INFORMATION TECHNOLOGY

DOI 10.20535/2411-1031.2021.9.2.249882
UDC 004.942

YAROSLAV DOROHYI,
OLEKSANDR SVYRYDENKO

METHODS AND APPROACHES ANALYSIS OF ARTIFICIAL INTELLIGENCE DESIGNING FOR REAL TIME STRATEGY GAME

The research provides a detailed analysis of approaches to creating AI in video games. The main area of research is AI for real-time strategies, as this genre is characterized by the complexity of the game environment and the practice of creating a comprehensive AI, consisting of several agents responsible for a particular aspect of the game. The analysis shows that the main areas of use of AI methods in strategies are strategic and tactical decisions, as well as analysis of the current situation and forecasting the enemy and his chosen strategy. Among the analyzed approaches to tactical AI, reinforcement, game tree search, Bayesian model, precedent-based solutions and neural networks are most often used. Popular approaches to building strategic AI are precedent-based decision-making, hierarchical planning, and autonomous achievement of goals. When creating a module for research and determination of plans, the most popular methods are deductive, abduction, probabilistic and precedent. In addition to the considered methods, others are used in the development, but they are not as popular as above, due to problems with speed or specific implementation, which does not allow to adapt them to the standard rules of genre games. Comparison of algorithms and implementations of AI in the framework of commercial and scientific developments. Among the main differences are the high cost of commercial development of complex agents, as well as the specifics of the scientific approach, which aims to create the most effective agent in terms of game quality, rather than maximizing positive impressions of players, which is the basis of commercial development. The reasons for insufficiently active development of scientific research in the field of AI for games in general and the genre of real-time strategies in particular are described.

Keywords: artificial intelligence, real-time strategies, video games, neural networks, tactical decision-making, strategic decision.

Problem statement. Artificial intelligence (AI) is widely used in various fields – from medicine to economics. However, despite this proliferation, computer games remain the most promising area for the development and analysis of the effectiveness of AI. In particular, real-time strategies should be distinguished among different genres. The reason for this is the complexity of the tasks facing the player, and AI in games of this genre.

Analysis of recent research and publications. AI for playing chess, checkers or other board games and its ability to play at the human level is familiar to us. In addition, after the victory of AI DeepBlue over grandmaster Kasparov [1], researchers have proven that AI in turn-based games can beat even the highest level of professional players. With real-time strategy games, the situation is very different. Unlike turn-based games, where players take turns seeing enemy positions change, real-time strategies change every second, players act synchronously, and one of the key elements is the use of "fog of war" technology – lack of visibility in parts of the map where the player has no control units or buildings. This significantly complicates the ability to analyze the current situation, thus increasing the difficulty of determining further action. One of the researchers of AI for real-time strategies, Michael Buro, said [2]: "To get a feeling for the vast complexity of RTS games, imagine to play chess on a 512×512 board with hundreds of slow simultaneously moving pieces, player views restricted to small areas around their own pieces, and the ability to gather resources and create new material".

Due to the high complexity and complexity of the problems that AI must solve in real-time strategies, the decision-making mechanism has a modular structure and consists of a set of agents, each of which is responsible for a separate component of analysis and decision-making. Because the data and expected results for each of the components are different, researchers and commercial developers use different algorithms and approaches to create a component of the decision-making mechanism.

The aim of this paper is to identify the main components into which AI is divided in strategies, as well as to determine the approaches and algorithms used to implement each of them. In addition, the study analyzes the difference between commercial and research work on the development of AI for strategies and identifies the reasons for the current state of development of AI for games.

The main material research. Areas of use of AI in real-time strategies. Popular commercial projects in the genre of real-time strategies have typical rules that apply to the entire genre. Usually at the beginning of the game, everyone has a center and several non-combat units – workers who build buildings and extract resources. Depending on the needs of the building there are different types: resource, technological, protective or manufacturers of units. Depending on the chosen strategy, the player can determine which building or unit is needed at the moment. Making these decisions has a big impact on the future of the game and depends on the strategy of the player as a whole. In this part, which researchers also call macro-control, the bot is responsible for the strategic decision-making module.

