Smart Events in Behavior of Non-player characters in Computer Games

Marcin Zieliński[1], Piotr Napieralski[1][0000−0003−1427−7791], Marcin Daszuta[1], and Dominik Szajerman[1][0000−0002−4316−5310]

Institute of Information Technology, Lodz University of Technology, Łódź, Poland
dominik.szajerman@p.lodz.pl

Abstract. This work contains a solution improvement for Smart Events, which are one of the ways to guide the behavior of NPCs in computer games. The improvement consists of three aspects: introducing the possibility of group actions by agents, i.e. cooperation between them, extending the SE with the possibility of handling ordinary events not only emergency, and introducing the possibility of taking random (but predetermined) actions as part of participation in the event.
In addition, two event scenarios were presented that allowed the Smart Events operation to be examined. The study consists of comparing the performance of the SE with another well-known algorithm (FSM) and of comparing different runs of the same event determined by the improved algorithm.
Comparing the performance required proposing measures that would allow for the presentation of quantitative differences between the runs of different algorithms or the same algorithm in different runs. Three were proposed: time needed by the AI subsystem in one simulation frame, the number of decisions in the frame, and the number of frames per second of simulation.

Keywords: Smart Events · NPC behavior · Behavior trees

1 Introduction

Computer games are a special area of use of artificial intelligence. Interesting cases are open-world games, where the player can move freely in an environment inhabited by numerous NPCs. Often one of the main elements there are interactions with these NPCs. Typically in the open-world games such as "Grand Theft Auto V" or "Elex", most of the NPC logic is related to the use of artificial intelligence in combat. Less attention is paid to the behaviors related to their daily life, in situations when the player does not interact with them in any way. However, it is thanks to AI outside of combat that the game world seems more alive in the eyes of the player. One of the reasons for this is the high complexity when an increased number of NPCs/agents have many different behaviors depending on the context. In this situation, the game software is complicated.

This work focuses on one of the most frequently chosen directions in the design of artificial intelligence systems in games, and more specifically the "Smart"

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77977-1_13

approach. A proposal was presented to expand the "Smart Events" solution, allowing for the elimination of its disadvantages.

2 Related work

Embedding logic within the environment has become a well-accepted solution for managing the complexity of behaviors in both science and industry. The main example of this is Smart Objects (SO) introduced by Kallmann and Thalmann[7]. They are widely used in the computer game industry in a simplified form. The SO stores the information and is responsible for positioning and playing the animation of the character that uses it. An example of SO could be a lever object in a game scene. The character does not need to know what this object is. To change its state, it simply performs the action of using SO, and the lever object itself takes care of its positioning and animation. As a result, there can be many different types of SO in a scene and the character only needs one action to use them all properly. The disadvantage of this solution is the lack of support for interrupting the behavior of characters and no possibility of nesting them.

A solution close to SO, eliminating this problem, was implemented by the creators of "The Sims 4" game [6]. The gameplay, in simple terms, consists of selecting the appropriate SO in order to interact with them for the character under consideration, and thus to satisfy his needs. The character performs actions of his own choosing (low priority), as well as those ordered by the player (high priority). Interactions with SO can consist of smaller blocks that are non-breakable. However, it is possible to break all interactions that consist of multiple blocks.

In the game "Bioshock: Infinite" the developers used SO while modeling the behavior of the player's assistant – Elizabeth [5]. SO are responsible for her behavior. The designers filled the game world with invisible tags that allow Elizabeth to draw attention to individual places. Thanks to this, her character can move around the area, finding elements of interest to her and interacting with them.

Another way of using SO was proposed in the game "S.T.A.L.K.E.R."[4]. The most important feature of an artificial life is that each NPC has a life of their own. In order to achieve this, the designers provided the characters with the opportunity to move between the different levels in the game, as well as remember the information obtained during their existence. In order to diversify the behavior of NPCs and enliven the game world, the designers implemented a SO development solution, called by them "Smart Terrains". ST are responsible for assigning tasks and behaviors to individual characters located in the area defined by them. This allowed for the addition of fractions bases, characters sitting together by bonfires, etc. to the game.

