



Article AViLab—Gamified Virtual Educational Tool for Introduction to Agent Theory Fundamentals

Vladimir M. Petrović *¹ and Branko D. Kovačević

School of Electrical Engineering, Signals & Systems Department, University of Belgrade, 11120 Belgrade, Serbia; kovacevic_b@etf.bg.ac.rs

* Correspondence: vpetrovic@etf.bg.ac.rs

Abstract: The development and increased popularity of interactive computer games, metaverses, and virtual worlds in general, has over the years attracted the attention of various researchers. Therefore, it is not surprising that the educational potential of these virtual environments (e.g., virtual laboratories) is of particular interest to a wider scientific community, with numerous successful examples coming from different fields, starting from social sciences, to STEM disciplines. However, when it comes to agent theory, which is a highly important part of the general AI (Artificial Intelligence) research focus, there is a noticeable absence of such educational tools. To be more precise, there is a certain lack of virtual educational systems dedicated primarily to agents. That was the motivation for the development of the AViLab (Agents Virtual Laboratory) gamified system, as a demonstration tool for educational purposes in the related subject of agent theory. The developed system is thoroughly described in this paper. The current version of the AViLab consists of several agents (developed according to the agenda elaborated in the manuscript), aiming to demonstrate certain insights into fundamental agent structures. Although the task imposed to our agents essentially represents a sort of "picking" or "collecting" task, the scenario in the system is rather gamified, in order to be more immersive for potential users, spectators, or possible test subjects. This kind of task was chosen because of its wide applicability in both, gaming scenarios and real-world everyday scenarios. In order to demonstrate how AViLab can be utilized, we conducted an exemplar experiment, described in the paper. Alongside its educational purpose, the AViLab system also has the potential to be used for research purposes in the related subjects of agent theory, AI, and game AI, especially regarding future system extensions (including the introduction of new scenarios, more advanced agents, etc.).

Keywords: autonomous agents; virtual laboratories; Artificial Intelligence; game AI; Educational Technology; serious gaming; Unity; virtual worlds; computer games

1. Introduction

Over the last few decades, we have witnessed extensive development and popularity growth of interactive computer games, metaverses or MUVEs (Multi-User Virtual Environments), and similar virtual environments. Although there are distinctions between them, in a more general sense, all these environments can be called "virtual worlds" [1–3]. These virtual worlds have millions of users, interacting with them on a daily basis. Some studies have shown that students (aside from social networks activities) spend up to 20 h a week in these environments [4].

Therefore, it is rather expected that they are recognized as a fruitful ground for various research directions [2]. What is of particular interest for this paper is the educational potential of these environments and related technologies. This potential is widely recognized in the academic community, starting from social sciences, to STEM disciplines. It could be noticed that virtual laboratories represent one of the most fruitful educational applications of virtual worlds technologies. Laboratory exercises represent an extremely important part (and in most cases inevitable part) of the education process in science, technology, and



Citation: Petrović, V.M.; Kovačević, B.D. AViLab—Gamified Virtual Educational Tool for Introduction to Agent Theory Fundamentals. *Electronics* 2022, *11*, 344. https:// doi.org/10.3390/electronics11030344

Academic Editors: Osvaldo Gervasi, JungYoon Kim and Savvas A. Chatzichristofis

Received: 17 December 2021 Accepted: 20 January 2022 Published: 24 January 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https:// creativecommons.org/licenses/by/ 4.0/). engineering disciplines, enabling students to conduct experiments, gain deeper knowledge, and visualize theoretical foundations of their learning process.

Although there are many successful examples of virtual laboratories and demonstration tools in different scientific disciplines, it can be noticed that there is very little work on the subject, when it comes to the theory of agents, which will be elaborated in the next section of the paper. At this point, agents and agent theory should be briefly clarified as terms. Namely, the term agent is used in many different research disciplines. However, in this paper, we are oriented toward the field of AI, and the way agents, as well as their autonomy, intelligence, and theoretical foundations, are defined from the AI perspective. Therefore, it should be also underlined that autonomous agents and agent theory in general are long ago proclaimed as one of the central topics in the field of AI [5]. At the same time, agents represent an extremely important part of the virtual worlds and their development process, and consequently game AI. Since the early beginnings of the field, some authors even observed game agents (NPCs) as an essential part of the interactive gaming worlds [6,7]. The illustrative examples of the intersection between academic research and virtual worlds can be found in seminal research presented in [8,9], where computer games and agents inhabit them are even promoted as a platform for human-level AI research. The autonomy of agents, as well as implications of agent technology in virtual worlds in general, are thoroughly researched in [3], where a broad source of valuable information can be found. Bearing in mind all the previously said, it could be concluded that an agent's importance is twofold, regarding not just the academic community but commercial applications as well. Considering this, one could also notice that there is thorough and fruitful academic research on agent theory, which over the years resulted in the development of various agent and multi-agent simulations related to numerous applications. At the same time, there is extensive research on the development of agents for commercial-based games (with many examples well described in [10]). However, as was mentioned earlier, it could be noticed that there is a certain deficiency when it comes to developing gamified systems specially dedicated to educational purposes in the field of agents.

