

TEACHING AID SUPPORTING THE TUITION OF AUTOMATION AT FACULTY OF APPLIED INFORMATICS

Karel Perutka & Petra Duchackova



This Publication has to be referred as: Perutka, K[arel] & Duchackova, P[etra] (2018). Teaching Aid Supporting the Tuition of Automation at Faculty of Applied Informatics, Proceedings of the 29th DAAAM International Symposium, pp.0085-0094, B. Katalinic (Ed.), Published by DAAAM International, ISBN 978-3-902734-20-4, ISSN 1726-9679, Vienna, Austria
DOI: 10.2507/29th.daaam.proceedings.012

Abstract

The paper is focused on the description of the newly created teaching aid designed to support auto-tutoring in the field of basics of automation for full-time and mainly for distance curriculum at the Faculty of Applied Informatics, Tomas Bata University in Zlin. It deals with the Laplace transform, the description of the properties of the linear systems and the description of the systems in the frequency domain including their characteristics in the form of the game. The game was tested by students and positively evaluated by them in the form of the questionnaire. The paper includes the brief theoretical background, the description of the game specification, the game implementation, the game from the user's point of view, the short examples of the source codes, and the results of the evaluation using the questionnaire.

Keywords: automation; computer game; MATLAB; SIMULINK; teaching aid

1. Introduction

Teaching aids, tools, or in more general, innovations in education, play an important role in the teaching process during last years. For example, there was created a nice project of entrepreneurship and innovation in engineering education in the School of Engineering of University of Minho based on developing and improving commercial products manufactured by industries [3]. The learning game to teach mechanical engineering was presented by Pons-Lelardeux et al. [4]. It also mentions the evolution of industrial design and manufacturing methods with a focus on more virtualization. Innovation in teaching financial mathematics is described in paper created by Lopes and Soares [5]. It includes the flipped classroom model as the new organizational design for the teaching and learning paradigm. Computer game as a learning and teaching tool for object-oriented programming is another area of implementation in the field of education as mention the study created by Seng and Yatim [13].

Moreover, there was created a new project for intelligent robotics for the remote training of engineers and specialists in the field of IT based on the inter-university network by Pryanichnikov, Katalinic et al. [10]. The idea that teaching of informatics in primary and secondary schools is more than programming presented Bollin et al. [8]. The role of reciprocal teaching and programmed instruction on learning outcome in computer science education was presented last year by Zandler and Reile [12].

With thirty years long history to implement digital education in Austrian primary and secondary schools, Micheuz et al. presented next level of modern education level [9]. There was also created nice paper describing how planning and task-solving processes influence the course performance of software engineering course [11].

In the field of automation and product development, there was created nice papers. For example one of them is showing the results of simulations of systems behavior [6], namely cyber-physical systems. The other describes the role of the advisor module in the hybrid assembly subordinating control structure [2] with the focus on the bionic assembly system. The design of control structures for complex manufacturing systems was presented by Katalinic [7]. It also includes the description of logistic organization planning, optimization, and scheduling of complex, flexible manufacturing systems.

There was created new teaching aid to support teaching the automation at the Faculty of Applied Informatics, Tomas Bata University in Zlin. It is described in the following chapters, and it is based on the work of Oplustilova [1] under Perutka supervision. Firstly, there is give the game specification describing the parameters of the game, its main menu, and game rules. It is followed by the description of the game from the programming perspective that includes the selected source codes of the newly created teaching aid. In the next part of the paper, the description from the user's view is presented. The final chapter shows the results of the questionnaire that evaluates the created teaching aid by students from the authors' faculty in the last year.

2. Game specification

2.1. Game description and parameters

The game "Automation Questions Quiz" is intended for all students involved in automation. It is for single player. The game contains a quiz with questions, but it also contains examples. The stand-alone game main dialog contains 16 boxes, and each box is dedicated to a particular topic. Once a question has been answered correctly, the player uncovers the box. The wrong answer to the question means the decrease in the player's lives. The player has three lives in total. If a player does not lose all three lives during a game, he or she will see the image hidden beneath the boxes.

