator for checking the possibility of stopping the algorithm.

The work of the *Presentation*(*AB*, *AG*) operator is to determine the affinities between antibodies and antigens, as well as the affinities between the population of antibodies of the AIS. At the same time, affinity is defined as the degree of similarity between the features of an antibody and an antigen, and the work of the immune algorithm is reduced to ensuring that, during the process of cloning and mutation, the clones reproduce the features of a particular antigen. The determination of antigen affinity is described by the expression:

$$affAg_i = \frac{1}{1+\left(Ds_i - Dc_i\right)+\left(Es_i - Ec_i\right)}, \tag{4}$$

where$affAg_i$ is the affinity of the *i*-th antigen; Ds_i is the saved strength indicator of the i-th antigen at the start of the battle; Dc_i is the current strength indicator of the *i*-th antigen at the current battle iteration; Es_i is the saved energy indicator of the *i*-th antigen at the start of the battle; Ec_i is the current energy indicator of the *i*-th antigen at the current battle iteration.

The antibody cloning operator *Cloning*(*AB*, *CL*) is used to create a set of clones for each antibody in the current AIS population for their further mutation. The number of clones produced by using the adaptive cloning operator is defined as:

$$n_i = C_i \cdot N_{AG}, \tag{5}$$

where n_i is the number of clones of the *i*-th antibody; C_i is the number of interaction variants between the antibody and the antigen; N_{AG} is the number of antigens.

The clone mutation operator Mutation(CL) significantly affects the speed of solving the target problem set for the AIS. It is the mutation mechanism that ensures changes are made to immune objects and brings the AIS closer to solving the problem. The Clonalg-rpg algorithm assumes the use of an adaptive mutation operator, which focuses on choosing the method of interaction with antigens.

According to the clone selection operator *ClonSelection*(*AG*, *CL*), one immune object, or one way of interacting with an active antigen, or a set of antigens, is selected, which makes the greatest contribution to solving the target problem.

The antibody population self-regulation operator *SelfRegulation*(*AB*, *CL*) has been added to the clonalg-rpg algorithm instead of the apoptosis operator.

The antibody population self-regulation operator *SelfRegulation*(*AB*, *CL*) has been added to the Clonalg-rpg algorithm instead of the apoptosis operator, which involves determining affinities or other metrics between the antibody population and the clones selected in the clonal selection process. The determination of the internal affinity index of a clone to the entire antibody population has the form:

$$aff'Ab_i = \frac{1}{S}\sum_{i=1}^{S}\left(1+\left(Ds_i - \left(Dc_i + \frac{1}{S}R\right)\right)\right)^{-1}, \tag{6}$$

where $aff'Ab_i$ is the affinity of the clone with the entire population of active antibodies, including the *i*-th antibody from which this clone was created; *S* is the size of the population of active antibodies; Dc_i is the strength indicator of the *i*-th antibody at the current iteration of the battle; *R* is the clone recovery indicator.

The possible mode of action for restoring the strength characteristic for a single antibody, or the entire antibody population, is selected according to the value of the greater affinity. In the case where a game bot has only one recovery method, the affinity between it and the active antibodies is determined only once.

The operator for checking the possibility of stopping the battle *Termination*(*AB*, *AG*) is used to check the possibility of stopping the game battle process. The Cloanlg-rpg algorithm uses a criterion operator for stopping the operation of the AIS, which involves stopping the operation of immune operators in the absence of active antigens or active antibodies.

Experimental studies

When studying the effectiveness of the Clonalg-rpg algorithm, about a hundred tests were conducted with different compositions of user teams and teams of game bots of different types and different factions. During these tests, teams were formed not randomly, but through specific settings, taking into account that spaceships can be divided into three classes according to their parameters and capabilities, where the first class is formed by the cheapest *support* and *scout* ships, the second class is formed by *cruiser* and *frigate* ships, and the third most expensive class is formed by heavy ships, which occupy two places on the game battle map – *destroyer* and *dreadnought*. At the same time, the parameters of these ships, such as attack power, energy, durability, etc. in the game balance are adjusted in such a way that first-class ships are inferior to second-class ships by approximately 25 %, and to third-class ships by approximately 50 %. Accordingly, second-class ships are inferior to third-class ships by 25 % in their basic indicators. However, for the sake of game balance, first-class ships outperform all other ships in terms of speed and make their move during a game battle first.

The Clonalg-rpg algorithm was tested on the same set of input data about the user's ship commands and game bots in three ways: without using the AIS, when the user controls both commands directly (this is an atypical way of working of the project, which is configured in a specific way); without using the AIS by choosing the game bot's action method randomly; and with using the Clonalg-rpg algorithm, which controls the behavior of game bots in battle. The main indicator of the effectiveness of the immune algorithm for controlling game bots is the number of victories of the AIS over

a person in game battles in which teams of ships with comparable or equal basic parameters participate. In addition, the effectiveness study took into account not only the percentage of victories of the AIS over the user, but also the number of ships remaining in the winning team. Accordingly, the following indicators were identified:

– the winner has 1 ship left;
– the winner has 2 ships left;
– the winner has 3 or more ships left.