Therefore, this paper will present the AViLab (Agent Virtual Laboratory) virtual gamified system, the development of which is inspired by the previously elaborated standpoints. It aims to contribute to filling the observed deficiency related to agent-oriented educational systems. AViLab in a certain sense follows the "serious gaming" manner and primarily aims to serve as an illustrative tool for the demonstration of agent theory fundamentals, experimentation, and visualization of theoretical concepts. Furthermore, this system also has the potential to be used for research purposes. Due to the manner in which it was designed, the system could also be easily extended in the future, enabling a different kind of agent research.

The rest of the paper is organized as follows. In the second section, related work will be described, providing important reference points to the potential reader. The third section will present the working principles of the AViLab, including the system details, current agent structures, task description, and other important system details that will be systematically explained. The fourth section presents an example of laboratory exercise conducted in the AViLab, presenting one of the ways in which AViLab can be utilized. The fifth section gives concluding remarks and a brief discussion.

2. Background Research

Over the years, virtual laboratories have placed themselves as an important addition to the learning processes, or even a full substitution for the real laboratories. An extensive study, presented in [11], compared students accessing real, remote, and virtual laboratories in the field of robotics and concluded that students entering a virtual laboratory showed comparable results to the ones entering the real laboratory, with no significant difference in their results. A very visually appealing virtual laboratory in the field of chemistry was presented in [12], describing several practical implementations of the developed system, such as the Boyle Law demonstration. Virtual laboratory for metrology learning is presented in [13], underlining one of the obvious advantages of virtual solutions considering many fields—real laboratory equipment is often very expensive, while virtual laboratory presents a more affordable option. Virtual laboratory in the field of biotechnology is presented in [14] while pointing out another advantage of such solutions, which is a high level of availability and location independence. Availability feature is shown to be particularly important during complex circumstances, such as the recent world COVID-19 pandemic, enabling users to conduct experiments in a safe manner [15].

Many illustrative examples are coming from engineering disciplines. Research presented in [16] analyzed virtual laboratories as a successful low-cost replacement for experiments in control engineering. An interesting example is coming from the field of mechanical engineering [17], where a game-based virtual learning environment was presented, enabling students to perform different experiments related to the fundamentals of gearing. Virtual laboratory for mechatronic systems, dealing with robotics and hydraulics, was elaborated in [18], pointing out the need for broader implementation of such systems in engineering fields. One of the early examples in the robotics field can be found in [19], where the developed ROBOMOSP system was aimed to be used in a multipurpose manner (as a research tool, for the training of operators, for learning the mathematical and physical principles of industrial manipulators). A thorough review and a valuable source of information, related to the subject of virtual laboratories, can be found in one of the seminal papers [20] in the field.

However, although there is a number of illustrative examples regarding developed virtual laboratories in science, technology, and engineering disciplines, it could be noticed that there is a rather limited amount of work on the subject in the field of agent theory. To be more precise, while there is a number of examples of incorporated agent technology, it could be noticed that there are very few cases of developed virtual laboratories and similar education and research environments, dedicated exclusively and primarily to agents. One of the rare examples of virtual laboratories dedicated especially to agents is presented in [21,22]. The environment called INSIGHT aimed to explore the behavior of autonomous agents in an immersive environment, such as a golf court. Another illustrative example comes from [23], where an environment with different kinds of agents (anthropomorphic, robotic, etc.) was proposed, aiming to cover the educational aspect, among other things. Namely, it is reported that the developed environment was used as a part of the University Course, enabling students to have a practical demonstration and insight into certain aspects of agents theory, AI, artificial life, etc. However, as we already pointed out, the number of this kind of educational environments, dedicated to agents, is very limited.

All the previously analyzed can serve as a clear indicator of the necessity of developing systems such as our AViLab, bearing in mind the importance of virtual laboratories and similar environments across the many scientific fields and at the same time an obvious deficiency of such systems dedicated to agents.

3. The AViLab System

In this section, we will present a concise description of our AViLab system. The main focus will be on, what we consider to be, essential details important for the potential reader. We will start by introducing an audience with software tools used in our development process and then continue with the detailed description of the defined task, algorithms, and control logic.

3.1. Software System Introduction

The AViLab gamified educational tool has its development roots in the initial work on a single agent presented in [24]. Development is performed by using a Unity's game engine, Microsoft Visual Studio IDE, and a C#. These software platforms offer a plethora of features and possibilities, in that way enabling us to develop efficient, as well as illustrative simulations. During the development, we insisted on a few basic principles of the system that needed to be fulfilled:

- Computational efficiency: This means that the system (including the agents) must be optimally developed, and therefore omit any unnecessary computational expense, in order to rationally use available computer resources.
- Generalization: Chosen tasks must have reasonably wide applicability.
- Immersiveness and Visualization: A "serious gaming" approach, meaning that the developed system must be visually appealing to human users, while at the same time embodying theoretical concepts.
- Flexibility: Applied software tools should enable us to relatively easily modify our system (scenery, agents, task, etc.), if necessary. In that sense, an object-oriented approach is an essential requirement for system development.
- Extensibility: Different agents and different task scenarios could be added to the AViLab system in the future.

One should notice that at the moment, the AViLab software system is not publicly available or open-source, as authors do not want to jeopardize their current and future related research and publications, as well as educational applications. However, further in the text, authors will try to thoroughly describe details of the system, in order to provide all the essentials for the potential reader.