Each question has four choice options. Only one option is always correct. Some questions include the "do not know" option. If a player answers the question using this option, it is taken as a bad answer, and the player loses one life. The Automation Questions Quiz is for all students who have some automation knowledge to know the right answers to the questions. In the game, there is also the possibility to repeat individual chapters from the subject of automation in the section "Sample examples.". You can run the game using the MenuKvizu command in the Command Window in MATLAB.

2.2. Game menu

After running the MenuKvizu command the "Automation Questions Quiz" menu will appear. The player here can choose whether to start the game, read the rules quiz to look at fixed examples or whether he/she wants to end the game.

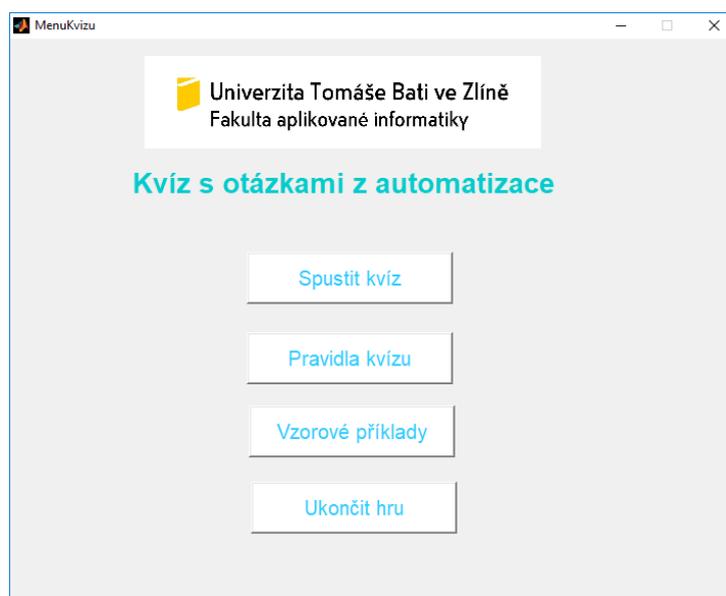


Fig. 1. Game Menu "Quiz on Automation Questions"

2.3. Game rules

The goal of the game is to answer all the questions hiding under the individual boxes and to show the hidden image. Only players decide on the order in which they will collect the boxes with questions. Each box contains some questions about a specific topic. The checkbox disappears after the player's correct answer to the question. If a player replies badly, he/she loses one life, come, but the box will disappear. The player has three lives in total. After all, lives have been lost, the game ends, and the game returns to the main menu. The number of lives is shown on the right side of the boxes. If the players answer all the questions that are hidden under all 16 boxes without losing all their lives, they will see the hidden image and win the game. Multiple topic fields contain mostly ten questions. Questions are before every game generated randomly. The game contains a total of 150 questions. Questions concern the following areas:

- Direct L-transformation
- Inverse L-transformation
- Selected L-Transformation properties
- Laplace transformation dictionary
- Decomposition of the function into partial fractions
- Linear differential equations, linear differential equations, and transmission
- Usage of poles and zeros
- Transition function and characteristic, pulse function and characteristic
- Frequency transmission and frequency characteristics



Fig. 2. Description of rules

3. The game from a programming perspective

3.1. Description

First, we designed the structure of the game. The game was developed in MATLAB version R2016a. For the stand-alone game "Automation Questions Quiz" we created one main GUI file. For the menu, rules, sample examples, individual questions, etc., we created additional GUI files. These files are linked to M-files. We created one new checkup feature to make the program more transparent to keep the source code from repeating several times.