A similar approach was proposed in [2]. The authors noticed the need to create a solution supporting the creation of complex NPC behaviors. Inspired by research in the field of crowd simulation, they adapted it for use in games and implemented a solution called Smart Areas. Objects of this type contain

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77977-1_13

the logic responsible for deciding what behavior a NPC should adopt while in their area. In addition, it is possible to determine: how many agents can take a specific behavior at a given moment, what are the conditions imposed on an agent before taking an action, what actions are to be performed during various events (e.g. leaving the area/entering the Smart Area).

The authors of [11] proposed their solution in the form of the so-called Smart Events. It allows to simulate the reaction of a character to emergency events. The aim of the creators was to maintain a low demand for computing power and good scalability of the solution. To achieve this, they moved the logic responsible for choosing the behavior of a given character from the object representing it to an instance of an event. In order to differentiate behaviors for many agents, the authors referred to the fact that a person has different social roles defined by, for example, age, profession, gender or family relationships. In a given situation, a person can only assume one of their roles. The event object stores information about the behavior that an individual should adopt depending on the state of the event (e.g. the extent of fire spread) and its current major social role. This allows an individual to make a simple decision about their choice of behavior.

As can be seen, the AI behavior of agents in computer games can be solved in various ways. Another group are methods based on the agents' memories and even their emotions [9].

2.1 Primed Agent

Primed agent is a concept proposed in [11]. Based on research in the field of psychology, they introduced the so-called priming, i.e. putting one of the cultural concepts of himself in front of another. Such a change may occur in response to the language or context of the conversation. A man may have many concepts, including conflicting ones, but at a given moment only one of them becomes active. Expanding the proposals contained in the research, they distinguished eight features that are considered in the priming process (Table 1).

Table 1. Features considered in the priming process together with their representative values [11].

Trait	Possible values
age	child, adult, elder
gender	male, female
ethnicity	American, European, Asian, African, Australian, Hispanic
religion	Christian, Jewish, Muslim, Hindu, Buddhist
vocation	firefighter, policeman, teacher, student
relational	mother/father, daughter/son, husband/wife, friend/stranger
political affiliation	democrat, republican

Each agent has its own trait values and can have more than one trait value. Besides, some of them may have fields with additional data, e.g. for a parent

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77977-1_13

they are the agent's children. Any of the traits can be brought to the fore. Both in [11] and in our solution, the priming process has been limited to appearing only when entering a location and when interacting with another agent.

2.2 Smart Events

The main idea behind the concept of Smart Events is to transfer the logic related to the reaction of agents to an event to the object representing this event [11]. This solution is an extension of the Smart Objects concept, in which the object informs the agent about how it should be used by this agent [7]. The developers defined the event as a planned or externally triggered addition or removal of a fact from the world model.

The Smart Event is represented by the following parameters: type, position, location, start time, end time, evolution (finite state machine), influence region, participants, event emergency level, corresponding (available) actions.

Communication between Smart Event and agents is carried out via the message board. The board is responsible for broadcasting and updating information about the evolution of the event. When an event starts, its information is sent to message boards according to the affected region. Then each board informs its subscribers about the event. Agents can be assigned to it statically – regardless of their location, or dynamically – by signing up when entering the board operation region and unsubscribing from it while leaving. An agent can choose whether he wants to react to the event he was informed about by comparing the level of emergency of the action he is currently performing to the emergency level of the event. If the current action turns out to be less important, it will ask the board to assign an appropriate, new action. Otherwise, the event will be ignored by him.

Smart Events may change over time. Associated agent's actions should also adapt to its development. FSM is used to model the evolution of an event. As it develops, the event will inform relevant message boards about changes in its status, including: emergency level, region of influence, and corresponding actions.

3 Method

Testing agent behavior algorithms in games is not easy. It is quite difficult to assess their plausibility, because a set of comparative data on human behavior may not be possible to prepare. Also, the sheer complexity of behaviors, changes, transitions between them, their pace does not facilitate such analyzes. At the same time, the examples should be relatively simple so that the mixing of various behaviors and various factors does not make observation impossible.

During this work, several tests were prepared consisting of scenes influencing the course of various elementary behaviors of agents. For the purposes of the presentation, two scenarios have been selected that cover the most representative and interesting cases possible. Elementary behaviors and their changes presented

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77977-1_13

in them can be adapted to other situations. In this way, you can generalize the solution.