At this point, we briefly comment that our system's GUI (part of it can be seen in Figure 1) is designed in a very user-friendly manner, giving basic information to a potential operator, such as system description, details about specific agent design, task description, operating instructions, etc. In addition, a potential user has the option to return to the main menu at any moment with the back button, or by simply pressing the ESC button on the device keyboard if the simulation is in progress.



Figure 1. The operation modes page of the AViLab system's GUI offers to start one of the developed agents or enter a manual mode.

3.2. The Imposed Task

The basic task used in our system essentially represents a sort of "picking" or "collecting" task. This kind of task is chosen because apart from its numerous applications in gaming scenarios, it has wide applicability in real-world scenarios as well. Illustrative examples come from the field of robotics: household mobile robots that perform some sort of garbage collection, service robots designed for cleaning up some hazardous area, or service robots picking the balls from the golf court (e.g., [25]), etc. Furthermore, with a change of scenery, this task could be easily modified to a "Search-and-Rescue" scenario. In simulated scenarios offered by our AViLab, agents or human users are faced with the task in which they have to eliminate alien enemy satellites (increasing their score by one, each time they eliminate an alien satellite) in a game-like space world. These alien satellites are randomly placed across the simulation world. We have developed two different main modes of spawning of satellites, aiming to cover two different scenarios—the so-called "static" and "dynamic" mode. Regarding the static mode, upon each new simulation cycle, a predefined number (which can be altered according to our needs) of alien satellites is spawned simultaneously at random positions across space. Therefore, the environment is set at the beginning of the simulation, and only changes in the environment are caused by the agent (or human) actions during the simulation. On the other hand, in a dynamic mode, a certain number of satellites is spawned simultaneously at the start of the simulation, while others are spawned "one by one" in equal time intervals during the simulation. In that way, a more dynamic environment is provided. Both modes use variations of an algorithm, specially designed for our AViLab system, which searches across the game space for collision-free positions and then spawns enemy satellites at those positions.

The developed algorithm behaves in a very intuitive manner (Figure 2): it starts with generating a random position in the virtual gaming world, then checks its availability through calculation, and spawns satellites if available, or returns to the first step if not. It should be kept in mind that upon the start of each new simulation cycle, a set of new random locations is generated and selected. The introduction of randomness into the process of locations selection is providing us with a more realistic emulation of real-world scenarios. Furthermore, as a consequence of the previously elaborated, we can avoid scenarios in which an agent or human can deliberately or accidentally exploit the predictability and immutability of the enemy locations in order to increase its performance score. Some of the previously elaborated are already emphasized in [24] and are out of great significance in experiments where we want to track the performance of the agents or humans, or even compare them between themselves.



Figure 2. UML-like presentation of the basic logic behind the spawning algorithm.

Besides the described main spawn modes, there is an option to entirely customize alien satellites spawning (before the start of simulation), as this can be useful for certain demonstration purposes and scenarios.

3.3. Modes of Operation

The current version of the AViLab system offers five modes of operation. As was mentioned earlier in the text, during the R&D process, we insisted on some development principles (defined in Section 3.1). One of our starting points was also to put the main accent on relatively basic structures of agents because we wanted to demonstrate some of the fundamentals related to agent theory. Namely, due to various fruitful research directions related to both academic and gaming industry agent applications, as well as the fact that competitiveness of the field itself cause advanced AI and machine learning to often take the central research focus, the authors of this paper have an impression that more simple solutions based on the relatively fundamental principles and features of agents are sometimes superficially processed or not investigated thoroughly enough. This is a particularly important issue bearing in mind that educational tools should serve to gradually build knowledge on a matter of interest, starting from the basics. Furthermore, one should also notice the long-ago elaborated principle, that depending on a situation, the main goal is not to always build the "smartest" possible agent, but the most optimal one [26].

Upon entering the "operation modes" page of the AViLab system's GUI, one can choose to activate one of the four agents or enter the manual mode (Figure 1). Four different types of agents are developed partially inspired by theoretical concepts of agent theory elaborated in one of the seminal works in the field [27]. Therefore, we insisted on some of the defined properties such as reactive behavior, and pro-activeness. The idea was to enable users to explore and visualize how different, mostly basic types of agents, cope with the imposed tasks, to compare their performance, or to compare them with human users as well. Therefore, a manual mode is also introduced, which offers a human user to enter the simulation.

In our previously mentioned simulated game-like environment (Figure 3), an agent or a human controls a spaceship, which it uses to eliminate satellites by simply touching them. For proper tracking of their performance, a display at the top of the screen (Figure 3) shows the number of eliminated enemy satellites (scored points), elapsed time, and energy level (health) of the agents/human users.



Figure 3. Simulated game-like scenario, where we have time and energy constraints. Agents (or human) must return to its power base, at the lower-left corner of the screen, in order to restore its health (level of energy).

Agent Type 1 has a rather basic reactive design. It wanders around the simulated game world (randomly changing direction in a pre-defined time period), with no sensorial

information about the positions of the alien satellites. The agent just knows that it must eliminate the enemy if it runs into it. In other words, it must react. Regarding this, it is designed on a sort of "touch sensor" principle, which allows him to make appropriate decisions (to recognize and destroy the enemy).