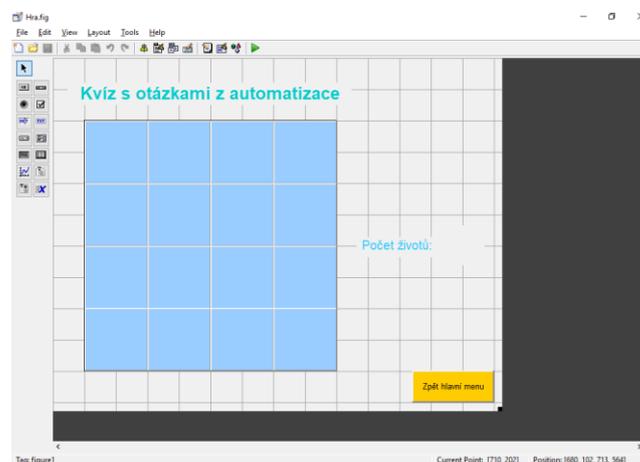


Fig. 3. Graphical game window created with GUIDE

The individual M-files are interconnected so that when the M-file ends, the second one starts, and vice versa. M-files are intertwined to each other.

3.2. Database of questions and answers

The database of questions and answers was based on our experience gained during the work with MATLAB and Simulink and also with the help of literature. For each topic, we created a new text file. In these files, the questions are written as questions, four answer options, the right answers. Question is listed as a chain, where instead of the spacebar is used the sign “;”

Before the player sees the selected question, the questions from the text file will be uploaded into the `fid` variable. When the user opens the question database file with using the `fopen` command, we are using the `strsplit` function divided the file string. We created a new variable that contains a random question from a set of questions. We used the `strrep` function that searches for the character “;” and replaces this sign by space. We also used this feature to display special character. In each question set, there are approximately ten questions that are randomly selected by `randi` function before viewing the question. Finally, the questions database file is closed. The sample code of database file is listed below.

Fig. 4. Example of a question database, selectable options, and the right answer

```
fid = fopen('otazka01.txt');
radek = 0;
while ~feof(fid)
data = fgetl(fid);
radek = row + 1;
OnDruhou = char(178);
NaTreti = char(179);
pom = strsplit(data); % function for string breakdown
for i = 1:6
    C{radek,i} = pom(i);
    C{radek,i} = C{radek,i}{1};
end
% sign replacement function - per space
C{radek,1} = strrep(C{radek,1}, ';', ' ');
for i = 1:6
    C{radek,i} = strrep(C{radek, i}, '*', NaDruhou);
    C{radek,i} = strrep(C{radek, i}, '#', NaTreti);
end
end
setappdata(0, 'questions', C);
radek = randi(10);
fclose(fid);
```

3.3. File game.m

This M-file serves as the main function of the program. Player uses this file to launch the entire game. The main function is `Game_OpeningFcn`, which loads and sets the image and sets the visibility of objects to be displayed or to be invisible in the window. It sets up a global variable for player's lives and global auxiliary variable defining the number of messages to be displayed. The image is set here by the `imshow` function (`imshow (imread('obrazek.jpg'),'Parent',handles.obrazek)`). This file contains additional features that are described below. The image used in the program was created in another program - Gimp 2.

The `otazka1_Callback` is the function where the object is set to invisible. When you call this feature, the object checkbox labeled as `question1` sets to invisible and similarly to figure (the game window is set to invisible). Invisibility is set using the `set` function, which first finds the desired object using `findobj` and then calls the `set` function. This function is the same for other objects (from `otazka1_Callback` function to `otazka16_Callback` function). The sample code is listed below.

```
set(handles1, 'Visible', 'off');
fig= findobj('Type', 'Figure', 'Tag' 'figure1');
set(fig, 'Visible', 'off ');
otazka01;
```