This section presents how Smart Events has been improved in this work. Then two scenarios to test it and measures that were used to compare the performance of the solutions were presented.

In their article, the creators of Behavior Objects described the lack of support for coordinating actions performed by agents as the main disadvantage of Smart Events. Another problem for them turned out to be that for each set of character traits, an event provides the same behavior [1].

3.1 Finite State Machine

FSM is a well-known algorithm used in computer games AI. It is used here for comparisons with the Smart Events method. An important element of the structure used is the message system. It allows agent to send immediate as well as delayed messages to a given machine. After receiving the message, it forwards it to the current state that can handle it based on the content it contains. Thanks to this solution, it is possible to send signals that may affect the operation of agents. Figure 1 presents an example of the firefighter's FSM.

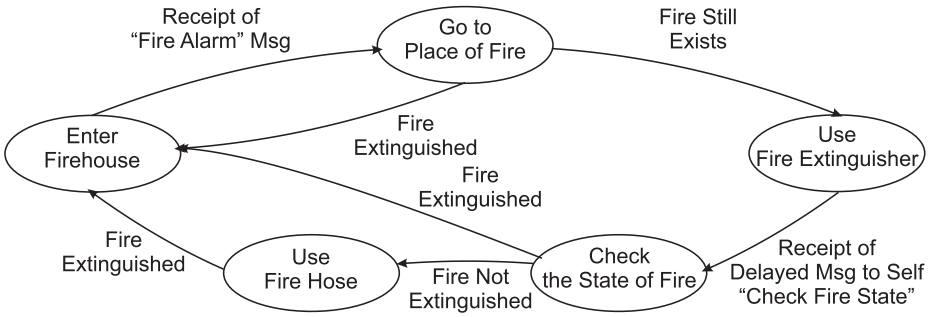

Fig. 1. An example of the firefighter's FSM.

3.2 The proposed solution

The extension of the Smart Events solution proposed in this chapter aims to eliminate the above-mentioned disadvantages. The task of the simulation using this extension is to show their improvement and to show the possibility of using the model in a non-emergency scenario.

The proposed development is largely based on parameterized behavior trees, which allows to eliminate the problem of the lack of coordination of agents' actions.

The improvement consists in replacing the action table, communication via the message board, and agent priming with the event behavior tree. When an

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77977-1_13

event occurs, the agents taking part in it cease to perform their actions and are deprived of their autonomy. The event behavior tree takes control over them. The leaves of this tree represent actions performed by controlled agents.

The Smart Events extension proposed in this chapter uses the event behavior tree, but leaves its foundations intact only by extending it. The structure of the event object has been extended to include a set of related group actions. The group behavior table differs from the basic one by adding a column specifying the role in which the agent can act in a given situation.

The very process of assigning of action to an agent has changed. When an event occurs, the agent is assigned behavior according to the rules known from Smart Events. However, when all agents participating in the event are informed about it and receive selected actions, the event starts an attempt to perform a group action. For this purpose, it looks for the behavior in the table for which all roles can be filled by its participants. In addition to meeting the conditions in the table, in order for an agent to be taken into account in assigning roles, the importance of the previously assigned individual action must be lower than or equal to the emergency level of the event. This solution ensures that the agent chooses the action that is more important at a given moment, regardless of its type.

The second problem – a small variety of selected actions, was taken into account in the simulation when designing behavior trees. In addition to standard control nodes, a random selection node was used that allows to perform an action drawn from a given set. In the second simulation scenario presented in section 3.4, it allowed to diversify the behavior of the shopkeeper while working at the cash register, and the customer while looking around the store. Another issue that helps to avoid low variety is the random success node shown in the behavior tree. It returns a positive execution status with a given probability, and thus decides about interrupting or further executing the actions that follow. It was used during the haggling of agents in order to simulate the decisions about lowering the price made by the shopkeeper.

3.3 Experiment: The "School" scenario

The tests performed in this scenario are designed to compare two AI methods in the same task. They are aimed at using all elements of the Smart Events solution. The tests performed with the use of the FSM method have been developed in such a way as to obtain runs as similar as possible to Smart Events.

During the simulation, an emergency event scenario in the form of a fire at school was used, created by the authors of the Smart Events model [11]. It uses of all the model's most important elements, which translates into the possibility of a good comparison of both tested solutions.

