Pedagogical agent in Multimedia Interactive Modules for Learning – MIMLE

Kristijan Kuk a,*, Ivan Milentijević b, Dejan Rančić b, Petar Spalević c

a School of Electrical Engineering and Computer Science Applied Studies, Belgrade, Serbia
b Faculty of Electronic Engineering, Nis, Serbia
c Faculty of Technical Sciences, K. Mitrovica, Serbia

1. Introduction

One of the basic goals of all educational systems is enhancing the quality of lectures and that may be done by applying the new teaching technologies. The new technologies should improve students’ learning process as well as teaching process in college education. The latter one should be completely adapted to students’ needs and attitudes. Many of the educational systems in use today are based on (VanLehn & Chi, 2007) used in intelligent tutoring systems (ITS). Depending on the tracing technique used for the student modeling is given in (Amižić, Stankov, & Rosić, 2002). The advantages of ITS, that is called model tracing. The learning system – MIMLE uses one of the diagnostic techniques of ITS, that is called model tracing. The advantages of the tracing technique used for the student modeling is given in Tex-Sys system (Amižić, Stankov, & Rosić, 2002). Depending on the given set of the student’s interaction with the system, the system intervenes by activating the agent which is actually a help option within the system. By applying Marcov decision process – MDP (Puterman, 1994), the pedagogical agent of the system decides if the Help window should appear or not. The goal of this paper is to determine the minimal intervention of the agent in the MIMLE system that will not irritate the student by constantly appearing, but will discreetly show up in certain situations only, to offer help option when necessary, and after that to disappear for the rest of the problem solving.

Making a progress towards higher levels of game is measure that indicate student advancing in learning process. The system evaluates the student through his or her interaction with the
system counting correct or incorrect answers and measuring time taken to act. Having in mind that the MIMLE system in this paper has two levels of problem solving corresponding to two level-game, we have also shown the model for evaluating the knowledge necessary for upgrading to the next level. The authors (Sestokas, Burns, & Forbes, 2009) address the problem of analyzing threshold for passing to the next level in several types of adaptive learning environments. If the student acquired more than 75% of correct answers, the system will let him or her pass to the next level; on the other hand, if the student’s results are below 50% of correct answers, he or she will be demoted to the previous level. Our model for assessing the amount of knowledge needed for the next level is based on determining the importance of coefficient in formula for calculating how successful student’s problem solving has been at a certain game level. The coefficient weights stands with input variable in a simple formula for the sum of the acquired values, such as: the number of correct answers, time taken and the number of hints used by the student. The value of the coefficient has been set based on the results acquired by students on previous tests and teacher experience.

This paper, after the introduction in Section 2, gives a survey of the existing classes of interactive multimedia learning modules used for courses in applied mathematics and physics and also proposes their combination as a new class MIMLE used in the course Computer Graphics. Section 3 set a problem, proposed its solution based on a pedagogical agent and describes its design. The states and characteristics of the student modeling in student’s learning process are given in Section 4, while the process of making decisions based on these states by the pedagogical agent is presented in Section 5. Further on, in Section 6, the implementation of the pedagogical agent in MIMLE system by using two models for making decisions on short-term and long-term knowledge of students. Section 7 gives the effects of using the MIMLE system to students preparing their exam. Finally Section 8 concludes the paper and discuss some important issues.

2. Multimedia Interactive Modules for Learning – MIMLE

The analysis of the existing Multimedia Interactive Modules for Learning in the field of education shows that there are 3 classes of applications used effectively for teaching students mathematics and applied physics. Those classes of the existing educational applications are:

- Class 1 – IMMEX / Gismo simulations
- Class 2 – simulation games
- Class 3 – educational games

Online simulations fall into the group of the first class of interactive multimedia applications. Relying on the cognitive load theory, Sweller (1999) and Mayer (2001) suggest that complex problems, procedures and complex problem solving can be better understood if they are being taught as connected units. In order to show those units in the best possible way (Fleisher, 2008) they recommend using practical examples through numerous online simulations – Gizmos, that are a product of the Explore Learning company. On www.explorelearning.com, a web page for explorative learning via Internet, there are numerous simulations in the area of mathematics and applied physics, represented in the form of interactive problems. The environment in which teachers can stimulate ‘spontaneous’ learning (Plager, 1973) has been represented through the usage of IMMEX – Interactive Multi-Media Exercises platform. This platform contains a large collection of online multimedia simulations for solving mathematical and other scientific problems (Underdahl, Palacio-Cayetano, & Stevens, 2001) where in order to acquire conceptual understanding, users build their knowledge through self-discovery.