Using such detail will allow you to investigate not only the fact of victory itself, but also the extent to which the winner dominated his opponent during the game battle, which may indicate the level of skill of the game. Table 1 shows the percentage of victories of the user's teams over their opponent's teams, taking into account the different ways of controlling the opponent's team, defined earlier.

Table 1

Frequency of user wins over opponent

Types of commands	Manual control	Random control	Control Clonalg-rpg
5 on 5	50 %	85,15 %	50,35 %
4 on 4	50 %	96,75 %	49,15 %

For each type of control of the team of game bots, 20 game battles were conducted to obtain statistical data on the results of battles and the choice of the type of actions of bots during the game battle. According to the results presented above, the least effective is the control of game bots made randomly, which almost always led to a guaranteed defeat. In contrast, the organization of game bot control based on the Clonalg-rpg algorithm showed results comparable to those obtained by involving a person to control the enemy ships. When reducing the number of ships in the team by using heavy ships, which have a greater variability of possible actions, the chance of victory of game bots increases slightly, which may indicate that in the case of increasing the number of attack and recovery options, the proposed immune method Clonalg-rpg will show greater efficiency than the user.

In order to assess the playing skill of the winning team, the number of ships remaining in the winning team after the game battle was recorded; these results are shown in Table 2. It can be seen that using the method of controlling game bots randomly creates the easiest conditions for the user, because in most cases, it was the user who won over the bot team, which is controlled randomly.

Table 2

Number of ships remaining in the winning team

Types of commands		Manual control	Random control	Control Clonalg-rpg
5 on 5	1	100,00 %	12,40 %	93,75 %
	2	0,00 %	82,35 %	6,25 %
	3+	0,00 %	5,25 %	0,00 %
4 on 4	1	100,00 %	3,60 %	95,00 %
	2	0,00 %	88,25 %	5,00 %
	3+	0,00 %	8,15 %	0,00 %

We can conclude that the use of the Clonalg-rpg algorithm creates difficult conditions for the user to win over a team of game bots, with the winning team usually having only one ship left, while the others are lost during the game battle.

A separate area of observation was the type of action that the game bot chooses and performs during its turn. This choice plays an important role, as it is what leads the game bot team to victory over the user's team or defeat. When allowed to make a move in the order of the queue, which is formed based on the speed of all active ships from both teams participating in the game battle, you can choose either to attack the enemy or to recover. It is impossible for either the game bot or the ship that is under the direct control of the user to miss a move. Table 3 shows statistical data on the frequency of choosing a particular method of action by game bots that are under the different types of control defined earlier.

It can be concluded that the immune algorithm Clonalg-rpg gives a greater advantage to the III method of attack, both at close and long distances, while minimizing the use of the II method of distributed attack of the enemy team's ships. In the case of a game bot's decision to attack the enemy under the guidance of Clonalg-rpg, the most attention is paid to methods of concentrated attack on a specific ship with the possibility of inflicting additional energy damage to a ship from the enemy team, which significantly affects its further capabilities in battle. In the case of recovery, the method of concentrated recovery of one selected ship is also preferred. This behavior can be described as a strategy of concentrated destruction of the weakest ships of the enemy team while maximally concentrated recovery of damaged ships of one's own team.

Table 3

Frequency of choosing methods of action in team battles

Variants of ship actions	Manual control		Random control		Control Clonalg-rpg	
	5 on 5	4 on 4	5 on 5	4 на 4	5 on 5	4 on 4
Method I of the close attack	19,33 %	9,82 %	24,56 %	18,89 %	20,29 %	11,01 %
Method II of the close attack	9,83 %	10,42 %	11,22 %	10,56 %	10,86 %	8,57 %
Method III of the close attack	3,15 %	7,89 %	2,22 %	5,56 %	3,90 %	9,76 %
Method I of long-range attack	22,00 %	6,05 %	17,89 %	10,56 %	20,00 %	6,25 %
Method II of long-range attack	5,83 %	10,62 %	6,22 %	10,56 %	4,86 %	8,57 %
Method III of long-range attack	3,15 %	7.89 %	2,22 %	5,56 %	3,90 %	9,76 %
Method I of recovery	19,87 %	18,65 %	24,56 %	18,89 %	19,67 %	18,75 %
Method II of recovery	14,02 %	22,42 %	8,89 %	13,89 %	14,52 %	22,32 %
Method III of recovery	2,50 %	6,25 %	2,22 %	5,56 %	2,00 %	5,00 %

It should be noted that the proposed Clonalg-rpg method is used to control game bots in RPG projects for the first time, so today it is quite difficult to make a comparison with other immune and non-immune methods used to organize a game bot control system. It is also important to note that the proposed immune method is easy to implement and modify, which makes it possible to further use it to control game bots in games of other genres.

