Simulation of artificial intelligence in a computer game

Elena V Soboleva¹ and Nadezda V Shalaginova²

¹ Department of Digital Technologies in Education, Vyatka State University, Kirov, Russia
² Department of Applied Mathematics and Informatics, Vyatka State University, Kirov, Russia

E-mail: sobolevaelv@yandex.ru

Abstract. Artificial intelligence, implemented in the space of a computer game, will allow performing a variety of practice-oriented tasks and time-consuming algorithms: finding ways, avoiding obstacles, conducting combat, choosing the appropriate strategy, simulating emotions, dialogue, etc. The main problem that should be solved when developing an appropriate gaming application is to simulate the mechanism for controlling all possible functions of the game object. The research methodology includes: analysis of scientific works of Russian and foreign authors; the method of system information analysis, the method of mathematical and information modeling; computer experiment method, as a kind of computational; decision making algorithms. The paper presents a solution to the problem, including the informed choice of a decision-making algorithm, and the design of a model of a fuzzy finite automaton that combines the advantages of a model of a generating finite automaton and fuzzy logic. Based on the results of testing the implemented model, a conclusion was formulated about the main advantage of the artificial intelligence model for the gaming application - its ability to make an unusual decision in the arisen game situation.

1. Introduction

Artificial intelligence is an integral part of almost any computer game. It makes the game application more fun and interesting. These factors, of course, support the relevance of this research. From a scientific point of view, artificial intelligence is a complex cybernetic computer-software-hardware system (electronic, including virtual, electronic-mechanical, bio-electronic-mechanical or hybrid) with a cognitive-functional architecture and its own or relevantly accessible (attached a) computing power of the necessary capacities and speed [1]. As a field of science, artificial intelligence is very extensive and aims at a huge number of tasks, such as pattern recognition, data analysis, theorem proof, machine translation and understanding of human speech, and many others, even trying to create real intelligence by artificial means [2]. For example, in the project “Kismet” of the Massachusetts Institute of Technology, an attempt is made to create artificial intelligence capable of learning, social interaction, and also the manifestation of emotions [3]. In terms of gaming applications, the above goes beyond the entertainment project.

2. Materials and methods

Analysis of the scientific literature on the problem of research, allowed to identify the basic concepts, methods and algorithms that are necessary to achieve the goal.
Gaming artificial intelligence means imitation of reasonable behavior in order to create a more interesting and believable game for a person. Animat (gaming agent) - a term used to define creatures living in virtual environments [4].

In any game, artificial intelligence has a set of actions (strategies) that are written in the form of scripts. A script is a program that automates some task. [5].

A game object controlled by artificial intelligence must perform various tasks. Each of these tasks is implemented by a specific script. The actions performed in the game can be divided into groups: movement in the game space; the battle; interaction with the world.

Moving around the game world is one of the most important parts of a game application. Many games are associated with the movement of characters around the map, without the realization of which the game loses its meaning [6].

One of the main tasks of traffic planning is the shortest path problem - finding the shortest path located between two points (vertices) on a graph. Graph theory refers the task of finding the shortest path to the most important classical problems. Therefore, at present there are many algorithms aimed at solving this problem. Next, the shortest path search algorithm A* will be considered.

A* – a search algorithm that searches for the lowest cost route from the initial vertex of the graph to the selected final vertex. [7].

In the process of the algorithm for the vertices is calculated function

\[ f(v) = g(v) + h(v), \]

where

\[ g(v) \] – least cost of the path in v from the starting vertex,
\[ h(v) \] – heuristic approximation of the cost of the path from v to the final goal [8].

In fact, the function \( f(v) \) is the path length to the target, which is made up of the distance traveled \( g(v) \) and an approximate estimate of the remaining distance \( h(v) \). The smaller the value of \( f(v) \), the earlier the vertex v will be viewed, since through it, presumably, the distance to the target is reached the fastest. Vertices opened by the algorithm stored in a queue with priority on the value of \( f(v) \) [8].

A* acts like the Dijkstra algorithm and looks at all the routes leading to the target first, which are the best, due to the available information (heuristic function) [9].

The behavior of the algorithm depends strongly on the heuristic function. In turn, the choice of heuristics depends on the formulation of the problem. It is often used to simulate the movement of a surface covered with a coordinate grid [4].