Agent Type 2 has no sensorial information about the positions of enemy satellites, as well. Similar to Agent Type 1, it is also based on a sort of emulated "touch sensor", which allows him to recognize and destroy enemy satellites when it runs into them. However, it also uses sensorial information about nature and the boundaries of the space in which it acts, to systematically search the space. When we say systematically, this means that its primary goal is to cover the entire area and search every part of the space. Therefore, it "patrols" horizontally (across Y-axis), from one side of the space to another, and upon reaching the side boundary it shifts up (shift-up value is equal to its own height) and continues its movement to the side boundary again.

Agent Type 3 is a variation of the previous agent. It has the same features. However, it uses a different movement pattern and searches the space in a vertical manner, across the X-axis. Therefore, upon reaching the side boundary, it moves aside for the value that is equal to its own width and then continues its movement to the side boundary again.

All the previously described agent types are designed in a sort of minimalistic manner (e.g., they do not have the ability to learn), according to our aim to put an accent on the behavior of rather basic structures, at the same time avoiding any unnecessary computational and design expense.

Agent Type 4 is a sort of omniscient agent. Like the previous three agents, it also does not have the ability to learn, but it has perfect information about the location of enemy satellites and the surrounding world (it detects all the enemies in the space, memorizes their locations, and then apply actions according to its agenda). The strategy that this kind of agent is applying basically represents a practical visualization of the nearest neighbor search. Namely, the agent finds the position of the nearest enemy and changes its direction accordingly. Upon reaching this position and eliminating the enemy, it changes its direction toward the next nearest satellite. It repeats this pattern until there are no more enemy satellites. This strategy is not always the most optimal solution, especially in time-limited tasks, which can be efficiently demonstrated in our AViLab system. However, we will not analyze this in our paper.

As we mentioned earlier in the paper, besides agents, our system also offers a manual mode. This mode enables a human user to enter a simulations scenario and try to fulfill the imposed task. A human user controls the spaceship by using the arrows on the device keyboard. This movement (as well as the movement of the agents) is designed carefully, so the speed is always the same, even in diagonal directions (at this point, one should notice that diagonal movement was a well-known "bug" in some early versions of popular computer games). It is important to emphasize that all the significant predefined parameters of the simulation scenarios are completely the same within all modes of operation (regardless of whether the agents or human users are performing the imposed task). This is very important if we want to ensure an unbiased comparison between them, when necessary.

3.4. Simulation Details and Main Control Arhitecture of Agents

In this sub-section, we will describe some general simulation details, which are already partially elaborated in [24] but could be applied to the AViLab system in general. At the beginning of each new simulation cycle, the previously selected agent or a human user starts to move across the surrounding space, in the form of a spaceship, and according to its own agenda. There is also a simulation time limit. At this point, it should be emphasized that depending on our needs, we could adjust the duration (time limit) of each simulation cycle. Consequently, upon reaching the defined time limit, the mission (imposed task) is aborted.

Our task also has some additional requirements which must be taken care of—such as the energy level of the spaceship. We introduced such a requirement in order to have a more applicable, and realistic task scenario (examples from everyday life include driving an automobile, or piloting a plane, where you have to take care of the fuel level and adapt your actions according to this). Accordingly, if the spaceship (which is under the control of agents or humans) runs out of energy during the simulation, it will stop performing its task. Therefore, as a consequence, it will fail to successfully achieve its task. As a penalty, its score automatically drops down to zero. To prevent this sort of scenario, during the simulation, the spaceship has to return (one or more times) to the power base, so it could restore its energy. The initial value of the energy level is defined at 100 units. For every second which spaceship spends outside the power base, the energy level decreases by four units. Similarly, upon its return to the power base, the energy level of the spaceship starts restoring every second while it is in there, on increasing by four units.

Considering our current agents, the "red flag" for returning to the power base is triggered when the energy level reaches 35 units. This particular value is chosen because it enables our agents to safely return to their power bases, even when they are located at the farthest part of the simulation space. On the other hand, the "green flag" for reactivating the agent and consequently leaving the power base is triggered upon a full recharge (reaching 100 units). Of course, it should be emphasized that this kind of agent strategy, depending on the particular situation, does not always represent the most optimal solution.

All the previously mentioned simulation details and values of parameters are chosen in order to provide efficient and illustrative simulations, and at the same time enable us to have an illustrative insight into the agent's capabilities. As some of the previously mentioned parameters, they can also be altered and adjusted to different values if needed.

The general behavior cycle, applied to all four agents, can be depicted in the following manner:

- Agent starts the defined task, by going out from its power base.
- Agent moves across the simulation world (according to the designed agenda), and eliminates enemy satellites upon detection.
- Agent returns to the power base when its energy level reaches the defined critical value.
- Agent restores its energy level in the power base.
- Agent starts a new behavior cycle by going out from the power base and continuing its mission when the energy level reaches its defined value.

A simplified graphical representation of the general behavior cycle is shown in Figure 4. The kind of agent behavior cycle which we want to provide can be accomplished with different sorts of methods. Important theoretical foundations and insights related to the very principles of the shaping of agent's behavior are well described in [28,29], covering a thorough analysis of different methods and algorithms. Bearing in mind the motivation and specific details related to the development of our AViLab system, we have chosen to construct an agent's control structure based on decision threes and FSMs (Finite State Machines), as the most suitable technique. FSMs are over the years widely accepted as a dominant technique for the shaping of agent behavior in computer games [28] and are well described in the available literature.