Conclusions

The solution of the current problem of controlling game bots in RPG games using a modified immune model of clonal selection is considered. A game application has been developed that simulates a futuristic world in the style of open space, where the player controls a group of spacecraft that compete with another such group of similar spacecraft. At the same time, each spacecraft in the user's team or in the team of his opponent has a list of specific characteristics that determine its capabilities. The game provides for several factions that create their own types of spacecraft and endow them with specific properties. Thanks to this, each faction can increase the significant characteristics of its ships in a unique way compared to the ships of other factions. In addition, each ship, both from the user's team and from the side of its opponent's team, can belong to one of the ranks, which significantly affect the cost and parameters of this ship. To describe the capabilities of the ships, various attack or recovery options were used, which focused on the features of their action from the point of view of game design.

Game bot control is considered as a special case of the optimization problem, for the solution of which the modified immune model of clonal selection Clonalg-rpg is used, according to which bots are considered as antibodies of the system that interact with foreign antigens to the system, which for bots are ships from the project user team. The work of the modified algorithm, Clonalg-rpg is described at the level of immune operators that perform the corresponding actions with the population of antibodies or antigens representing the bots of the system and the user.

During the experimental study of the effectiveness of the Clonalg-rpg algorithm, about a hundred tests were performed with different compositions of user teams and teams of game bots of different types and different factions. During these tests, the teams were formed not randomly, but through specific settings, taking into account that the spaceships were divided into three classes according to their parameters and capabilities. The results of the experimental studies showed that the proposed immune model Clonalg-rpg is effective and simple to implement, which makes it possible to use it in other game genres.

References

1. Panchanadikar R., Freeman G., Li L., Schulenberg K., & Hu Y. (2024). A New Golden Era' or 'Slap Comps': How Non-Profit Driven Indie Game Developers Perceive the Emerging Role of Generative AI in Game Development." Extended Abstracts of the CHI Conference on Human Factors in Computing Systems. NY, USA: ACM. 1–7. https://doi.org/10.1145/3613905.3650845

2. Immonen J. (2024). Primary Tools and Techniques in Game Development. Bachelor's thesis. Lappeenranta-Lahti University of Technology LUT, 43 pages. https://lutpub.lut.fi/bitstream/handle/10024/168411/

3. Szolin K., Kuss D. J., Nuyens F. M., & Griffiths M. D. (2023). "I am the character; the character is me": A thematic analysis of the user-avatar relationship in video games. Comp. in Hum. Behavior, Vol. 143, 107694. https://doi.org/10.1016/j.chb.2023.107694

4. Quiroga M. A., Diaz A., Román F.J., Privado J., & Colom R. (2019). Intelligence and video games: Beyond "brain-games". Intelligence, Volume 75, 85-94. URL: https://www.elsevier.com/locate/intell

5. Simons A., Wohlgenannt I., Zelt S., Weinmann M., Schneider J., and vom Brocke J. (2023). Intelligence at play: game-based assessment using a virtual-reality application. Springer https://doi.org/10.1007/s10055-023-00752-9

6. Lum Y., & Li W. (2022). Techniques and Paradigms in Modern Game AI Systems. Algorithms, 15(8), 282. https://doi.org/10.3390/a15080282

7. Fan X., Wu J., & Tian L. (2020). A Review of Artificial Intelligence for Games. Part of the Lecture Notes in Electrical Engineering book series (LNEE, vol. 572, 1821. URL: https://doi.org/10.1007/978-981-15-0187-6_34

8. Snyder R. (2023). 10 Best Spaceships in Video Games URL: https://www.dualshockers.com/best-spaceships-in-video-games/

9. Mohd T., Bravo-Garcia F., Love L., Gujadhur M., Nyadu J. (2023). Analyzing strengths and weaknesses of Modern Game Engines. International Journal of Computer Theory and Engineering, 15(1), 54–60. https://www.ijcte.org/vol15/IJCTE-V15N1-1330.pdf

10. Silić M., Džaja B., Rogulj R., & Turić H. (2024). Optimizing Game Ready Asset Creation Pipeline: From Concept to Implementation in Unreal Engine 5. ORGANIZER, 732. https://hdl.handle.net/11159/654467

11. Dasgupta D., Yu S., & Nino L. F. (2011). Recent Advances in Artificial Immune Systems: Models and Applications. Applied Soft Computing. Elsevier, 1574–1587. URL: https://https://doi.org/10.1016/j.asoc.2010.08.024

12. Korablyov M., Fomichov O., Chubukin O., Antonov D., & Dykyi S. (2023). Immune Model for Controlling Characters in Computer Games. XI International Scientific and Practical Conference “Information Control Systems and Technologies (ICST)”, 214–226. https://ceur-ws.org/Vol-3513/

13. Shekhar S., Sharma D. K., Agarwal D. K., & Pathak Y. (2022). Artificial Immune Systems-Based Classification Model for Code-Mixed Social Media Data. IRBM, Vol. 43, Iss. 2, 120-129. URL: https://doi:10.1016/j.irbm.2020.07.004

Дата першого надходження рукопису до видання: 19.09.2025
Дата прийнятого до друку рукопису після рецензування: 15.10.2025
Дата публікації: 28.11.2025