The structure of the world model was prepared in a way that allows to reproduce the course of the presented scenario. It has the necessary components, such as the fire department, police building and school, as well as all the agents used. The designed section of the city, which is a model of the simulation world, consists of three buildings (Fig. 2):

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77977-1_13

– fire brigade – a place where a firefighter is, equipped with a desk, fire extinguisher and an alarm bell,
– police station – place where a policeman is, equipped with a desk and an alarm bell,
– school – the place where the teacher works and where the pupil is, mother and child go to this building. It consists of a hall, bathroom and one classroom. There are benches, a blackboard and a bucket in the classroom. This is where the fire breaks out. The bathroom is equipped with a towel and a tap with a sink. In the hall there is a fire extinguisher, a fire hose and an alarm bell.

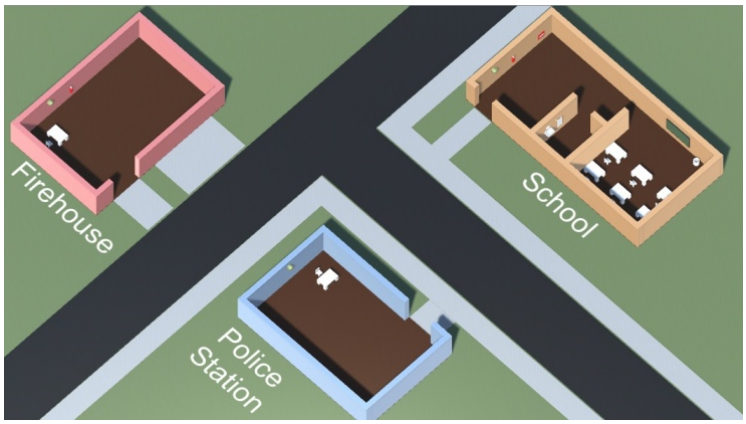

Fig. 2. Simulation world model with buildings listed: fire station, police station, and school.

In the simulation scenario, there were six agents with different roles:

1. Firefighter – Occupation: Fireman, Workplace: Fire Department Building, Initial Behavior: Sit at the desk.
2. Policeman – Occupation: Policeman, Workplace: Police Station, Initial Behavior: Sit at Desk.
3. Teacher – Occupation: Teacher, Workplace: School Building, Initial Behavior: Teach at the blackboard.
4. Student – Occupation: Student, Workplace: School Building, Initial Behavior: Sit at the School Desk.
5. Son – Relationship: Agent "Mothers's" Son, Initial Behavior: Follow Mother.
6. Mother – account: agent "Son's" mother, initial behavior: take her son to school.

Agents perform their initial actions until a fire breaks out. When an event begins, they begin to implement behaviors related to it. All actions occurring during a fire along with the conditions under which they occur are presented in the table shown in Figure 3.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77977-1_13

Trait	Value	Priming	EmergencyLevel	Action
= **Vocation**	Teacher	Enter(School)	Low	PourWater ()
= **Vocation**	Teacher	Enter(School)	Medium	ExtinguishFire ()
= **Vocation**	Teacher	Enter(School)	High	LeadAwayFromSchool ()
= **Vocation**	Student	Enter(School)	Low	StareAtFire ()
= **Vocation**	Student	Enter(School)	Medium	FollowLeader ()
= **Vocation**	Student	Enter(School)	High	FollowLeader ()
= **Vocation**	Firefighter	Enter(emergency)	Low	Smoother ()
= **Vocation**	Firefighter	Enter(emergency)	Medium	ExtinguishFire ()
= **Vocation**	Firefighter	Enter(emergency)	High	FightFire ()
= **Vocation**	Policeman	Enter(emergency)	Low	Smoother ()
= **Vocation**	Policeman	Enter(emergency)	Medium	ExtinguishFire ()
= **Vocation**	Policeman	Enter(emergency)	High	ManageCrowd ()
= **Relational**	Son	Interact(myParent)	Low	StareAtFire ()
= **Relational**	Son	Interact(myParent)	Medium	FollowLeader ()
= **Relational**	Son	Interact(myParent)	High	FollowLeader ()
= **Relational**	Mother	Interact(myChild)	Low	Smoother ()
= **Relational**	Mother	Interact(myChild)	Medium	LeadAwayFromSchool ()
= **Relational**	Mother	Interact(myChild)	High	LeadAwayFromSchool ()

Fig. 3. The action table used during the fire in the simulations.