A UML-like graphical representation of the general control structure that we applied to the agents in our game-like educational system can be seen in Figure 5. We used a sort of hierarchical FSM as the main control framework. To be more precise, the energy levels of our agents are out of the highest priority. Therefore, when the energy level reaches a critical value, it activates a transition from the current state to the state that is higher in the hierarchy. As a consequence, the agent instantly stops its current actions, end enters a higher state. Therefore, the agent heads toward its power base in order to restore its energy. Upon reaching the defined energy level (in our case—fully recharging), another transition is activated, causing the agent to go back to the lower states and continue to perform its regular task. At this point, one should notice that upon restoring their energy levels, agents Type 2 and Type 3 return to the place they were at before their energy reached the critical value (they previously memorize the location), and from that point continue

their behavior/movement pattern. In other words, they continue their mission, right where they were stooped. In that way, we are disabling the redundant behavior—in which the agent would every time search the same part of simulation space. With agent Type 1, this "return condition" is not necessary, since it is moving in random directions. For Agent Type 4, we found that the "return condition" is also rather irrelevant.



Figure 4. A simplified graphical representation of the general behavior cycle that is applied to agents.



Figure 5. UML-like presentation of the hierarchy in the agent's control system.

4. Example of the Experimental Setup

In this section, we will describe an example of an experimental setup conducted in the AViLab system. This kind of setup aims to show several possibilities of AViLab utilization at the same time. The goal of this experimental setup is to examine and demonstrate how successfully can relatively simple agent architectures (with limited knowledge about the surrounding world, sensorily inferior to humans, and without the ability to learn) cope with the imposed task, under specific time-limited conditions, and whether they can be compared in the terms of efficiency with the test groups of human users. In other words, we want to demonstrate and visualize the already mentioned premise, that depending on a situation the main goal does not always have to be developing of the "smartest" possible agent, but more simple solutions can also be optimal (with an often significant advantage, reflected in the fact that they are computationally and algorithmically less demanding).

Therefore, we will use agents 1 to 3. In order to have a clearer insight into the performance of agents, we will compare them to the two test groups of untrained human users. In addition, we will use two different environments (static and dynamic spawning mode). In this experimental setup, we limited the simulation cycle time to 60 s. The time limit does not allow for an agent/user to explore every single part of the space—in the case of Agent Type 2 and Agent Type 3, they cannot patrol the entire space world for the given time. We deliberately designed the experiment in such a way, because we wanted to see how agents and human users will handle the task under "tight" time conditions.

As is mentioned, we examined developed agents and test groups of human users in both, static and dynamic mode spawning scenarios. In both cases, the total number of enemy aliens was the same (set to 30), but they are spawned in a different manner as was explained earlier in the text. In order to provide a thorough insight into the performance of the developed agents and human test subjects, 50 simulation cycles are performed. The number of simulation cycles was chosen carefully, based on the work presented in [24]. Namely, a too-small number of simulations would probably lead to unreliable and possibly false results (e.g., agent would be falsely superior compared to the previously untrained human user). At the same time, we must take care not to have excessive repetition of the task. As was recorded in the literature, humans tend to explore possible permutations through repetition [30], and this repetition leads to learning from experience [31]. Consequently, with too many repetitions of the task, the influence of what we call "a purely untrained phase" on the overall data results would be diminished.

Simulation results for all three agents in "static mode", including a basic statistical analysis, are given in Table 1. With the observation of the obtained data, it can be seen that the performance of Agent Type 1 was rather constant, bearing in mind its built-in constraints. Therefore, it can be concluded that its overall design ensured a solid behavior. Agent Type 1, like the other two agents as well, never failed during the imposed task (which is according to the way they are designed). While its performance results during each simulation cycle vary—which is rather expected, bearing in mind the agent's limited knowledge about the outer world and its lack of sensorial information. Agent Type 2 had a smaller standard deviation, comparing to Type 1. However, it scored significantly fewer average points, max score, and achieved a lower median value. Performance of the Agent Type 3 can be observed as the best out of these three. Although it did not achieve the max score of Type 1 in any of the 50 simulation cycles, it had a better average score and median value, followed by a significantly smaller standard deviation. Although agents Type 2 and Type 3 had similar strategies, due to the shape of the spaceship, which affected the shifting value, Agent Type 2 managed to cover more space during its "patrolling". Therefore, it managed to achieve better scores.

Parameter \ User	Agent Type 1	Agent Type 2	Agent Type 3
Number of failed missions	0	0	0
Min score	9	6	14
Max score	21	12	20
Arithmetical mean value	15.62	9.2	16.92
Median value	16.00	9	17
Standard deviation	3.19	1.49	1.3

Table 1. Performance data for three agents in a static mode environment, based on the 50 simulation cycles.

In order to have a better insight into the performance quality of the agents, we compared them with two test groups of human users (with each group counting 10 individuals). Namely, previously untrained human users entered the manual mode of our simulation system, where they had the same task as the agents (one group of humans entered the static spawning mode, while the other one entered the dynamic spawning mode). At this point, the term "untrained" should be clarified more precisely. Namely, this means that selected groups of human users had no prior experience in this particular game-like simulation, and at the same time, they also had a negligible amount or no experience at all, in similar types of simulations and computer games. One should also notice that both groups of human users are chosen, while taking care of gender and age diversity, as much as it was possible.