The second and third class of these applications consist of the games used as a motivational factor in the learning process. With the use of simulation games in economy and military forces, students comprehend the complex system much faster and better than they could in classical way, by merely describing it. By using a computer, student has a possibility to study, manipulate and practice the examples in an active way by changing the parameters and analyzing the results, thus acquiring new knowledge and experiences. Romiszowski, Jost, & Chang (1990) believe that in the teaching process we can convey very serious lessons in the form of 3D interactive simulations. On the other hand, educational games have a goal to teach student something, e.g. some skill or general knowledge. The very foundation of these games is satisfying students’ need for esteem and acknowledgement through competition with others. However, the goal of educational games is not to enable student to ‘play’, but to teach them content and the game is just a convenient method and a way to conduct studying (Prensky, 2001).

After analyzing the existing classes of educational interactive applications, we came to a conclusion that none of them fits the specifics of the material of the Computer Graphics course, taught at the first year of vocational studies in Belgrade at the School of Electrical Engineering and Computer Science Applied Studies. Considering that the students to whom the new kind of studying system is to be offered, did not show significant motivation for studying when using multimedia applications that fall into the above mentioned categories, we have introduced some pedagogical elements into our learning system to make it more appealing to them. Since our system is being shown to the first year students in the form of learning modules, it was necessary to select such a surrounding that would conform to the affinities, previous knowledge and the age of the end users. So, the realized system has the interface that students come across in various games. Furthermore, in the system, or module in the further text, we did not just insert the games interface, we have implemented the basic games elements as well, such as: result, time and difficulty level.

We have named these new applications the Multimedia Interactive Module for Learning – MIMLE. They represent new applications that comprise game elements and represent 2D interactive simulations for learning designed for the new Net-generation of students. Those systems actually represent a new class of applications which are an intersection of a combination of the three classes of the existing educational applications: IMMEX/Gismo simulations, simulation games and educational games (Fig. 1).

An innovative way of testing knowledge on computer tests has served as a basic idea for the way to present the teaching material to a student. By using the techniques of the first class of the

![Fig. 1. MIMLE – a new class as an intersection of the three existing classes of educational applications.](image-url)
innovative tests – select/recognize (Parshall, Davey, & Pashley, 2000) we got an idea that the answer in interactive multimedia learning module can be offered as a sequence of images that student can click on. Since the buffer that we work with in Z-buffer algorithm uses 16 bits, the task in ‘Z-buffer’ module (which falls into MIMLE class) is to determine the value of every bit, i.e. the content of the registry in different situations shown in the problem. The correct answer that will fill out the content of one bit is one of the offered answers that are shown to student in the form of squares that need to be selected.

As shown in Fig. 2, ‘Z-buffer’ module has two main parts, the user interface and the pedagogical agent. Visual display of the system interface suits the kind of students that it was made for, i.e. students aged from 18 to 20. We used the motives of contemporary applications as well as motives of the graphic interface Aero, found in operating systems Windows Vista and Windows 7. Graphics are adjusted to the age of the students so they would find that interface agreeable. The interface incorporates visually attractive effects of glow, transparency and reflection, which makes the so called ‘fancy’ technology (Fig. 3). The whole design and architecture of the ‘Z-buffer’ module has been done with ActionScript 3.0, an object oriented programming language, supported by Adobe Flash CS3 package.

3. Pedagogical agents

The main problem that creators of distance learning systems encounter is the inability of the system to be personalized for each student and to enable individual attention. Giving instructions in face to face communication gives an individual an opportunity to study effectively and fully understand the lessons. Speech trainer and virtual trainer, as types of pedagogical agents, are used as means of personalizing the installment in these systems. The possibility of using the pedagogical agent in a learning system is a very important option that replaces the instructor that would be present in real conditions. Pedagogical agents are computer generated virtual mentors in form of humans, animals or shapes that usually possess voice, emotional characteristics and the ability to learn.