Upon completion of the construction of buildings in which you can hire combat units, the player gets access to the army. The army consists of units of various types and purposes. Units must be used in accordance with their characteristics – units designed to deter the enemy, needed on the front, support units must be in the rear. The tactical decision-making module is responsible for controlling troops during skirmishes or microcontrol.

From the beginning, only a small part of the map is available to the player – most need to be explored. Also, the player does not know the position or positions of opponents and the current situation with buildings and troops. This requires intelligence – the direction of their units in the parts of the map that are covered by the "fog of war" and analysis of the situation. The intelligence and planning module is responsible for this.

It is worth noting that in addition to the above three modules of AI in real-time strategies, there are also training bots for the correct response to the situation. Researchers often combine this part of AI with a module to determine enemy plans.

Tactical decision-making. To create tactical AI in the bot, researchers and engineers use different approaches, the most common options:

– reinforcement learning;
– game-tree search;
– bayesian mode;
– case-based reasoning;
– neural networks.

Reinforcement learning is an area of machine learning where the agent must choose actions that maximize conditional reward. Unlike teacher training, the agent is not provided with the correct input and output parameters, and suboptimal agent decisions are not corrected.

For StarCraft in 2011 [3], researchers created a module for tactical decision-making, which was based on a combination of a neural network and a training algorithm with reinforcement learning, which was used to implement tactical control for each of the soldiers. For small groups, the algorithm has proven to be much more efficient than embedded scripts.

In 2017, the researcher considered the use of classical learning with reinforcement for real-time strategies, and also proposed solutions to some problems, such as solving a problem with a large branching factor [4].

Another group of scientists from the Netherlands in 2018 used reinforcement learning for testing in their own environment [5]. The agent combined hierarchical reinforcement learning with a multi-level perceptron that handled high-level commands to improve the quality of system training.

Case-based reasoning are often used in strategy games, where microcontrol is more important than strategic planning. Making decisions based on precedents in a broad sense is a choice of actions based on already known sequences. Created in 2009 by researchers, the Warcraft module [6] was able to effectively prove itself in microcontrol in battles with built-in AI and become an effective assistant to the player, providing effective microcontrol of troops, allowing the player to focus on high-level planning. Case studies can also be combined with reinforcement learning [7] for the effective use of units and their interaction at the tactical level.

Game tree search is one of the algorithms that is rarely used to make strategic decisions, but it can perform tactical decisions well.

Researchers [8] have created a system that, analyzing known strategies, sought Nash equilibrium – under this equilibrium, the player cannot gain an advantage (increase winnings) by changing the strategy unilaterally. Only the tactical component was selected for the study, and micro-management by each individual unit was excluded – management took place in groups and concentrated on movement. As a result, the system was able to defeat the bot in the game in almost all situations, because for each specific strategy developed by the developers of the game bot, the system responded with an improved strategy.

For closed source games (which are most commercial projects) it is quite difficult to use search-based algorithms. Therefore, in 2012, researchers to test AI based on a search for a game tree in the game StarCraft created an analogue of the game called SparCraft, with somewhat simplified rules, but in general the same gameplay [9]. The test results showed the effectiveness of the agent in 92% of cases, given the limited time to choose a solution – all calculations took place in real time.

The Bayesian model, which is a set of random variables and dependencies between them using an oriented acyclic graph, can also be used in tactical decision-making to determine the direction of movement of units [10].

Neuroevolution is a technique of using evolutionary algorithms to train a neural network. The rtNEAT approach [11] can be used to influence both the coefficients and the topology of the neural network model to control each individual unit. The input to the model includes data that the unit receives from the environment, as well as from neighboring units. Each neural network carrier had a specific current efficiency function that depended on position, health, level, and other parameters. Units that showed weak combat performance data were replaced by more efficient units. The testing was conducted in conditions that are quite convenient for AI, but it showed its effectiveness with the same number of units and the same type on both sides.