At this point, the audience should be reminded once again that simulation conditions are exactly the same, whether the agents or the human users enter the simulation scenario (e.g., human users and agents move across the space with the same speed, etc.). In that way, a fair and objective comparison is provided. There are, of course, two obvious main differences between them. The first one is reflected in the fact that all agents are sensorily inferior compared to humans, and do not have learning capabilities. The second one is reflected in the fact that, unlike the agents, human users do not have a predefined trigger for their return to the power base and restoration of the energy level (in other words—they are not safe from failing). Consequently, human users must behave in an intuitive manner, and therefore choose which recharging strategy will apply, and adapt the strategy if needed. Table 2 shows the obtained simulation results, regarding the test group of untrained human users in a static environment.

Parameter \ User	HU1	HU2	HU3	HU4	HU5	HU6	HU7	HU8	HU9	HU10
Number of failed missions	14	4	9	4	2	7	2	0	10	1
Min score	0	0	0	0	0	0	0	6	0	0
Max score	27	26	27	25	24	27	23	22	27	22
Arithmetical mean value	13.86	15.08	15.58	14.24	15.26	15.6	15.2	14.4	15.48	13.80
Median value	15.00	15.00	17.00	14.00	15.00	17.00	15.00	14	17	14.00
Standard deviation	9.67	6.70	8.52	6.19	4.77	7.40	4.16	4.26	8.85	4.56

Table 2. Performance data of untrained human users in a static mode environment, based on the 50 simulation cycles.

While observing the experimentation process, it was noticed that human users differently coped with the imposed task, which was rather expected. This mainly resulted in their selection of different tactical approaches—some of them had more aggressive strategies resulting in a larger number of failures, while some had a more careful approach to the problem, aiming to reduce failures, even at the cost of lower scores. However, they all learned from experience to a certain degree and enhanced their strategies and overall performance during the time. With a closer look at the statistical parameters shown in Tables 1 and 2, we can conclude that all human users scored higher max scores than any of the three agents. They were also superior to Agent Type 2, regarding the mean and median value. On the other hand, agents Type 1 and Type 3 were, generally speaking, more successful than any of the human users in a static environment. These agents achieved higher mean values than any of the human test users. Furthermore, as can be seen from the statistical parameters, these agents had a more constant performance, which can be an extremely significant advantage for certain types of tasks.

We also tested agents and humans in a dynamic spawning mode. Table 3 shows the performance of another test group of humans, while Table 4 shows the performance of the agents. The second test group of human users generally achieved lower scores compared to the first one. One could also notice that agents decreased in their performance. The change of environment particularly affected the performance of Agent Type 1. However, Agent Type 3 remained overall better compared to the humans.

Table 3. Performance data of untrained human users in a dynamic mode environment, based on the 50 simulation cycles.

Parameter\User	HU1	HU2	HU3	HU4	HU5	HU6	HU7	HU8	HU9	HU10
Number of failed missions	10	4	13	4	2	5	0	1	12	1
Min score	0	0	0	0	0	0	7	0	0	0
Max score	20	24	24	25	20	23	19	21	27	18
Arithmetical mean value	10.46	13.32	13.44	13.44	12.9	13.22	12.92	13.28	13.32	12.2
Median value	11	13	16	14	13	13	13	13	16	13
Standard deviation	6.30	6.41	8.67	5.97	4.39	6.22	3.52	4.81	8.48	3.93

Table 4. Performance data for three agents in a dynamic mode environment, based on the 50 simulation cycles.

Parameter \ User	Agent Type 1	Agent Type 2	Agent Type 3
Number of failed missions	0	0	0
Min score	4	4	11
Max score	16	10	16
Arithmetical mean value	9.82	6.6	13.48
Median value	10.00	6.5	14
Standard deviation	2.86	1.23	1.28

By further increase of the number of tested human users, it could be expected that there will be those who would in certain measure outperform Agent 3 in both environments. Nevertheless, results of this experiment can be observed as a decently strong indicator regarding the appropriate level of capabilities and performance of agent structures designed according to our agenda.

Summing up the obtained experimental data can lead us to a conclusion that in certain scenarios, a carefully designed and tuned control algorithm implemented in an agent's behavior, can to a certain degree rather successfully compensate for a lack of sensorial information and complex AI. This represents a very important exercise demonstration, keeping in mind that we often meet constraints regarding the available computational resources in present highly complex virtual environments [29]. One should also notice, once again, that the ultimate goal does not always represent the development of the "smartest" possible agent, but rather the most adequate for the given situation, especially when it comes to computer games applications [26,32]. Therefore, in some scenarios, an inexpensive design can represent a better choice over a complex AI, regarding all the parameters. All of the previously demonstrated in this exercise is not only significant from the aspect of already mentioned gaming worlds, but also from the aspect of real-world applications, where inexpensive (yet reasonably efficient) design can be a crucial segment of the development process. In the end, it should be also underlined that both groups of

human participants unanimously evaluated their experience with the AViLab system as highly positive.

5. Concluding Remarks and Future Work

This paper presented the realization of the AViLab software system aiming to serve as the educational tool dedicated to experimentation and demonstration, regarding an agent's features and basic principles. Our main objective was to build a game-like system specially dedicated to agents while focusing on some of the fundamentals of agent theory. Bearing in mind that the concept of agents and related theory can often seem rather abstract to those that are getting introduced to the field (students, pupils, etc.), we strongly believe that systems such as our AViLab can help in visualization, practical demonstration, and a therefore better understanding of theoretical fundamentals.