3.4 Experiment: The "Shop" scenario

The purpose of this scenario was to explore the new possibilities offered by the improvement of the SE system.

The simulation scenario was inspired by [10] where the parameterization of behavior trees is considered. The situation of a customer coming to the store and trying to buy goods at a satisfactory price is presented there. A similar scenario takes place in the case of the simulation in this work.

Upon entering the building, the customer looks around him looking for interesting products. Then he walks over to the shopkeeper and he is trying to haggle a lower price for the pot, which he wants to buy. The shopkeeper may agree to a discount by reducing the price by a given value. If the customer is satisfied with the price, he agrees to it, then takes the pot and leaves the store. Otherwise, the customer tries to persuade the shopkeeper to reduce the price. If the shopkeeper does not agree to the discount, the customer looks around again.

The location used in the simulation shows the shop building. It is a shopkeeper's workplace, equipped with shelves with products, a counter and pedestals on which pots stand. The simulation world model is presented in Figure 4.

During the test, two simulation runs with a different value of the random seed were performed. This allowed to show the differences in the actions performed by agents, and thus draw attention to their variety.

At the very beginning of the simulation, both agents enter the store to adopt their initial behavior. The customer goes to the store shelves downstairs and starts looking around. He had a choice of three places in the store. The one in

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77977-1_13

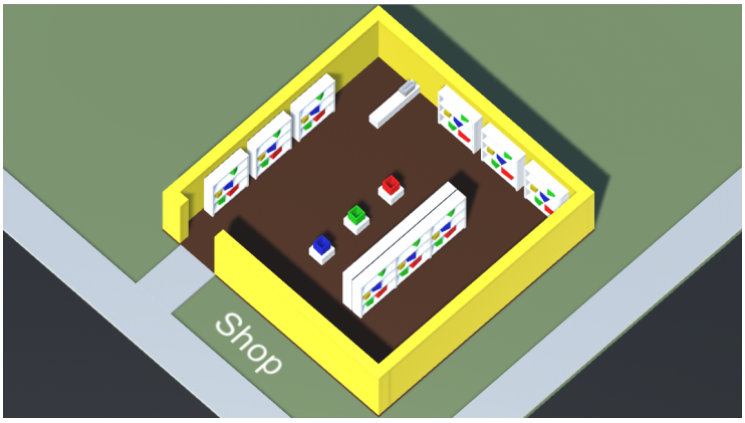

Fig. 4. A simulation world model that is a store where the scenario takes place.

which it performs this action has been selected using the previously mentioned random selection node. At that time, the shopkeeper is standing at the counter. From time to time, he whistles and sings (shown as text). His actions are similarly selected randomly from the pool. When the customer completes his look action, he triggers a haggle event. As both agents meet the conditions to be placed as shopkeeper and customer, they stop performing their actions and the group event behavior tree takes control of them.

Then, the customer comes to the counter and the shopkeeper greets him. Since the price of the pot he wants to buy is 6 coins and his budget is 4, he begins to haggle. The customer asks for a price reduction, then the shopkeeper decides to refuse, and the event ends. Such a verdict was established by the shopkeeper using a random success node with a 50% chance of a successful action. When the event is over, the agents resume performing their individual actions.

The customer selects the same location again where he performs a look action. The shopkeeper, on the other hand, makes a random decision to do something on his own. After the end of his behavior, the customer starts the haggling event again. This time, however, he manages to get a lower price. As the shopkeeper lowered the price of the pot to 5 coins, the customer still cannot afford it. So he continues to try to get the lower price. The shopkeeper again agrees to the reduction, thus meeting the conditions of the customer. The customer then agrees to buy a pot, takes it from the store and leaves, ending the event and simulation.

3.5 Performance measures

Measurement data was taken every frame. In order to reduce the effect of the difference in the number of frames on the obtained results, they were given in the form of arithmetic means and maximum values.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77977-1_13

All tests were conducted using a computer with an Intel Core i7-2670QM quad-core processor with a base clock of 2.20GHz, Nvidia GeForce GT 540M graphics card and 8 GB of RAM.