In general, the use of neural networks in the construction of agents for tactical decision-making is a common practice. For example, researchers using the convolutional neural network (CNN) were able to achieve significantly better results compared to other algorithms [12]. In addition, it should be noted that such neural networks are slower than most algorithms that are actively tested in our time, but despite this, convolutional neural networks show better results. In addition, the researchers suggest that the combination of the latest advances in hierarchical search and convolutional neural networks is the key to defeating human players in such a complex real-time strategy space for AI agents.

Convolutional neural networks are also used for such tasks as determining the winner in the current situation [13]. μRTS (real-time micro-strategy) was used to test the solution, and a specific type was used for the neural network – multidimensional convolutional neural networks.

Making strategic decisions. Strategic decisions, or macromanagement, are a high-level part of AI in strategy games and affect the game in the long run – the technology studied or the building built at the beginning of the game can change the situation and affect the development of events at the end of the game. Planning systems are used to determine strategic actions – they have shown their effectiveness in real products and research. The complexity of the problem is that strategic decisions depend on the opponent’s decisions, but most of the enemy’s actions and movements are inaccessible to the player or bot through the fog of war (FOW), which closes the part of the map where there are no player units / buildings. Because of such problems, decisions have to be made based on incomplete or even missing data, which can significantly affect the consequences. The main approaches are precedent-based planning, hierarchical planning and autonomous achievement of goals. Behavioral trees or evolutionary systems are less commonly used.

The basic options for AI for games are to create a set of goals with a conditional reward. Such approaches were used in the first games in the genre of real-time strategy and the task of the bot was to score as many points as possible by performing different tasks and achieving different goals. The value of the task changed during the game depending on the current situation.

Decision making based on precedents in the strategic module has a similar scheme to the tactical one: developers create a set of known sequences of actions depending on the current situation, and AI seeks the most similar options for the current situation and uses them. The advantages of this approach compared to others are its adaptability to new strategies and the lack of need for lifelong learning.

For the first time, AI for making strategic decisions using precedents was created by researchers in 2005 [14]. An important advantage was the adaptability to different strategies of the enemy – at that time, most agents of AI showed good results only against the “static” enemy, who had one scripted strategy. When changing the strategy of the enemy, the agent had to go through an additional stage of training and adaptation.

The combination of decision-making based on precedents with the idea of fuzzy sets allowed to abstract information about the state [15]. This allowed researchers to greatly simplify the state space. The choice of strategic decisions is guided by the number of buildings and units.

Hierarchical planning is a system of distribution of goals in the game in a hierarchical network of tasks, where there are both basic high-level tasks (victory over the opponent) and low-level tasks, which are subtasks for high-level (hire a worker). Hierarchical planning is often combined with other algorithms for strategic planning, such as precedent-based decisions.

Hierarchical planning is often formalized by researchers into a well-defined and structured hierarchical network of tasks. In such a network, high-level tasks can be divided into a series of sequential actions to be performed by the agent [16].

A hierarchical network of tasks was used to create a strategic decision-making agent in the open source strategy Spring [17]. This agent architecture allowed him to respond effectively to the loss of buildings or to better extract resources. The approach has shown high effectiveness against embedded AI.

Another approach using hierarchical planning is to create a network of sequential, parallel and conditional actions, which are performed depending on the current state [18]. All solutions are synchronized through a conditional “board”, where the results of previous tasks are stored.

Autonomous achievement of goals is a technique of choosing sequences of actions that change depending on the situation. Upon receipt of new data, the agent can change the sequence of actions that will indicate the implementation of the goal [19]. This allows the bot to respond effectively to abrupt changes in the game and be flexible in achieving goals without switching from them. This approach is often combined with others, such as precedent-based decision-making.

To build the agent, the researchers created a system using ABL (A Behavior Language). A feature of the system was the ability to adjust plans in the event of unexpected events. Plans could also be implemented in parallel if they were independent [20].

Exploration and determination of plans. From strategic planning as a separate module can be distinguished the technique of determining the enemy’s plan and strategy. Because of the fog of war technology often used in strategies, the definition of plans is almost always based on incomplete information, so technicians often use a ready-made knowledge base or training module. In this part the following techniques will be considered:

– deductive;

- abductive;
- probabilistic;
- precedent.