As was discussed in the paper, virtual laboratories in their essence offer many useful features. They are more affordable than real laboratory equipment. They enable the repeatability of experiments. In addition, they offer a high level of availability and location independence, as they can be installed on almost any personal computer. Furthermore, virtual laboratories can sometimes offer experimentation possibilities unfeasible or unviable in the real world.

Our AViLab system can be utilized in several ways. Users can experiment with changing the parameters while tracking down the performance of a single agent. They can compare agents between themselves, or include test subjects and compare them with agents, under a certain experimental setup (as was demonstrated in our exemplary experiment). You can also customize spawning, in order to demonstrate certain scenarios of interest. Therefore, different scenarios can be designed, depending on desired learning/experimentation/demonstration agenda. In other words, our system is suitable to be used as a demonstration tool during course lectures, as well as for laboratory exercises (designed according to the aim of the lab supervisor), aiming to provide efficient demonstration of the important insights of the agent's technology fundamentals in illustrative, as well as an immersive manner (e.g., exploiting agent's predictability, autonomy, control architectures, etc.).

Considering future work, besides the standard parameters that we can change, such as the number of satellites, time-limit of the simulation, etc., we also plan to work on a few different variations of our simulation scenario features, as well. Consequently, this could enable us with a wider framework for demonstration. Furthermore, since our system is developed in such a way (due to an object-oriented programming approach), that in future work it can be rather easily expanded with additional modes, upgrades, and scenarios; we consider the development of a "Battle Arena" mode, where human users can compete directly against a chosen agent. A deeper pedagogical analysis of the system, oriented toward the user experience, is yet to be thoroughly researched, with careful and broader elaboration of different experimental and demonstrations scenarios. Of course, development directions oriented toward pedagogical agents are something to be thought about in future work. However, modifications in this direction will be thoroughly analyzed, while taking care not to violate the basic principles of system development elaborated in the paper.

In the end, one should notice once again that the developed AViLab system aims not only to serve as an educational tool but at the same time has the potential for various research applications.

Author Contributions: Conceptualization, V.M.P.; methodology, V.M.P.; software, V.M.P.; validation, V.M.P.; formal analysis, V.M.P.; investigation, V.M.P.; resources, V.M.P.; data curation, V.M.P.; writing—original draft preparation, V.M.P.; writing—review and editing, V.M.P.; visualization, V.M.P.; supervision, B.D.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Conflicts of Interest: The authors declare no conflict of interest.