The first of the performance measures was the time needed by the AI system in one simulation frame. A similar measurement method was used in the work of the authors of the artificial intelligence system, which was ultimately to be used in a high-budget computer game with an open world [8].

Another measured value is the number of decisions the AI system has to make in the frame, and also completely throughout the simulation. The authors of the Smart Events solution used a similar comparison in their article, but without giving specific numbers [11]. The total number of decisions taken for the same scenario is also a value independent of any offsets of agents' execution, which makes it good for performance comparisons.

The last value tested in the experiment is the number of frames per second achieved during the simulation. Measurements of this type were carried out for several solutions with different numbers of agents belonging to the crowd in the simulations conducted in [3].

4 Results and discussion

4.1 Comparison to FSM

As previously mentioned, it is difficult to directly assess the believability (realism) of agent's behavior. However, the proposed solution you can be compared with the one widely used in games, e.g. FSM.

Figure 5 shows comparison of the run of the "School" scenario according to the FSM and SE methods:

1. top-left: SE-driven simulation after the first twenty seconds,
2. top-right: SE-driven simulation after the first sixty seconds,
3. bottom-left: FSM-driven simulation after the first twenty seconds,
4. bottom-right: FSM-driven simulation after the first sixty seconds.

The two methods are compared by putting together the paths that the agents travel through the event sequence. For each agent and each time instant t_i, the distance (in meters) between its locations (p_i) was calculated in both simulations: FSM and SE (eq. 1). The average and maximum distances calculated for the agents participating in the scenario are presented in Table 2. As can be seen, the measured values also confirm a very high similarity of behaviors.

$$d(t_i) = ||\boldsymbol{p_i^{FSM}}(t_i) - \boldsymbol{p_i^{SE}}(t_i)|| \tag{1}$$

Performance tests were conducted using measures introduced in subsection 3.5.

Comparing the performance results obtained in the tests, it can be concluded that the solution using Smart Events is more efficient (Table 3). Despite the slight differences caused mainly by the small scale of the simulation, the percentage gain in efficiency is noticeable.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77977-1_13

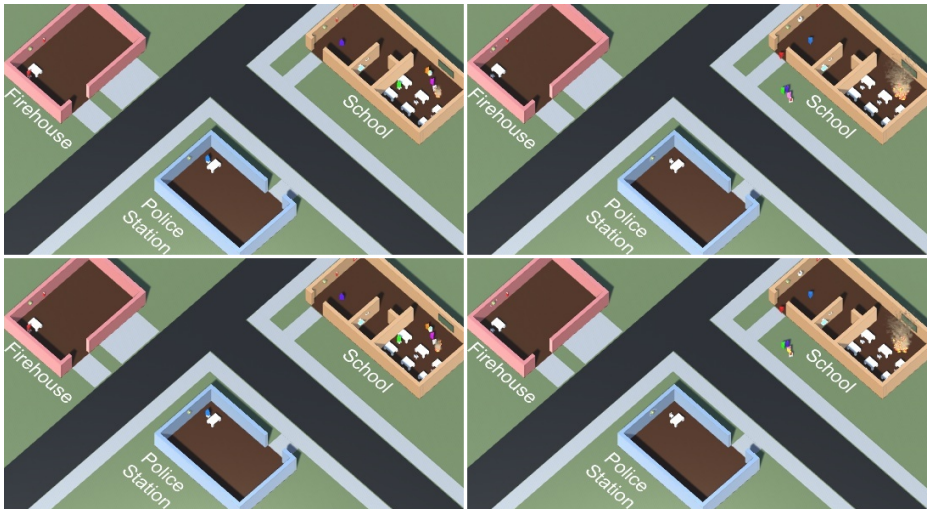

Fig. 5. Comparison of the two steps of "School" scenario with both methods: SE and FSM. The agent positions for both algorithms in the same phases of the event are so similar that the differences are difficult to spot.

Table 2. Summary of calculated distances for each agent.

Agent	Average d [m]	Maximum d [m]
Firefighter	0.032772	0.37537
Policeman	0.009708	0.162012
Teacher	0.174445	0.527891
Student	0.104061	1.155141
Son	0.18145	1.142276
Mother	0.136615	0.958957

The number of decisions made by the AI system decreased significantly, which translates to the time it takes in a single simulation frame. This, in turn, directly affects the time that can be spent on other systems during game development.