Microsoft created a programmable software agent that can be coded within Windows applications and that provides users with instructions on how to use the software. Microsoft Word popularized this concept through animated characters, such as: The Office assistant paper clip and Merlin the wizard.

Based on the definition given by AIMA – Artificial Intelligence: A Modern Approach (AIMA, Russell & Norvig, 1994), agents are divided into two large groups: reflex agents and goal-oriented agents. Reflex agent is a string of simple conditional actions formed as follows: if condition then action. In the core of goal-oriented agents lies a priori knowledge of the expert in the given area.

Some critics suggest that the agents turn student’s attention towards the process of learning and by doing that they interfere with the learning itself. The interaction of the agent with the student must be such that it does not hinder student’s actions but gives him or her instructions and makes the process of the successful problem solving easier (Fischer, 1994). Also, the goal of this paper is to determine the minimal number of interactions of an agent with a student so they would not be a distraction and would not interfere with problem solving.

The research (Kim, Keller, & Baylor, 2007) conducted on students of one engineering college in America, about positive effects...
of using textual messages in relation to the same messages generated by animated agents, has shown that the plain textual messages had the same effect on students’ learning as the textual messages generated by animated agents. Having that in mind, in this paper we exclusively use the textual messages shown in Help windows as a means of agent-student interaction.

In our MIMLE educational application, through the Help window, realized as a text message, and generated by a pedagogical agent, student is given a ‘piece of knowledge’, in the form of a theorem or a definition, necessary for successful solving the problem given in the module. MIMLE module is not a typical educational game, it uses only minimal elements of fun to increase student’s knowledge based on all these details. Based on the results, the module is supposed to determine the agent’s intervention. Aside from assessing the current level of knowledge so he or she could get help through the Help window, that is opened by a student before the second answer. The abbreviations used in Table 1 are: Correct – C, Incorrect – Ic, Yes – Y and No – N.

We have used Markov decision process theory (MDP) to prove the correctness of the hypothetical states (given in Table 1), in which a student can find their own state after providing certain combinations of events, based on which pedagogical agent reacts.

5. Markov decision process

Markov’s models are mathematical models that describe stochastic processes, i.e. the processes that generate an accidental outcome sequence according to certain probabilities (Schapire & Kostas, 2003). Interaction between the environment and the agent in it can be shown as follows:

\[
S_0 \xrightarrow{a_0} S_1 \xrightarrow{a_1} S_2 \xrightarrow{a_2} \ldots
\]

where at the beginning, the agent is at a state $S_0$ and is performing an action $a_0$. The system gives the reward $r_1$ to the agent and moves to the state $S_1$. After that, at the state $S_1$ the agent performs the action $a_1$, receives the reward $r_2$ and the system goes to state $S_2$ and so forth.

In stochastic world action $a_t$ that is at a state $S_t$ is not the result of a new exact state, but a distribution of probability $P(S_{t+1} | S_t, a_t)$ through all possible states $S_t + 1$. The probability of state $S_t$ changing to the state $S'$ by action $a_t$ is shown in the following equation:

\[
P_{S_{t+1}} = P(S_{t+1} | S_t = S, a_t = a)
\]

The given equation is called the crossing function. As a consequence of crossing of the state $S$ into the state $S'$ when action $a$ has been performed, we can expect the reward $r$, given as:

\[
R_{S_{t+1}} = E[r_{t+1} | S_t = S, a_t = a, S_{t+1} = S']
\]

The choice of a proper action for the current state is defined by policy $S \rightarrow A$ (where $S$ is a string of possible states and $A$ is a string of possible actions). The main goal of Markov decision process is to find the optimal policy that will maximise the expected gain for every state at MDP. If a space of state and action is finite, then MDP is called a finite MDP (Derman, 1972). The most important aspect of dynamics of a finite MDP is determining $P_{S_{t+1}}$ and $R_{S_{t+1}}$.

6. Implementation of a pedagogical agent into the MIMLE

The pedagogical agent in ’Z-buffer’ module has a double role. One of the tasks this module performs is determining student's current level of knowledge so he or she could get help through the agent’s intervention. Aside from assessing the current level of knowledge, the pedagogical module is supposed to determine the long term knowledge that student gained by playing a certain level. Based on the results, the module is supposed to determine if the student has enough knowledge to pass to the next level of ‘Z-buffer’ module or not.