References

- 1. Kumar, S.; Chhugani, J.; Kim, C.; Kim, D.; Nguyen, A.; Dubey, P.; Bienia, C.; Kim, Y. Second Life and the New Generation of Virtual Worlds. *Computer* **2008**, *41*, 46–53. [CrossRef]
- 2. Bainbridge, W.S. The Scientific Research Potential of Virtual Worlds. Science 2007, 317, 472–476. [CrossRef] [PubMed]
- Petrović, V.M. Artificial Intelligence and Virtual Worlds—Toward Human-Level AI Agents. *IEEE Access* 2018, 6, 39976–39988. [CrossRef]
- 4. Thompson, C.W. Next-Generation Virtual Worlds: Architecture, Status, and Directions. *IEEE Int. Comput.* 2011, 15, 60–65. [CrossRef]
- Jennings, N.R.; Sycara, K.; Woodridge, M. A Roadmap of Agent Research and Development. *Auton. Agents Multi-Agent Syst.* 1998, 1, 7–38. [CrossRef]
- Doyle, P. Virtual Intelligence from Artificial Reality: Building Stupid Agents in Smart Environments. In Proceedings of the AAAI '99 Spring Symposium on Artificial Intelligence and Computer Games, Palo Alto, CA, USA, 22–24 March 1999; pp. 37–41.
- van Lent, M.; Laird, J.E.; Buckman, J.; Hartford, J.; Houchard, S.; Steinkraus, K.; Tedrake, R. Intelligent Agents in Computer Games. In Proceedings of the Sixteenth National Conference on Artificial Intelligence (AAAI-99), Orlando, FL, USA, 18–22 July 1999; pp. 929–930.
- 8. Laird, J.E.; van Lent, M. Human-Level AI's Killer Applications: Interactive Computer Games. AI Mag. 2001, 22, 15–25.
- 9. Laird, J.E. Research in Human-Level AI Using Computer Games. Commun. ACM 2002, 45, 32–35. [CrossRef]
- 10. Risi, S.; Preuss, M. From Chess and Atari to StarCraft and Beyond: How Game AI is Driving the World of AI. *KI KünstlicheIntell*. **2020**, *34*, 7–17. [CrossRef]
- 11. Tzafestas, C.T.; Palaiologou, N.; Alifragis, M. Virtual and Remote Robotic Laboratory: Comparative Experimental Evaluation. *IEEE Trans. Educ.* **2006**, *49*, 360–369. [CrossRef]
- 12. Gervasi, O.; Riganelli, A.; Pacifici, L.; Laganà, A. VMSLab-G: A virtual laboratory prototype for molecular science on the Grid. *Future Gener. Comput. Syst.* **2004**, *20*, 717–726. [CrossRef]
- 13. Ballu, A.; Yan, X.; Blanchard, A.; Clet, T.; Mouton, S.; Niandou, H. Virtual metrology laboratory for e-learning. *Procedia CIRP* **2016**, 43, 148–153. [CrossRef]
- 14. Abramov, V.; Kugurakova, V.; Rizvanov, A.; Abramskiy, M.; Manakhov, N.; Evstafiev, M.; Ivanov, D. Virtual Biotechnological Lab Development. *BioNanoScience* 2017, 7, 363–365. [CrossRef]
- 15. Kapilan, N.; Vidhya, P.; Gao, X.-Z. Virtual Laboratory: A Boon to the Mechanical Engineering Education during COVID-19 Pandemic. *High. Educ. Future* **2021**, *8*, 31–46. [CrossRef]
- 16. Goodwin, G.C.; Medioli, A.M.; Sher, W.; Vlacic, L.B.; Welsh, J.S. Emulation-Based Virtual Laboratories: A Low-Cost Alternative to Physical Experiments in Control Engineering Education. *IEEE Trans. Educ.* **2011**, *54*, 48–55. [CrossRef]
- 17. Aziz, E.-S.; Chang, Y.; Esche, S.K.; Chassapis, C. A Multi-User Virtual Laboratory Environment for Gear Train Design. *Comput. Appl. Eng. Educ.* **2014**, *22*, 788–802. [CrossRef]
- Petrović, V.M.; Nikolić, B.; Jovanović, K.; Potkonjak, V. Development of Virtual Laboratory for Mechatronic Systems. In Advances in Robot Design and Intelligent Control, Proceedings of the 25th Conference on Robotics in Alpe-Adria-Danube Region—RAAD 2016, Belgrade, Serbia, 30 June–2 July 2016; Springer International Publishing AG: Berlin/Heidelberg, Germany, 2016; pp. 622–630.
- Jaramillo-Botero, A.; Matta-Gómez, A.; Correa-Caicedo, J.F.; Perea-Castro, W. ROBOMOSP: ROBOtics Modeling and Simulation Platform. *IEEE Robot. Autom. Mag.* 2006, 13, 62–73. [CrossRef]
- 20. Potkonjak, V.; Gardner, M.; Callaghan, V.; Mattila, P.; Guetl, C.; Petrović, V.M.; Jovanović, K. Virtual laboratories for education in science, technology, and engineering: A review. *Comput. Educ.* **2016**, *95*, 309–327. [CrossRef]
- Strippgen, S.; Christaller, T. INSIGHT: A Virtual Laboratory for Looking into an Autonomous Agent. In Proceedings of the PerAc '94. From Perception to Action, Lausanne, Switzerland, 7 September 1994; pp. 388–391.
- Strippgen, S. INSIGHT: A virtual laboratory for looking into behavior-based autonomous agents. In Proceedings of the First International Conference on Autonomous Agents (AGENTS '97), Marina del Rey, CA, USA, 5–8 February 1997; ACM Press: New York, NY, USA, 1997; pp. 474–475.
- 23. Jung, B.; Milde, J.-T. An Open Virtual Environment for Autonomous Agents Using VRML and Java. In Proceedings of the Fourth Symposium on Virtual Reality Modeling Language (VRML '99), Paderborn, Germany, 23–26 February 1999; pp. 7–11.
- 24. Petrović, V.M. An Inexpensive Design of Agent's Behavior During a "Picking Task" in a Simulated 2-D Virtual Game-Like Environment. In Proceedings of the 2021 IEEE Zooming Innovation in Consumer Technologies Conference (ZINC), Novi Sad, Serbia, 26–27 May 2021; pp. 16–20.
- 25. Pereira, N.; Ribeiro, F.; Lopes, G.; Whitney, D.; Lino, J. Autonomous Golf Ball Picking Robot Design and Development. *Ind. Robot.* **2012**, *39*, 541–550. [CrossRef]
- 26. Lidén, L. Artificial Stupidity: The Art of Intentional Mistakes. In *AI Programming Wisdom 2*; Rabin, S., Ed.; Charles River Media Inc.: Hingham, MA, USA, 2004; pp. 41–48.
- Franklin, S.; Graesser, A. Is it an Agent, or just a Program? A Taxonomy for Autonomous Agents. In Proceedings of the Third International Workshop on Agent Theories, Architectures, and Languages, Budapest, Hungary, 12–13 August 1996; Springer: Berlin/Heidelberg, Germany, 1997; pp. 21–35.

- 28. Cavazza, M. AI in Computer Games: Survey and Perspectives. Virtual Real. 2000, 5, 223–235. [CrossRef]
- 29. Khoo, A.; Zubek, R. Applying Inexpensive AI Techniques to Computer Games. IEEE Intell. Syst. 2002, 17, 48–53. [CrossRef]
- 30. Coyne, R. Mindless repetition: Learning from computer games. Des. Stud. 2003, 24, 199–212. [CrossRef]
- Brinkmann, M. Repetition and Transformation in Learning. In *Transformative Learning Meets Bildung: An International Exchange;* Sense Publishers: Rotterdam, The Netherlands, 2017; pp. 73–83.
- 32. Cass, S. Mind Games. IEEE Spectr. 2002, 39, 40–44. [CrossRef]