The frame rate increase achieved in each simulation run was insignificant. It can be assumed that for more complex scenarios with more agents the performance differences would be noticeable to a greater extent.

The results achieved in the performance tests conducted allowed for the conclusion that the Smart Events solution is less demanding than the one using the FSM.

4.2 Improved Smart Events

The improved SE action is shown on the basis of the "Shop" scenario. The stages of haggling between agents are presented in Figure 6. This is result for the second of the described in subsection 3.4 runs. There can be observed the key events

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77977-1_13

Table 3. Summary of performance test results for both compared methods.

Tested method	Time in one frame (μs)		Number of decisions in a frame		Total number of decisions	Average number of frames per second
	average	maximum	average	maximum		
FSM	0.517186	1602.909	0.128684521	184	764	79.5774344
SE	**0.338778**	**1255.364**	**0.121447455**	**118**	**735**	**80.57757943**
Difference	-34%	-22%	-6%	-36%	-4%	1%

from the point of view of the simulation of the group action driven by improved SE:

1. top-left: the customer triggers the haggle event, and shopkeeper speaks to him,
2. top-right: the customers begins to haggle,
3. bottom-left: the effect of random success node is positive and shopkeeper agrees to lower the price,
4. bottom-right: the price has reached a value that is satisfactory for the customer, so the whole event ends with a purchase.

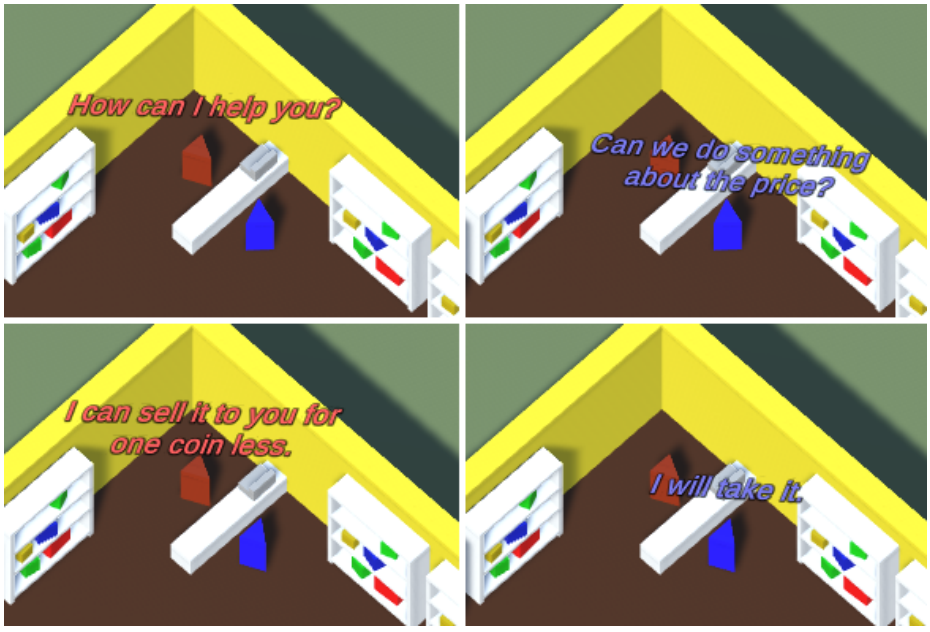

Fig. 6. Subsequent stages of agent interactions in the process of haggling.

Except that both runs end differently, the differences between them can also be seen when comparing their performance using a method analogous to the one

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77977-1_13

used in the comparison of SE with FSM. Table 4 shows the results. Lengthening the duration of the simulation through a longer haggling process meant that in the second run, the total number of decisions was doubled. The simulation in both cases showed the same scenario with two agents, therefore the maximum number of decisions in a single frame did not change. Similar to previous tests, due to the small scale, the percentage difference in average frames per second is close to zero.

Table 4. Summary of performance test results for both simulation runs using the improved method.