Defining a plan by the deductive method determines the plan by comparing the current situation with a hypothetical situation from a set of known plans. With the help of the deductive method, it is possible to determine the enemy's plan even from a limited amount of information, because conclusions can be drawn even from the first beginnings of the preparation of a certain plan [21].

Developed by researchers [22], the tree of solutions for the game Starcraft can also be considered one of the implementations of the deductive method for recognizing plans. A database of game repetitions was used to train the system, focusing on the sequence of buildings and technologies.

The abductive method is the ability to deduce from the combination of hypotheses and conclusions additional hypotheses. Based on this data, the planner creates a goal that will be adjusted if the hypothesis is not confirmed or new data about the enemy indicate the need to use other actions and set new or expand the list of current goals. Researchers have created a strategic AI using the technology of autonomous achievement of goals and an abductive method to determine the plan and strategy of the enemy [20]. This method requires constant improvement of the base of goals and new conditions to set precise goals or better adjust existing ones.

Probabilistic methods use statistics and probabilistic estimates of actions and their results at different intervals. To build a model of the probabilistic strategy of the enemy's development through the analysis of the order of construction of buildings use records of games of real players without prior training.

Probabilistic methods are used with Markov models – player development is represented by a set of states and connections (potential transitions) between these states [23]. It is worth noting that the approach used depends on the specifics of the faction (race / nation) and is used in the initial stages of the game.

The previous study was developed with the addition of a dynamic Bayesian model and took into account the reconnaissance of the enemy base and potentially unexplored areas [24].

AI in commercial and scientific projects. One of the important problems for AI in gaming systems in general and in the genre of real-time strategy in particular, is the different goals for business and for scientists. For business, the main goal of AI is to maximize player satisfaction with the product. For scientists, the main goal is to maximize efficiency, so the maximum approximation of the number of victories of AI over players to 100%. This is the main problem and the reason for using completely different algorithms, because players in most cases do not want bots to lose almost all the matches.

Another important factor is the volume of research, and therefore their high cost, because real-time strategies are real conflicts that are simulated in an environment with simplified rules, but still very close to real ones. Some gaming companies are still trying to create quality AI and do so quite successfully. For the game Blitzkrieg 3 in 2017, a tactical AI based on a neural network was created [25], which after training led troops at the level of professional players.

Conclusions. The study examines scientific and commercial approaches to creating AI. There are three main areas where solutions are created using AI in modern games in the genre of real-time strategy: tactical decision-making or micromanagement, strategic decision-making or macromanagement, and planning and intelligence, this area also includes training. Approaches to the creation of AI in each of the three areas are considered, the results of research are analyzed.

Summing up the results, we can say with confidence that currently there is no single right approach to creating AI. Good results show methods in which researchers combine different approaches and algorithms. Artificial neural networks and precedent-based planning should be singled out – in combination with other algorithms and a high-quality learning system, the researchers obtained good results.

REFERENCES

[1] R. Morris, "Workshop Summary Kasparov vs. Big Blue: The Significance for AI", *ICGA Journal*, vol. 20, no. 3, pp. 204-206, 1997, doi: https://doi.org/10.3233/ICG-1997-20315.

[2] M. Buro, "Call for AI research in RTS games", in *Proc. of the AAAI Workshop on AI in Games*, 2004, pp. 139-141.

[3] A. Shantia, E. Begue and M. Wiering, "Connectionist reinforcement learning for intelligent unit micro management in StarCraft", in *Proc. of International Joint Conference on Neural Networks*, 2011, pp. 1794-1801, doi: https://doi.org/10.1109/IJCNN.2011.6033442.

[4] L. Amado, "Reinforcement learning applied to RTS games", in *Proc. of the 2017 Workshop on Adaptive Learning Agents*, 2017, pp. 1-8.

[5] N. Remi, J. Krebbers, M. Drugan, and M. Wiering, "Hierarchical Reinforcement Learning for Real-Time Strategy Games", in *Proc. of International Conference on Agents and AI*, 2018, pp. 470-477, doi: https://doi.org/10.5220/0006593804700477.