6.1. Short term knowledge assessment – Agent’s intervention model

Based on MDP, the pedagogical agent exclusively decides on its own actions based on the action of the state in which a student is. Depending on its interaction with ‘Z-buffer’ module, a student can find himself in one of the four states: knows, doesn’t know, accidental hit, accidental miss, there is $S = \{\text{knows}, \text{doesn’t know}, \text{accidental hit}, \text{accidental miss}\}$. Possible decisions, the agent’s actions – we represent as ‘show’ or ‘unshow’.

Graphical transition is a convenient way of showing the dynamics of the finite MDP. Fig. 4 shows a transitional graph to illustrate agent’s intervention in ‘Z-buffer’ module. On the graph there are two types of nodes: state node and action node. The state nodes show ‘action-state’ couple (the little filled circle denominated by

<table>
<thead>
<tr>
<th>$Answer_{t-1} (A_{t-1})$</th>
<th>$Help/H_t$</th>
<th>$Answer/A_t$</th>
<th>$State/S_t$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ic</td>
<td>N</td>
<td>Ic</td>
<td>Doesn’t know</td>
</tr>
<tr>
<td>Ic</td>
<td>Y</td>
<td>C</td>
<td>Accidental hit</td>
</tr>
<tr>
<td>Ic</td>
<td>Y</td>
<td>C</td>
<td>Knows</td>
</tr>
<tr>
<td>C</td>
<td>N</td>
<td>C</td>
<td>Doesn’t know</td>
</tr>
<tr>
<td>C</td>
<td>N</td>
<td>C</td>
<td>Knows</td>
</tr>
<tr>
<td>C</td>
<td>Y</td>
<td>C</td>
<td>Accidental miss</td>
</tr>
<tr>
<td>C</td>
<td>Y</td>
<td>C</td>
<td>Knows</td>
</tr>
</tbody>
</table>
the state's name and which connects the nodes of the action of an agent by a line), while action nodes represent a possible action of the agent for every state of the student (the big open circle that is denominated by the name of the action). The start is at the state S and by going down the line in the direction of the arrow we come to the agent's action (S, a). After that, a student's state reacts to the agent's action, crossing to the next state, S'. For each state a on the graph there is a triple (S, S', a). The probability of crossing to the state S' is denominated as $P_{SS'}$, while the expected reward for that state is denominated as $R_{SS'}$.

The case where an agent performs the action 'show' at a state when a student has just started the problem solving in 'Z-buffer' module is pointless, because the student could not have been in the state 'doesn't know' at the beginning of the problem solving. We can offer the following actions to an agent, depending on student's state:

- $A(\text{knows}) = \{\text{unshow}\}$
- $A(\text{doesn't know}) = \{\text{show}\}$
- $A(\text{accidental miss}) = \{\text{probability for unshow} P^{u}_{SS} (1- \alpha), \text{probability for show} P^{s}_{SS} (\alpha)\}$
- $A(\text{accidental hit}) = \{\text{probability for show} P^{s}_{SS} (\beta), \text{probability for unshow} P^{u}_{SS} (1- \beta)\}$

If student's current state during learning is marked as 'knows', then an agent's activation is completely unnecessary for further learning. Agent's intervention in that state would only hinder student in solving the given problem and might even cause him/her to answer incorrectly. On the other hand, when a student is in 'doesn't know' state, the appearance of a Help window is of immeasurable importance, because by showing the basic definitions for the given area, student is being directed the right way towards the correct answer. The case when student accidentally gives the right answer during the problem solving is a state when an agent is supposed to start the 'show' action, because the observed correctness during the problem solving should be supported. The probability of the correct answer $- \alpha$, is great, even though student is at a state 'doesn't know'.