Test	Time in one frame (μs)		Number of decisions in a frame		Total number	Average number of frames
	average	maximum	average	maximum	of decisions	per second
First	0.353066	1466.216	0.026282051	44	123	115.9219803
Second	0.421891	3085.444	0.044221479	44	357	116.1455381
Difference	19%	110%	68%	0%	190%	0%

5 Conclusions

The aim of this work was to show a new approach to smart events and to present the overall performance of this method. The tests conducted in the first scenario allowed to confirm that the tested solution give similar results and has lower performance requirements than a competitive solution based on FSMs. The second scenario – simulation of the customers's haggling with the shopkeeper made it possible to test the new approach in practice. It also showed the possibility of applying the tested solution in events not related to emergencies. The use of a group behavior tree in the event allowed for the creation of conditions for designing coordinated actions of agents. On the other hand, the use of nodes related to randomness in character behavior trees resulted in a significant diversification of the course of the scenario. The proposed extension of Smart Events gave satisfactory results in the simulation.

The measurement method prepared as part of the work allowed to examine the efficiency of the presented SE solution. The obtained results made it possible to perform two tests: the comparison of the SE and the popular FSM method as well as the comparison of the two simulation runs of impoved SE, additionally confirming their differentiation.

SE allows developers to easily model the behavior of secondary NPC characters in the event of various types of emergency and non-emergency events that can be easily applied in a game.

Taking into account the development of the smart approach in modeling artificial intelligence in games so far, it can be concluded that this is a good direction for the development of this area of game development.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77977-1_13

Acknowledgment

This work was supported by The National Centre for Research and Development within the project "From Robots to Humans: Innovative affective AI system for FPS and TPS games with dynamically regulated psychological aspects of human behaviour" (POIR.01.02.00-00-0133/16). We thank Mateusz Makowiec and Filip Wróbel for assistance with methodology and comments that greatly improved the manuscript.

References

1. Cerny, M., Plch, T., Marko, M., Gemrot, J., Ondracek, P., Brom, C.: Using behavior objects to manage complexity in virtual worlds. IEEE Transactions on Computational Intelligence and AI in Games **9**(2), 166–180 (jun 2017). https://doi.org/10.1109/tciaig.2016.2528499
2. Cerny, M., Plch, T., Marko, M., Ondracek, P., Brom, C.: Smart areas - a modular approach to simulation of daily life in an open world video game. In: Proceedings of the 6th International Conference on Agents and Artificial Intelligence. SCITEPRESS - Science and and Technology Publications (2014). https://doi.org/10.5220/0004921107030708
3. Gu, Q., Deng, Z.: Generating freestyle group formations in agent-based crowd simulations. IEEE Computer Graphics and Applications **33**(1), 20–31 (jan 2013). https://doi.org/10.1109/mcg.2011.87
4. Iassenev, D., Champandard, A.J.: A-life, emergent AI and S.T.A.L.K.E.R. (2008), https://aigamedev.com/open/interviews/stalker-alife/
5. IGN: Bioshock Infinite – The Revolutionary AI Behind Elizabeth (2013), https://www.youtube.com/watch?v=2viudg2jsE8
6. Ingebretson, P., Rebuschatis, M.: Concurrent interactions in the sims 4. In: Game Developers Conference (2014), http://www.gdcvault.com/play/1020190/Concurrent-Interactions-in-The-Sims
7. Kallmann, M., Thalmann, D.: Modeling behaviors of interactive objects for real-time virtual environments. Journal of Visual Languages & Computing **13**(2), 177–195 (apr 2002). https://doi.org/10.1006/jvlc.2001.0229
8. Plch, T., Marko, M., Ondracek, P., Cerny, M., Gemrot, J., Brom, C.: An ai system for large open virtual world. In: Proc. 10th Annual AAAI Conf. on Artificial Intelligence and Interactive Digital Entertainment (2014)
9. Rogalski, J., Szajerman, D.: A memory model for emotional decision-making agent in a game. Journal of Applied Computer Science **26**(2), 161–186 (2018)
10. Shoulson, A., Garcia, F.M., Jones, M., Mead, R., Badler, N.I.: Parameterizing behavior trees. In: Motion in Games, pp. 144–155. Springer Berlin Heidelberg (2011). https://doi.org/10.1007/978-3-642-25090-3_13
11. Stocker, C., Sun, L., Huang, P., Qin, W., Allbeck, J.M., Badler, N.I.: Smart events and primed agents. In: Intelligent Virtual Agents, pp. 15–27. Springer Berlin Heidelberg (2010). https://doi.org/10.1007/978-3-642-15892-6_2

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77977-1_13