`BackHeavyMenu_Callback` - a function that asks the player if he/she wants to quit the game. After choosing an object back from the main menu, a dialog will appear asking if the user wants to leave the game. This dialog is displayed using the `questdlg` function. If a player answers that he/she wants to end the game, the quiz window closes, and the game menu is displayed. If he/she answers "No," he/she will continue the game. A code sample is shown below.

```
konec = questdlg ('Are you sure you want to end the game?', ...
'End of game', ' Yes', 'No', 'No');
if strcmp(konec, 'Yes')
    delete(Hra);
    MenuKvizu;
else
    set(handles.zpetHlavniMenu, 'Visible', 'on');
end
```

`figure1_CloseRequestFcn` - a feature that asks the player if he/she wants to end the play. When you mark a cross in the top right corner, a dialog box appears asking if the user wants to leave the game. This dialog is also displayed using the `questdlg` function. If a player answers that he/she wants to end the game, the quiz window closes, and the game menu is displayed. If he/she answers "No," he/she will continue the game.

3.4. File otazka01.m

This M-file is the same as other M-files `otazka02.m` to `otazka16.m`. In these files, there are open questions and retrieve individual files from a database of questions. The questions are stored in the files with `txt` extension. Questions are saved in files by topic. These files can be used later to add additional questions. This file contains several features.

`otazka01_OpeningFcn` - A function in which the question database file is opened and later retrieved. The way data is retrieved and how random choices are made is mentioned in the questions and answers database section of the game. Once the data from the file is retrieved, the individual response options are set to individual objects and buttons using the `set` function. This function sets the correct answer to a variable `spravnaodpoved`, which is used for the evaluation of the questions. The program then saves the correct answer to the `setappdata` variable that is listed below.

```
C = getappdata (0, 'otazky');
set(handles.otazka, 'String', C{radek,1});
set(handles.moznostA, 'String', C{radek,2});
set(handles.moznostB, 'String', C{radek,3});
set(handles.moznostC, 'String', C{radek,4});
set(handles.moznostD, 'String', C{radek,5});
spravnaodpoved = C{radek,6};
setappdata(0, 'odpoved', spravnaodpoved);
```

optionA_Callback - function in which the values of global variables are for the lives. The result of the question is set here in the result variable and selection of options (in this case, the object labeled as Options) player to a variable vyber. This variable is evaluated in the control function (result, selection, life, helper). After selecting this object, the question window closes. Sample code is shown below.

```
global zivoty;
global pomocna;
zivoty = getappdata(0,'zivoty');
pomocna = getappdata(0,'pomocna');
vysledek = getappdata(0,'odpoved');
vyber = get(handles.moznostA , 'String ');
closereq;
kontrola( vysledek,vyber,zivoty,pomocna);
```

figure1_CloseRequestFcn - a feature that displays a message in which the player is informed that he/she must answer the displayed question and must choose one of the suggested options. The message is displayed using the msgbox function; the code is listed below.

```
nelze = msgbox ('You did not answer the question, you have to answer ...');
uiwait(nelze);
```

3.5. File kontrola.m

This M-file was created for us because of the better orientation in the source code. Part of this feature would be repeated many times in the program. That's why we used the creation of our control function. In the first part of this function, we check the global variable for the result (stored in the variable is the correct answer), and for a global variable selection (the variable is stored in the option chosen by the user). We are using the if condition. If the chosen player will be the same as the correct answer, so the global variable lives (the stored life value in the variable) does not change. If the result is not equal to the selection, then variable lives is decreased by 1. In both cases, the auxiliary variable is used to evaluate the total number of answered questions. After evaluation, questions set variable lives, and its value is displayed in the appropriate object.