[6] A. Aamodt, "Case-based reasoning for improved micromanagement in Real-time strategy games", in *Proc. of the Workshop on Case-Based Reasoning for Computer Games*, 2009, pp. 1-11.

[7] M. Molineaux, D. W. Aha, and P. Moore, "Learning Continuous Action Models in a Real-Time Strategy Environment", in *Proc. of the Twenty-First International Florida AI Research Society Conference*, Palo Alto, 2008, pp. 257-262.

[8] F. Sailer, M. Buro, and M. Lanctot, "Adversarial Planning Through Strategy Simulation", in *Proc. of the IEEE Conference on Computational Intelligence and Games*, Piscataway, 2007, pp. 80-87, doi: https://doi.org/10.1109/CIG.2007.368082.

[9] D. Churchill, A. Saffidine, and M. Buro, "Fast Heuristic Search for RTS Game Combat Scenarios", in *Proc. of the Eighth AAAI Conference on AI and Interactive Digital Entertainment*, Palo Alto, 2012, pp. 112-117, doi: https://doi.org/10.5555/3014629.3014650.

[10] G. Synnaeve, and P. Bessière, "A Bayesian Model for RTS Units Control Applied to StarCraft", in *Proc. of the IEEE Conference on Computational Intelligence and Games*, Piscataway, 2011, pp. 190-196, doi: https://doi.org/10.1109/CIG.2011.6032006.

[11] I. Gabriel, V. Negru, and D. Zaharie, "Neuroevolution Based MultiAgent System for Micromanagement in Real-Time Strategy Games", in *Proc. of the Fifth Balkan Conference in Informatics*, New York, 2012, pp. 32-39, doi: https://doi.org/10.1145/2371316. 2371324.

[12] M. Stanescu, N. A. Barriga, A. Hess, and M. Buro, "Evaluating real-time strategy game states using convolutional neural networks", in *Proc. of IEEE Conference on Computational Intelligence and Games*, Santorini, 2016, pp. 1-7, doi: https://doi.org/10.1109/CIG.2016. 7860439.

[13] J. Huang, and W. Yang, "A multi-size convolution neural network for RTS games winner prediction", in *Proc. of MATEC Web of Conferences*, vol. 232, 2018, pp. 1-5, doi: https://doi.org/ 10.1051/matecconf/201823201054.

[14] D. Aha, M. Molineaux, and M. Ponsen, "Learning to Win: Case-Based Plan Selection in a Real-Time Strategy Game", in *Proc. of ICCBR 2005: Case-Based Reasoning Research and Development*, 2005, pp. 5-20, doi: https://doi.org/10.1007/11536406_4.

[15] P. Cadena, and L. Garrido, "Fuzzy Case-Based Reasoning for Managing Strategic and Tactical Reasoning in StarCraft", in *Proc. of MICAI 2011: Advances in AI*, Mexico, 2011, pp. 113-124, doi: https://doi.org/10.1007/978-3-642-25324-9_10.

[16] H. Muñoz-Avila, and D. Aha, "On the Role of Explanation for Hierarchical Case-Based Planning in Real-Time Strategy Games", in *Proc. of 7th European Conference Advances in Case-Based Reasoning*, Lecture Notes in Computer Science, Berlin, 2004, pp. 1-10.

[17] J. Laagland, "A HTN Planner for a Real-Time Strategy Game", 2008, pp. 1-10. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.406.8722&rep=rep1& type= pdf. Accessed on: June 12, 2021.