In addition to that, if the agent does not show the Help window, even if student accidentally makes a mistake, for which there is a great probability, shown as $\beta$, that means that student is giving correct answers and that an agent's intervention is not necessary in that case. However, if student is at the state 'knows' at one moment during the learning, and at a state 'doesn't know' at the other, then the probability of the correct answer is small and is shown as $1 - \beta$. By observing the answer after the agent's intervention, it can be shown as a unit of reward. If the rewards $R_{\text{accidental miss}}$ and $R_{\text{accidental hit}}$ go down to 0, which makes $P^u_{SS} \rightarrow 1$ with a large number of correct answers, that shows that the given policy is correct. This system is then shown as a finite Markov decision process (MDP), and we can enlist the probability of crossing and expected rewards, as shown in Table 2.

Precision testing of a model for a pedagogical agent in an interactive multimedia system has been conducted during the summer semester of 2009/10, for the teaching unit 'Z-buffer', the algorithm on Computer Graphics course. The research was conducted on students of the first year of vocational studies in Belgrade at the College of Electrical Engineering and Computer Science. After the lecture on distance learning system – Moodle, students got MIMLE 'Z-buffer' as additional material to prepare that part of their lessons. After that an experiment was conducted which implied observing students’ activities while playing the MIMLE and analyzing the results they accomplished. Students who took part in the experiment were 183. The results of playing MIMLE were recorded in an electronic base from which the accomplished results were afterwards analyzed.

The precision of the short term model of the pedagogical agent was determined by applying the right policy of $\pi^*$ in MDP. The validity of the applied $\pi^*$ policy, which was the basis for constructing an agent's intervention in MIMLE graph, as shown in section 5, was executed by the parameters of reward:

$$R^a_{SS} = \frac{\# \text{answers correct } S}{\# \text{count } S}$$

From the large amount of data that represents memorized actions and interventions of an agent during the game play of 'Z-buffer' module, we abstracted those combinations that are defined as student's state in student modeling. In Table 3 there is an example of abstracted data in one sequence of the answers and the state of an agent in MIMLE system, that we have analyzed. Denominations used in the sequence of agent's answers and states are as follows:

- Correct $- C$, Incorrect $- Ic$, Show $- S$ i Unshow $- U$.

Fig. 5 presents analyzing the results in Table 3. performed on the basis of sequence information compiled from answers before an agent's intervention, agent's states and answers after an agent's intervention, in the order of steps.

As the number of sequences increased we got the result in which student was giving more and more correct answers after an agent's intervention, by which we decreased the function of the reward almost to zero. The large number of interactions lead to the following conclusion: $\# \text{count } \rightarrow \infty \Rightarrow R^a_{SS} \rightarrow 0 \Rightarrow P^{u}_{SS} \rightarrow 1$. Based on the given conclusion we have shown that the given policy of an agent’s intervention – $\pi^*$ has given the expected results, and by that we have proven the propriety of its usage.

6.2. Long term knowledge assessment – The model of passing onto the next level

To assess student's knowledge level needed for the next level of playing in MIMLE, we have developed a model for the assessment of the level of knowledge that uses coefficients as variable values
which are theoretically determined based on learning by experience. The rule for percentage determining of knowledge applies basic arithmetic operations on the incoming parameters of the model. The importance of the influence of the incoming values on the final calculation of knowledge was determined based on the empirical values of the rates, given by teacher. Current level of knowledge for a level of an educational game that student possesses is given in the following formula:

\[
P(X = \text{Mastered}) = a^i \left( \sum_{i=1}^{n} A_i - h^i \sum_{i=1}^{n} H_i \right) + t \left( \sum_{i=1}^{n} A_i - h^i \sum_{i=1}^{n} H_i \right) t_a + \sum_{i=1}^{n} H_i t_b \]

(5)

where \(a\), \(h\) and \(t\) represent coefficients with three basic parts of the formula. Values that coefficients can have may vary from 0 to 1.

The first part of the formula has the most important share in calculating student’s final level of knowledge on a certain level of the game, because it uses the number of correct and incorrect answers on that level – \(A_i\).

The second importance is the second part of the formula that denotes how many times Help window has been used during the game – \(H_i\). This part of the formula has a negative sign because it lowers the probability of the final level of student's knowledge on a current level. The part with time has the least importance for calculating knowledge. When student uses Help window, the time needed to give the answer increases, which causes the sum of the student’s knowledge to decrease.

Values and \(t_a\) in \(t_b\) time part of the formula show average time that student needs to give the answer when using Help window and when the student is not using the Help window.