The next section checks to see if the number of lives has passed. If the value of the lives is equal to zero then using the msgbox function the message appears on the screen that the player has lost. The quiz window closes, and the game menu window opens. The last part of the check is to check if the player has answered all the questions. This task is done by a global variable pomocna. The value of the variable increases each time when the question is answered. After the value of this variable is equal to sixteen, the window will display the message that the user has won the game. The msgbox function is used again, and the source code is listed below.

```
function kontrola(vysledek,vyber,zivoty,pomocna)
if strcmp(vysledek,vyber)
    zivoty = (zivoty + 0);
    setappdata(0,'zivoty',zivoty);
    pomocna = (pomocna + 1);
    setappdata(0, 'pomocna', pomocna);
else
    zivoty = (zivoty -1);
    zivoty = (zivoty + 0);
    setappdata(0,'zivoty',zivoty);
    pomocna = (pomocna + 1);
end
zivotPom = findobj('Tag','PocetZivotu');
set(zivotPom, 'String', zivoty);
if zivoty == 0
    Prohra = msgbox ('You've lost the number of lifes, end of game. ');
    uiwait(Prohra);
    delete(Hra);
    MenuKvizu ;
else
```

```

    fig= findobj('Type', 'Figure', 'Tag', 'figure1');
    set(fig, 'Visible', 'on');
end
if pomocna == 16
    vyhra = msgbox('You have won!');
    uiwait(vyhra);
    delete(Hra);
MenuKvizu ;
else
    fig= findobj('Type', 'Figure', 'Tag', 'figure1');
    set(fig, 'Visible', 'on');
end

```

3.6. File MenuKvizu.m

This file is used to switch between individual windows - game, game rules, solved examples, etc. It contains the following functions: MenuKvizu_OpeningFcn, RunKviz_Callback, UkoncitHru_Callback. The close function is used to close the window and open the second window.

4. The game from the user's view

The user starts the game using the MenuKvizu command the MATLAB Command Window. The application runs in any version of MATLAB. After the command has been entered, the main game menu window opens.

Game menu - In the game menu, the user can choose whether to run the game, read the game rules, look at the solved examples, or quit the game. The user can select the option only with the mouse. The user moves the mouse over the button and clicks on it.

Game Rules - When selecting the Game Rules, the user will see a new window describing the rules of the game. Here is the Back button to return to the main menu. Control of this button again works only with the mouse by clicking on the button. Clicking on the cross in the top right corner closes the window and ends the game.

Quit the game - When selecting Exit Game, the main game window closes. You can also end the game in the main menu using the cross in the upper right corner.

Sample examples - If a user chooses Sample Examples option, a new window will appear, where the user can view the progress of the task from the number of the tasks.