[18] B. Weber, and S. Ontañón, "Using Automated Replay Annotation for Case-Based Planning in Games", in *Proc. of 8th International Conference on Case-Based Reasoning*, Seattle, 2010, pp. 1-10.
[19] M. Molineaux, M. Klenk, and D. Aha, "Goal-Driven Autonomy in a Navy Strategy Simulation", in *Proc. of the Twenty-Fourth AAAI Conference on AI*, Palo Alto, 2010, pp. 1548-1554.
[20] B. Weber, M. Mateas, and A. Jhala, "Applying Goal-Driven Autonomy to StarCraft", in *Proc. of the Sixth AAAI Conference on AI and Interactive Digital Entertainment*, Palo Alto, 2010, pp. 101-106, doi: https://doi.org/10.5555/3014666.3014685.
[21] F. Kabanza, P. Bellefeuille, F. Bisson, A. Benaskeur, and H. Irandoust, "Opponent Behaviour Recognition for Real-Time Strategy Games", in *Proc. of Plan, Activity, and Intent Recognition: Papers from the AAAI Workshop. Technical Report WS-10-15*, Palo Alto, 2010, pp. 29-36.
[22] B. Weber, and M. Mateas, "A Data Mining Approach to Strategy Prediction", in *Proc. of the 2009 IEEE Symposium on Computational Intelligence and Games*, Piscataway, 2009, pp. 140-147, doi: https://doi.org/10.1109/CIG.2009.5286483.
[23] E. Dereszynski, J. Hostetler, A. Fern, T. Dietterich, T. Hoang, and M. Udarbe, "Learning Probabilistic Behavior Models in Real-Time Strategy Games", in *Proc. of the Seventh AAAI Conference on AI and Interactive Digital Entertainment*, Palo Alto, 2011, pp. 20-25.
[24] E. Dereszynski, J. Hostetler, A. Fern, T. Dietterich, and T. Hoang, "Inferring Strategies from Limited Reconnaissance in Real-Time Strategy Games", in *Paper presented at the Conference on Unceertainty in AI*, Avalon, 2012, pp. 367-376.
[25] J. Zhu, J. Villareale, N. Javvaji, S. Risi, M. Löwe, R. Weigelt, and C. Harteveld, "Player-AI Interaction: What Neural Network Games Reveal About AI as Play", in *Proc. of CHI Conference on Human Factors in Computing Systems*, Yokohama, 2021, pp. 1-17, doi: https://doi.org/10.1145/3411764.3445307.

The article was received 01.06.2021.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

[1] R. Morris, "Workshop Summary Kasparov vs. Big Blue: The Significance for AI", *ICGA Journal*, vol. 20, no. 3, pp. 204-206, 1997, doi: https://doi.org/10.3233/ICG-1997-20315.
[2] M. Buro, "Call for AI research in RTS games", in *Proc. of the AAAI Workshop on AI in Games*, 2004, pp. 139-141.
[3] A. Shantia, E. Begue and M. Wiering, "Connectionist reinforcement learning for intelligent unit micro management in StarCraft", in *Proc. of International Joint Conference on Neural Networks*, 2011, pp. 1794-1801, doi: https://doi.org/10.1109/IJCNN.2011.6033442.
[4] L. Amado, "Reinforcement learning applied to RTS games", in *Proc. of the 2017 Workshop on Adaptive Learning Agents*, 2017, pp. 1-8.
[5] N. Remi, J. Krebbers, M. Drugan, and M. Wiering, "Hierarchical Reinforcement Learning for Real-Time Strategy Games", in *Proceedings of International Conference on Agents and AI*, 2018, pp. 470-477, doi: https://doi.org/10.5220/0006593804700477.
[6] A. Aamodt, "Case-based reasoning for improved micromanagement in Real-time strategy games", in *Proc. of the Workshop on Case-Based Reasoning for Computer Games*, 2009, pp. 1-11.
[7] M. Molineaux, D. W. Aha, and P. Moore, "Learning Continuous Action Models in a Real-Time Strategy Environment", in *Proc. of the Twenty-First International Florida AI Research Society Conference*, Palo Alto, 2008, pp. 257-262.
[8] F. Sailer, M. Buro, and M. Lanctot, "Adversarial Planning Through Strategy Simulation", in *Proc. of the IEEE Conference on Computational Intelligence and Games*, Piscataway, 2007, pp. 80-87, doi: https://doi.org/10.1109/CIG.2007.368082.

[9] D. Churchill, A. Saffidine, and M. Buro, "Fast Heuristic Search for RTS Game Combat Scenarios", in *Proc. of the Eighth AAAI Conference on AI and Interactive Digital Entertainment*, Palo Alto, 2012, pp. 112-117, doi: https://doi.org/10.5555/3014629.3014650.