If the movement is possible only in four directions, then the Manhattan distance should be chosen as a heuristic:

\[ h(v) = |v.x - t.x| + |v.y - t.y|, \]

where \( t \) – end point.

Chebyshev distance is applied when diagonals are added to four directions:

\[ h(v) = \max(|v.x - t.x|, |v.y - t.y|). \]

If the movement is not limited to the grid, then it is possible to use the Euclidean distance in a straight line.

Depending on the game situation, artificial intelligence must choose the appropriate strategy. Decision algorithms perform this task.

3. Results

In this paper, the authors selected and studied various decision-making algorithms in games (decision trees, state machines, fuzzy logic), which made it possible to formulate the following conclusions:

- decision trees use a recursive algorithm, which will have a negative impact on the computational complexity of the application;
• state machines have many advantages, but also a disadvantage associated with over-robotic animat control;
• fuzzy logic allows the modeling of artificial intelligence closer to human thinking, in connection with what the model of fuzzy finite state machine that combines the advantages of the model generating finite automata and fuzzy logic has been studied and selected to implement.

Developed game application is focused on the mobile platform. The programming language is Java (as the official programming language on Android). Android Studio was chosen as a development environment. The Java framework Libgdx is selected to implement the game structure and work with graphics.

A 2d RPG game application will be implemented to test the game artificial intelligence model. RPG (RPG «Role-Playing Game») is a genre of computer games, in which the basis of the gameplay is the “living” of a certain role. The player takes control of the hero or heroine, with a set of standard characteristics and skills. The game contains animats interacting with the player in various ways provided by the plot of the game, for example, a battle.

In developing the model, a list of the characteristics and actions of the characters was compiled. A set of characteristics of game characters: level of health (Hp); energy level (Energy); protection level (Protection); intelligence level (intellect); melee attack power (DDstrength); ranged attack power (RDDstrength). All game characters will have this feature set.

The minimum set of gaming agent strategies includes: goal pursuit; melee attack; ranged attack; escape from the goal; target search; inaction. In addition, for a greater variety of game strategies, each  character has unique characteristics, which are called skills. Skills allow characters to own multiple variations of each strategy. Skills may vary in their strength and value.

Each character strategy implements a specific script. Strategies will be chosen, given the output values of the fuzzy finite state machine.

3.1. Description of the work of a fuzzy state machine

• Running of the method of the main logic module. Input parameter - game statistics. Statistics is the input variable set for a fuzzy state machine. After the update, the fuzzification of clear values of game statistics takes place to obtain a set of fuzzy variables (α).
• Updating of a fuzzy set of game states (δ). There is a processing of the rules in which a lot of fuzzy variables of previous states and updated information about the world participate.
• Formation of a set of fuzzy inference (β), which implements the method of working with the rules for obtaining a fuzzy inference. The rules take into account the updated set of states of a fuzzy finite automaton and the input set of fuzzy variables derived from game statistics. It consists of the stages of aggregation, activation and accumulation. After the completion of the third stage, the algorithm returns a set of output fuzzy variables corresponding to game strategies.

3.2. An example of the work of artificial intelligence
Fuzzy states: {fight, fallback, calm}.
Input set: {hp, enemyhp, inflictedDMG, receivedDMG, distance, energy}. Each of these linguistic variables consists of 3 fuzzy variables {low..., mid..., hurt...}, denoting the degree of the statistics element.
Output set: {dd, rdd, follow, heal, run, donothing}
The test model will be an aggressive gaming agent who will attack the player whenever possible.

3.2.1. Rules for states. The transition to a state of combat occurs:
not highdistance (when approaching animat);
not lowhurt, not lowhealth, not lowhurt (when causing damage to the animat, not a small amount of remaining health and not a small amount of damage caused to the animat);
lowenemyhp (if the enemy has low health).

The transition to a state of escape occurs:

- not highdistance (when approaching animat);
- lowhealth (with a small amount of health animat).

The transition to a state of exploration of the world occurs:

- highdistance, highhealth (with a large distance from the enemy and a large number of health animat).

3.2.2. Rules for the output set of strategies. Melee attack strategy:

- fight, lowdistance (combat state and close range).

Ranged attack strategy:

- fight, middistance, not lowenergy (combat state, average distance and not a small level of energy).

Pursuit strategy:

- fight, not lowhealth, not lowdistance, not lowenergy (state of combat, not a little health, not close distance and not enough energy).