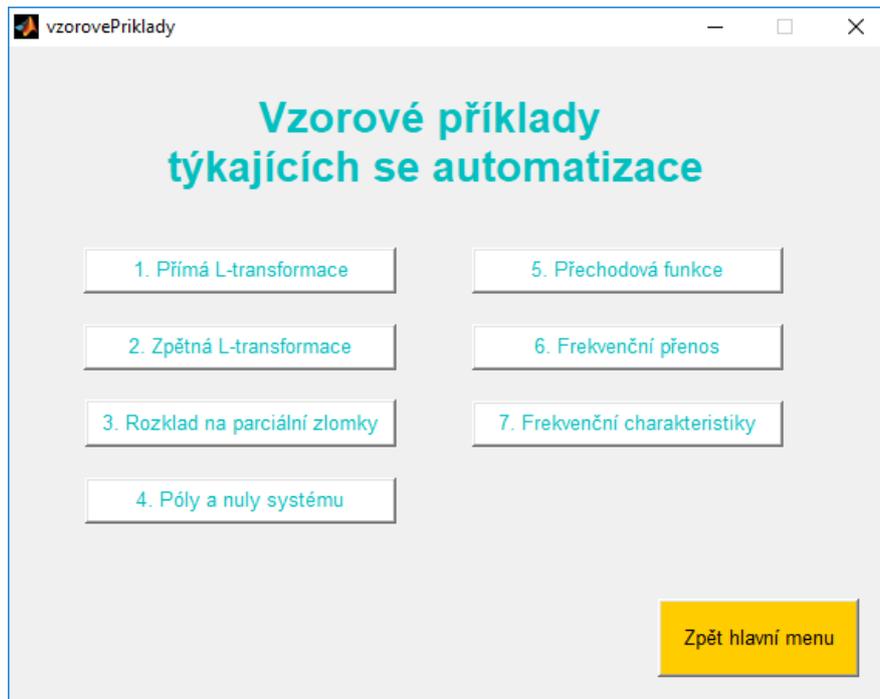


Fig. 5. Examples of automation issues

The control is the same as with the previous windows using the mouse. After selecting from several examples of sample examples, a new window opens to solve the problem. Using the Back Menu button, the user will get back to the game menu.

Game - When selecting this option, the user will see a newly uploaded game window.

The right part of the window shows the number of lives. Then it is only for the user who chooses the box. When the player clicks on the checkbox and clicks on the mouse, the player will see a random question. The player can choose from four options. Only one option is always correct.

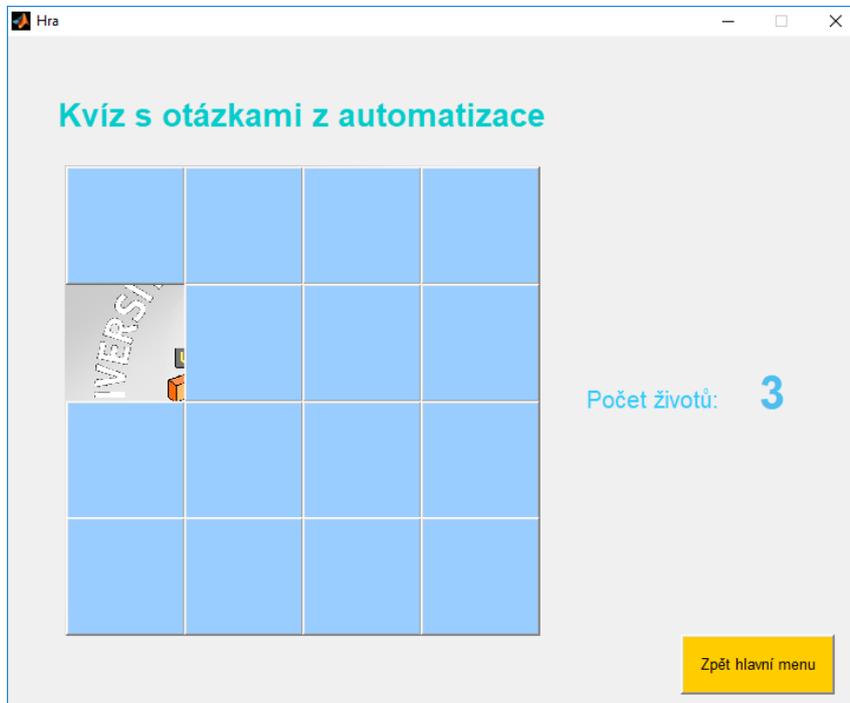


Fig. 6. Window with the main game

After choosing one from four options, the question window closes, and the game opens again with individual fields. If the player has answered the question not correctly, then the life of the player is deducted in the right part of the window. After answering a question, players will always uncover a portion of the image under the boxes. If a player loses all his or her lives, a window will appear in which the player is informed of the loss. Once the OK button has been selected, the player uses the mouse to return to the game menu.

If a player answers all the questions in the right order without losing all the lives, the player reveals the hidden image and displays an information window in which the player is informed then he/she wins. Once the OK button has been selected, use the mouse to return to the game menu. If a player chooses the Back button of the main menu, a dialog box will appear asking the player if he/she wants to leave the game. If the player selects Yes button, the game is restarted. After selecting the No button, the dialog box closes, and the game window is displayed. If a player selects the cross button in the top right corner of the game, he or she will display a dialog asking the player if he or she wants to leave the game.