- Time \(t_a\) – average time needed to give an answer when not using the Help window,
- Time \(t_b\) – average time needed to give an answer when using the Help window,

where \(t_a + t_b = t_{\text{total}}\) – total playing time.

In this model best measured time is indicated as \(t_{\text{limit}}\) and the rule of time is interpreted as:

\[
\text{IF } (t_{\text{total}} < t_{\text{limit}}) \text{ THEN PROBABILITY OF KNOWLEDGE IS GREATER}
\]

Based on the experience that teachers have, the value for the importance of the rates in given formula is shown in Table 4.

As it can be seen in Table 4, the most important coefficient for calculating the sum of student’s knowledge for the given level is the rate \(a\). The influence of the first part of the formula on calculation of the sum of student's knowledge is great in the case when the number of correct answers in the game is large as well. Because of that, the value assigned to this rate is 0.50. The lesser value, but also important for calculating knowledge, is assigned to the rate from the part of the formula which observes how many times help has been used – \(h\), and it is represented with the value of 0.30. The third part of the formula has the least share in the calculation so the coefficient assigned to it has the value of just 0.15.

Calculations made to determine if student has grasped the current level of the game and if they should pass onto the next level are based on the values acquired during the playing of MIMLE and with the help of formula (5). Pedagogical agent decides on that: if \(P(X) > 75\%\) then the student has grasped the current level.
and should be let to pass onto the next one, if not, student stays on the same level.

The precision of a long term model in a pedagogical model was determined by the standard technique of comparing practical results that students acquired on a test and the appraisal of their knowledge by a pedagogical agent. We got encouraging results on the precision of students' knowledge appraisal generated by MIMLE based on empirical values for rates. The mistake made by the MIMLE compared to the practical results acquired on a paper test, was very small: $\varepsilon = 0.16$. By comparing the results with the same set of data in the archive data, we also came to a conclusion that the values in formula 4 are not very sensitive to certain changes. The accuracy of the appraisal stays almost the same if, for example, we change the value of the rate $'a'$ from 0.50 to 0.75 and the value of the rate $'b'$ from 0.30 to 0.55. Slightly larger sensitivity of appraisal has been noticed while changing the value of the rate $'c'$. In order for a longterm model to stay precise in student's knowledge appraisal, the value of the rate $'c'$ can vary from 0.15 up to 0.30. How much knowledge student should have for a system to transfer him/her to the next level of the game, depends exclusively on teacher's assessment of how the level of understanding of that teaching unit will affect the learning of the other teaching units of that course. As a border value for the given game level, we have taken the blue dot that signifies that a student has grasped 75% of the teaching material, thus becoming successful. By calibrating the value of importance of coefficients in formula 5, the blue dot has been kept, which shows that the given long term model is very precise in appraisal of student's knowledge and immune to the changes of the value of rates that can be very different, depending on teacher's subjective assessment.

7. Experimental results

Testing of learning efficiency with the use of MIMLE was performed during the summer semester 2009/10 for the teaching unit ‘Hidden surface techniques’. During those lectures on the course on Computer Graphics students were informed about the principle of the ‘Z-buffer’ algorithm operation. The professor presented the teaching unit with the use of a classic method, with blackboard and without modern teaching resources such as a projector, Power Point presentation, etc. After the lectures on the distant learning system – Moodle, students also obtained MIMLE ‘Z-buffer’ module as additional material for exam preparation. After that, a pedagogical experiment was carried out, and it consisted of monitoring students’ activities in the period prior to taking the exam, and analyzing the results they accomplished on the final exam.

The simple experiment to effects of using the MIMLE system to students preparing their exam included 183 students, and 4 predefined groups were monitored: $\Pi$ (the group which did not attend the lectures and did not use MIMLE), $\Pi_I$ (the group which did not attend the lectures, but used MIMLE), $\Pi_T$ (the group which attended the lectures but did not use MIMLE), $\Pi_I$ (the group which attended the lectures and used MIMLE). Characteristics of the results distribution on the final test are shown in Table 5.