Treatment strategy:

- fallback, not lowdistance, not lowhealth (state of retreat, not close distance and low health);
- calm, not highhealth (state of calm and not much health).

Escape strategy:

- fallback, lowhealth, not highdistance, highhurt (state of retreat, low health, not great distance, a lot of damage);

World exploration strategy:

- calm, highdistance (state of calm, long distance)

3.2.3. Description of the behavior of the animat according to the rules. Approaching the animat:

- if the distance is greater than the distance attack, the animat starts pursuit;
- when approaching a ranged attack distance, the animat begins to attack;
- after the energy ends, the animat runs up to a close range of attack and attacks;
- if you run too far away, the animat starts stalking and then attacks again;
- with a small amount of health and close proximity of the enemy, the animat begins to retreat;
• if the animat has retreated for some distance, then it begins treatment and then again attacks the enemy;
• if the enemy runs away for a long distance, the animat loses it from sight and goes into a state of calm;
• animat can change behavior, slightly deviating from the same sequence of actions.

4. The discussion of the results
During the study, game artificial intelligence was designed and implemented. The structure of artificial intelligence can be divided into 3 modules: the character description module; decision making module; hierarchy of game objects.

The decision module is a fuzzy generating state machine. The input fuzzy set is made up of game statistics, the set of states consists of the combat state, the retreat state and the world exploration state. Output fuzzy sets are estimates of strategies. The highest score is chosen. The class of Mob gaming agent uses a decision module for the implementation of mental activity. To test the implemented artificial intelligence, an RPG game application was implemented.

A test set of rules for the behavior of the animat has been spelled out in detail to test the possibility of tuning and the correctness of the model. One example of testing is presented in the results.

Testing of the application has shown the adequacy of the compiled model of game artificial intelligence. The behavior of the animat in various game situations looks logical and corresponds to the intended result. Transitions between states and animat strategies are carried out smoothly and do not look too robotic. There is a variety of actions and a slight deviation from the constantly repeating sequence.

5. Conclusion
The study focused on the main actions performed by the game artificial intelligence. The authors reviewed the existing methods of decision making in games and reasonably chose an implementation method based on a fuzzy generating state machine. With the help of the selected method, it was possible to design a model of game artificial intelligence. The model was implemented and tested on the example of a basic set of states, strategies, and rules that set hostile behavior.

Testing has shown the performance and adequacy of the implemented game artificial intelligence. With the help of sets operating with a fuzzy state machine, it is possible to set different types of behavior for the game agent, implementing various game agents in one game and without changing the structure of the algorithm.

The resulting adoption module can be used in various games, which makes it a flexible tool for implementing artificial intelligence.

Acknowledgments
This research is funded by Russian Science Foundation under Grant No 18-78-10053.

References
[1] Makryniotis T 2014 Game design with the aid of computer semiotics Journal of Gaming and Virtual Worlds 6(3) 255–73
[2] Browne C and Maire F 2010 Evolutionary game design IEEE Transactions on Computational Intelligence and AI in Games 2 (1) 1–16
[3] Brundage M 2014 Limitations and risks of machine ethics Journal of Experimental and Theoretical Artificial Intelligence 26(3) 355–72
[4] Bryson J J and Kime P P 2011 Just an artifact: Why machines are perceived as moral agents IJCAI International Joint Conference on Artificial Intelligence 1641–46
[5] Soboleva E V, Galimova E G, Maydangalieva Z A, and Batchayeva K K 2018 Didactic value of gamification tools for teaching modeling as a method of learning and cognitive activity at school Eurasia Journal of Mathematics, Science and Technology Education 14(6) 2427–44
[6] Umarov I and Mozgovoy M 2014 Creating believable and effective AI agents for games and simulations: Reviews and case study *Contemporary Advancements in Information Technology Development in Dynamic Environments* 25

[7] Pirovano M and Lanzi P L 2014 Fuzzy tactics: A scripting game that leverages fuzzy logic as an engaging game mechanic *Expert Systems with Applications* 41 (13) 6029–38

[8] Hernández-Orallo J, Martínez-Plumed F, Schmid U, Siebers M and Dowe D L 2016 Computer models solving intelligence test problems: Progress and implications *Artificial Intelligence* 230 74–107

[9] Lukov S V 2017 Artificial intelligence and cyberspace *Horizons of humanitarian knowledge* 2