5. Evaluation using Questionnaire

The following questionnaire was created in purpose to evaluate the created teaching aid by students of regular and distance curricula. The questions were the following ones:

- 1) Have you used the created teaching aid?
- 2) If so, did it help you?
- 3) Would like to enlarge the tool?
- 4) Would you like to use this aid in the different subject of the curriculum?

The obtained results are displayed below.

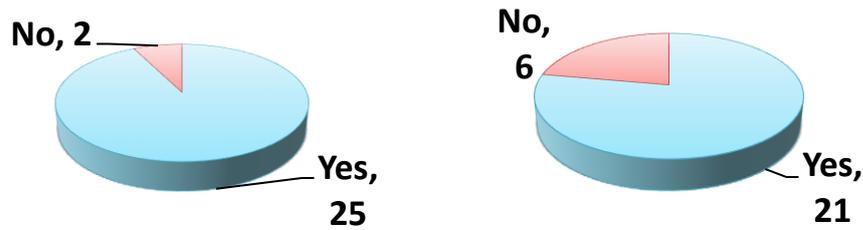


Fig. 7. Answers on Question 1 (left) and Question 2 (right)

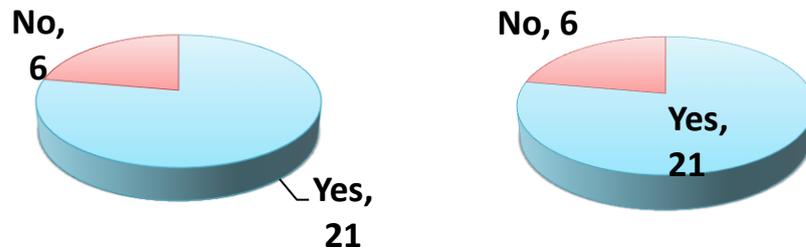


Fig. 8. Answers on Question 3 (left) and Question 4 (right)

It is clear from the presented results of the questionnaire that in general students positively evaluated created tool. Moreover, they would like to use it in the different subject and, mostly, they would also like to enlarge the existing database of questions.

6. Evaluation of created solution

The created game became the part of new modern educational aids according to the modern strategy of teaching that implements the lectures and practices in the attractive form. Games are leading applications in this area. The students confirmed that they are more attracted to the studies. The tests that verified the knowledge of students from the area of automation confirmed the better knowledge of students who played the game in comparison to those who did not use this game.

The issue of the paper was to solve the problem – to find something to increase the knowledge of basic automation at the faculty in the given curricula. It is important to understand the game functionality, and it is also important to have the knowledge of MATLAB and SIMULINK in purpose to finish the created game. The contents of the game are limited to the basic automation that MATLAB and SIMULINK offer.

7. Limitations and future plans

Although the paper has reached its aims, there were some limitations. First, because the game is new, it was verified only by a small group of students who were attending the classes. Therefore, it is necessary to use it for some years to generalize the result and to find what to improve and enlarge. Second, the content of the game is specific. It was created for the given curriculum, so it is necessary to modify or enlarge it in purpose to use this game by users from another faculty.

Future work is divided into several parts. Firstly, it is necessary to use the game for more years in purpose to evaluate the game from the users' point of view. Lately, there will be realized necessary correction based on the experience with the teaching aid.

8. Conclusion

The necessity to attract the study curricula about automation was the main problem that had to be solved. It was solved by a modern approach using the game as the idea of teaching aid. It leads to newly created application practicing the student knowledge of automation using the quiz. This application was positively evaluated by students. Students used it before final test from automation in the selected subjects. It covers the whole introductory course to automation and examples on how to solve them.

Future work will be oriented on the more complex enlargement of the newly created application. The enlargement will be focused on the higher increase in the number of topics and questions in the database and on the implementation in more subjects of the existing curricula.