The total accomplishment of students who used MIMLE regardless of whether they attended the lectures or not (the second and fourth row in Table 1, $\Pi_T + \Pi_I - 37.8\%$) was better than the accomplishment of students who did not use MIMLE (the first and third row in Table 5, $\Pi_T + \Pi_I - 29.8\%$). On the other hand, the difference between the arithmetic means in incorrect answers given by the groups that used MIMLE and those that did not use it was very small – 3.2%. It means that students in the groups that did not use MIMLE gave incorrect answers to the same extent as students in the groups that used it, which shows that students of that group were not interested in the given teaching unit regardless of the way of presenting the teaching material.

The total percentage of correct answers given by groups that used MIMLE and attended lectures ($\Pi_T - 23.0\%$) in comparison with all other groups shows that these groups showed the best results, which still gives high significance to classic lectures supplemented with MIMLE ‘Z-buffer’ module. The results on the final exam showed that the generation of students that had a chance to use MIMLE (generation 2009/10) acquired the teaching unit ‘Z-buffer’ more successfully than the generation of students who did not use such teaching material (generation 2008/09). Proportion of results accomplished by these generations can be seen in Fig. 6. On the basis of the result proportion we can conclude that the generation 2009/10 increased the number of correct answers by almost 50%, by eliminating a big number of “I don’t know” answers. The use of MIMLE, besides increasing the level of knowledge in the given teaching unit, also gave a lot of students confidence in the acquired knowledge.

The result of such learning is that students establish interaction in order to recognize the main concepts of the teaching topic and their mutual connection, with amusing environment which reminds them of playing a game. Students’ capability of making conclusions on the basis of playing games was checked in this work.

### Table 5

Characteristics of the results distribution on the final test.

<table>
<thead>
<tr>
<th>Groups</th>
<th>Correct answer (%)</th>
<th>Incorrect answer (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Pi_I$</td>
<td>11.5</td>
<td>13.1</td>
</tr>
<tr>
<td>$\Pi_T$</td>
<td>14.8</td>
<td>11.5</td>
</tr>
<tr>
<td>$\Pi_I$</td>
<td>18.0</td>
<td>4.9</td>
</tr>
<tr>
<td>$\Pi_T$</td>
<td>23.0</td>
<td>3.3</td>
</tr>
</tbody>
</table>

![Fig. 6. Results on the final exam.](image-url)
by test questions on the final exam of the course in Computer Graphics for the teaching unit 'Z-buffer'. The effort students had to make using MIMLE in the learning process showed that its usage helped students to better understand and acquire given material with minimum time spent. Students’ playing games in MIMLE lasted only 3 minutes on average, and the result on the final test for this group of questions improved by almost 50%. This was the reason why the authors of this work decided to make and apply MIMLE in other teaching units of the course in Computer Graphics in the future as well.

8. Conclusion

In this paper, we have presented the Multimedia Interactive Modules for Learning – MIMLE as a new class of applications which are an intersection of a combination of the three classes of the existing educational applications: IMMEX/Gismo simulations, simulation games and educational games. Dynamic simulations combined with educational games represent a new and better way of the learning process by using student’s knowledge and skills in interactive computer games. The approach of keeping student’s attention with the minimal usage of game elements characterizes the MIMLE system. The MIMLE system can represent teaching material in the world of games, improve already acquired skills in games context and show the task results in the form of game results.

A feature of some of the educational games is that they use pedagogical agents (Conati & Zhao, 2004) that interact with student through animated cartoon-like characters and objects. The usage of animated characters in the role of a player can either motivate agogical agents (Conati & Zhao, 2004) that interact with student in games context and show the task results in the form of game results. The effort students had to make using MIMLE in the learning process showed that its usage helped students to better understand and acquire given material with minimum time spent. Students’ playing games in MIMLE lasted only 3 minutes on average, and the result on the final test for this group of questions improved by almost 50%. This was the reason why the authors of this work decided to make and apply MIMLE in other teaching units of the course in Computer Graphics in the future as well.

The contribution of this work is that in both student’s knowledge appraisal models the pedagogical agent reacts correctly during the learning process and by doing it agent does not cause negative effects during the learning process. Students aged 18 to 20 tend to diminish the importance of help and suggestions of teachers, so we have paid special attention to the interaction of the pedagogical agent with the students of that age as end users of the given MIMLE.

References