[10] G. Synnaeve, and P. Bessière, "A Bayesian Model for RTS Units Control Applied to StarCraft", in *Proc. of the IEEE Conference on Computational Intelligence and Games*, Piscataway, 2011, pp. 190-196, doi: https://doi.org/10.1109/CIG.2011.6032006.

[11] I. Gabriel, V. Negru, and D. Zaharie, "Neuroevolution Based MultiAgent System for Micromanagement in Real-Time Strategy Games", in *Proc. of the Fifth Balkan Conference in Informatics*, New York, 2012, pp. 32-39, doi: https://doi.org/10.1145/2371316. 2371324.

[12] M. Stanescu, N. A. Barriga, A. Hess, and M. Buro, "Evaluating real-time strategy game states using convolutional neural networks", in *Proc. of IEEE Conference on Computational Intelligence and Games*, Santorini, 2016, pp. 1-7, doi: https://doi.org/10.1109/CIG.2016. 7860439.

[13] J. Huang, and W. Yang, "A multi-size convolution neural network for RTS games winner prediction", in *Proc. of MATEC Web of Conferences*, vol. 232, 2018, pp. 1-5, doi: https://doi.org/ 10.1051/matecconf/201823201054.

[14] D. Aha, M. Molineaux, and M. Ponsen, "Learning to Win: Case-Based Plan Selection in a Real-Time Strategy Game", in *Proc. of ICCBR 2005: Case-Based Reasoning Research and Development*, 2005, pp. 5-20, doi: https://doi.org/10.1007/11536406_4.

[15] P. Cadena, and L. Garrido, "Fuzzy Case-Based Reasoning for Managing Strategic and Tactical Reasoning in StarCraft", in *Proc. of MICAI 2011: Advances in AI*, Mexico, 2011, pp. 113-124, doi: https://doi.org/10.1007/978-3-642-25324-9_10.

[16] H. Muñoz-Avila, and D. Aha, "On the Role of Explanation for Hierarchical Case-Based Planning in Real-Time Strategy Games", in *Proc. of 7th European Conference Advances in Case-Based Reasoning,* Lecture Notes in Computer Science, Berlin, 2004, pp. 1-10.

[17] J. Laagland, "A HTN Planner for a Real-Time Strategy Game", 2008, pp. 1-10. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.406.8722&rep=rep1& type= pdf. Accessed on: June 12, 2021.

[18] B. Weber, and S. Ontañón, "Using Automated Replay Annotation for Case-Based Planning in Games", in *Proc. of 8th International Conference on Case-Based Reasoning*, Seattle, 2010, pp. 1-10.

[19] M. Molineaux, M. Klenk, and D. Aha, "Goal-Driven Autonomy in a Navy Strategy Simulation", in *Proc. of the Twenty-Fourth AAAI Conference on AI*, Palo Alto, 2010, pp. 1548-1554.

[20] B. Weber, M. Mateas, and A. Jhala, "Applying Goal-Driven Autonomy to StarCraft", in *Proc. of the Sixth AAAI Conference on AI and Interactive Digital Entertainment*, Palo Alto, 2010, pp. 101-106, doi: https://doi.org/10.5555/3014666.3014685.

[21] F. Kabanza, P. Bellefeuille, F. Bisson, A. Benaskeur, and H. Irandoust, "Opponent Behaviour Recognition for Real-Time Strategy Games", in *Proc. of Plan, Activity, and Intent Recognition: Papers from the AAAI Workshop. Technical Report WS-10-15*, Palo Alto, 2010, pp. 29-36.

[22] B. Weber, and M. Mateas, "A Data Mining Approach to Strategy Prediction", in *Proc. of the 2009 IEEE Symposium on Computational Intelligence and Games*, Piscataway, 2009, pp. 140-147, doi: https://doi.org/10.1109/CIG.2009.5286483.

[23] E. Dereszynski, J. Hostetler, A. Fern, T. Dietterich, T. Hoang, and M. Udarbe, "Learning Probabilistic Behavior Models in Real-Time Strategy Games", in *Proc. of the Seventh AAAI Conference on AI and Interactive Digital Entertainment*, Palo Alto, 2011, pp. 20-25.