9. References

- [1] Oplustilova, P. (2018). Application Design to Support Automation Training for Full-time and Distance Curricula, Thesis, Faculty of Applied Informatics, Tomas Bata University in Zlin, Zlin, Czech Republic
- [2] Haskovic, D., Katalinic, B. & Kukushkin, I. (2015). Role of the Adviser Module in the Hybrid Assembly Subordinating Control Structure, *Procedia Engineering*, Vol. 100, 2015, pp. 1706-1713, ISSN 1877-7058, DOI 10.1016/j.proeng.2015.01.546
- [3] Soares, F. O., Sepúlveda, M. J., Menteiro, S., Lima, R. & Dinis-Carvalho, J. (2013). An integrated project of entrepreneurship and innovation in engineering education, *Mechatronics*, Vol. 23, 2013, pp. 987-996, ISSN 0957-4158, DOI 10.1016/j.mechatronics.2012.08.005
- [4] Pons-Lelardeux, C., Galaup, M., Segonds, F. & Lagarrigue (2015). Didactic study of a learning game to teach mechanical engineering, *Procedia Engineering*, Vol. 132, 2015, pp. 242-250, ISSN 1877-7058, DOI 10.1016/j.proeng.2015.12.476
- [5] Lopes, A. P. & Soares, F. (2018). Perception and performance in a flipped Financial Mathematics classroom, *The International Journal of Management Education*, Vol. 16, 2018, pp. 105-113, ISSN 1472-8117, DOI 10.1016/j.ijme.2018.01.001
- [6] Canadas, N. Machado, J., Soares, F., Barros, C. & Varela, L. (2018). Simulation of cyber physical systems behaviour using timed plant models, *Mechatronics*, Vol. 54, 2018, pp. 175-185, ISSN 0957-4158, DOI 10.1016/j.mechatronics.2017.10.009
- [7] Katalinic, B. (1992). Design of Control Structures for Complex Flexible Manufacturing Systems, *IFAC Proceedings Volumes*, Vol. 25, No. 8, 1992, pp. 26-33
- [8] Bollin, A., Pasterk, S., Antonitsch, P. & Sabitzer, B. (2016). Software Engineering in Primary and Secondary Schools – Informatics Education is More Than Programming, *Proceedings of 29th IEEE Conference on Software Engineering Education and Training*, Dallas, USA
- [9] Micheuz, P. Pasterk, S. & Bollin, A. (2017). Basic Digital Education in Austria. One Step Further, *Proceedings of 2017 World Conference on Computers in Education (WCCE)*, Dublin, Ireland
- [10] Pryanichnikov, V. E., Katalinic, B., Kirilchenko, A. A., Khelemendik, R. V., Kuvshinov, S. V., Vician, D. & Uglesic, A. (2015). New Creative Educational Technologies for Inter-University Network, *Procedia Engineering*, Vol. 100, 2015, pp. 259-268, ISSN 1877-7058, DOI 10.1016/j.proeng.2015.01.366
- [11] Bollin, A., Reci, E., Szabó, S., Szaboóva, V. & Seibenhofer, R. (2018). Applying a maturity model during a software engineering course – How planning and task-solving processes influence the course performance, *The Journal of Systems & Software*, Vol. 144, 2018, pp. 397-408, ISSN 0164-1212, DOI 10.1016/j.jss.2018.07.009
- [12] Zandler, A. & Reile, S. (2018). The effect of reciprocal teaching and programmed instruction on learning outcome in computer science education, *Studies in Educational Evaluation*, Vol. 58, 2018, pp. 132-144, ISSN 0191-491X, DOI 10.1016/j.stueduc.2018.05.008
- [13] Seng, W. Y. & Yatim, M.H.M. (2014). Computer Game As Learning and Teaching Tool For Object Oriented Programming in Higher Education Institution, *Procedia – Social and Behavioral Sciences*, Vol. 123, 2014, pp. 215-224, ISSN 1877-0428, DOI 10.1016/j.sbspro.2014.01.1417