[24] E. Dereszynski, J. Hostetler, A. Fern, T. Dietterich, and T. Hoang, "Inferring Strategies from Limited Reconnaissance in Real-Time Strategy Games", in *Paper presented at the Conference on Unceertainty in AI*, Avalon, 2012, pp. 367-376.

[25] J. Zhu, J. Villareale, N. Javvaji, S. Risi, M. Löwe, R. Weigelt, and C. Harteveld, "Player-AI Interaction: What Neural Network Games Reveal About AI as Play", in *Proc. of CHI Conference on Human Factors in Computing Systems*, Yokohama, 2021, pp. 1-17, doi: https://doi.org/10.1145/3411764.3445307.

ЯРОСЛАВ ДОРОГИЙ,
ОЛЕКСАНДР СВИРИДЕНКО

МЕТОДИ ТА ПІДХОДИ АНАЛІЗУ ПРОЄКТУВАННЯ ШТУЧНОГО ІНТЕЛЕКТУ ДЛЯ СТРАТЕГІЇ РЕАЛЬНОГО ЧАСУ

Дослідження містить детальний аналіз підходів до створення штучного інтелекту у відеоіграх. Основним напрямом досліджень є штучний інтелект для стратегій реального часу. Це обумовлено тим, що даний жанр характеризується складністю ігрового середовища та практикою створення комплексного штучного інтелекту, який складається з кількох агентів, відповідальних за певний аспект гри. Основними сферами використання методів штучного інтелекту в стратегіях є стратегічні та тактичні рішення, а також аналіз поточної ситуації та прогнозування дій противника та обраної ним стратегії. Серед проаналізованих підходів до тактичного штучного інтелекту найчастіше використовуються підкріплення, пошук дерева ігор, байєсівська модель, прецедентні рішення та нейронні мережі. Популярними підходами до побудови стратегічного штучного інтелекту є прийняття рішень на основі прецедентів, ієрархічне планування та автономне досягнення цілей. Тоді як при створенні модуля дослідження та визначення планів найпопулярнішими методами є дедуктивний, абдукційний, ймовірнісний та прецедентний. Крім розглянутих методів, у розробці використовуються й інші. Однак, їхня застосовність обмежується швидкістю або специфічністю реалізації. Цим ускладнюється адаптування до стандартних правил жанрових ігор. Наведено порівняння алгоритмів і реалізацій штучного інтелекту в рамках комерційних і наукових розробок. Серед основних відмінностей – висока вартість комерційної розробки комплексних агентів, а також специфіка наукового підходу, який спрямований на створення найбільш ефективного агента з точки зору якості гри, а не максимізацію позитивних вражень гравців, що є основою комерційного розвитку. Описано причини недостатньо активного розвитку наукових досліджень у галузі штучного інтелекту для ігор загалом та жанру стратегій реального часу зокрема.

Ключові слова: штучний інтелект, стратегії в реальному часі, відеоігри, нейронні мережі, тактичне прийняття рішень, стратегічне рішення.

Дорогий Ярослав Юрійович, доктор технічних наук, доцент, доцент кафедри автоматики і управління в технічних системах, Національний технічний університет України “Київський політехнічний інститут імені Ігоря Сікорського”, Київ, Україна, ORCID 0000-0003-3848-9852, argusyk@gmail.com.

Свириденко Олександр Андрійович, магістрант, Національний технічний університет України “Київський політехнічний інститут імені Ігоря Сікорського”, Київ, Україна, ORCID 0000-0003-0241-3485, sania.sviridenko@gmail.com.

Dorohyi Yaroslav, doctor of technical sciences, associate professor, associate professor at the automation and control in technical systems academic department, National technical university of Ukraine “Igor Sikorsky Kyiv polytechnic institute”, Kyiv, Ukraine.

Svyrydenko Oleksandr, master’s student, National technical university of Ukraine “Igor Sikorsky Kyiv polytechnic institute”, Kyiv, Ukraine.