The 12th International Scientific Conference
eLearning and Software for Education
Bucharest, April 21-22, 2016
10.12753/2066-026X-16-073

TEACHING PROGRAMMING BY DEVELOPING GAMES IN ALICE

Adrian FLOREA, Arpad GELLERT
"Lucian Blaga" University of Sibiu, Computer Science and Electrical Engineering Department, Emil Cioran Street No.4, Sibiu, Romania
adrian.florea@ulbsibiu.ro, arpad.gellert@ulbsibiu.ro

Delilah FLOREA, Adrian-Cristian FLOREA
"Samuel von Brukenthal" National College of Sibiu, Informatics and Computer Science Department, Huet Square, No. 5, Sibiu, Romania
delilah_florea@yahoo.com, acflorea99@gmail.com

Abstract*: What do we teach? How do we teach? Who do we teach? These three interrelated questions are not necessarily new, but are fundamental in terms of reshaping the teaching-learning process in order to accomplish the educational needs of a digital native society. Current solutions rely on merging traditional ways of teaching with modern approaches such as e-Learning, interactive teaching, open-ended questions, collaborative team-work, professional challenges and competitions in order to require student's brainpower. The concept of learning by doing is extended to learning by gaming. The main goal of this paper is to propose a solution to a fundamental problem in Computer Engineering Education: attracting students, capturing the interest and retaining them in order to acquire a profession in this area. Specifically, students will learn the fundamental concepts of procedural and object-oriented programming in Java by developing games in Alice 3.0 – a 3D interactive animation environment which uses text-based programming in order to create visual programming blocks. Based on advanced hardware-software technologies, our game represents a pleasant learning alternative to the traditional education system. Although it is suitable for beginner learners of any age greater than 8, this software application may be especially engaging for girls. The software application is free, easy to extend, and additionally, being built as a game develops specific skills such as ambition, desire to win, strategic thinking, motivation and perseverance in order to improve player's performance, understanding the solving methods and the joy of discovering new things. The usefulness of this paper is twofold: first, for teachers who, starting from the John C. Dana quote's "who dares to teach must never cease to learn", will be able to implement new teaching methods, applying thus the concept of lifelong learning in their didactic, scientific and pedagogic activity; second, for students, who can become active in the learning process by creating interactive educational games.*

Keywords: *Alice; game; teaching; learning; education; programming.*

I. INTRODUCTION

Education is the evolution engine of modern society, and education through programming, with proper tools of Information Technology and Communications can be considered the primary means of changing the mentality of the population. Researchers in the science education consider that teaching young generation ("digital natives") to gain a computational and critical thinking is much more important than teaching them specific professional skills [1]. *What do we teach? How do we teach? Who do we teach?* These three interrelated questions are not necessarily new, but are fundamental in terms of reshaping the teaching-learning process in order to accomplish the educational needs of a digital native society. In this work, we try to answer these questions from the teacher's perspective in the field of Computer Engineering Education.

"What do we teach?" – Basically, disciplines and fundamental concepts (algorithms, data structures, object oriented programming (OOP)) and software programming technologies. American specialists insist on integration of programming as a means for modelling and simulating relevant scientific topics from biology, chemistry, physics, etc., using visual programming [2]. Major local and global organizations such as Oracle and Adfaber.org [3, 4], Code.org [5] supported by Microsoft, Apple, Amazon, The Khan Academy, Wolfram Research etc., encourage teaching algorithms and programming languages as it promotes critical and computational thinking, useful in many fields: mathematics, engineering, law, culture, economics, medicine, sports, entertainment, etc. [1]. On the other side, the researchers suggest that the future in Computer Engineering Education must be addressed in a cultural-scientific integrated manner, not just purely technically [6].

"How do we teach?" – To answer this second question we cannot ignore the third: "Who do we teach?", in other words what is the target audience to whom we address? What are the concerns specific of the age, how deep is rooted the sense of responsibility? How much involved in teaching are the students and what kind of temptations have students to face against, in learning process?

"How do we teach?" – Current solutions rely on merging traditional ways of teaching based on the blackboard, chalk and monologue exposure in front of students, with modern approaches such as e-Learning, focused on online multimedia applications and technologies, developed in a collaborative manner, interactive teaching, open-ended questions, collaborative team-work, professional challenges and competitions in order to require students' brainpower. Here is determinant the psychosocial task of the professors and their responsibility regarding mastery of concepts and modern theories of cognitive development, capitalization of methods and techniques of knowledge, and students' stimulation for intensive work, involvement of teachers in exciting activities together with students. The concept of learning by doing is extended to learning by gaming. A learning alternative accepted by digital natives is not only based on learning exercises (learning by doing) but, if is possible through games (learning by gaming), by engaging teachers beside students to these training activities.

The main goal of this paper is to propose a solution to a fundamental problem in Computer Engineering Education: attracting students, capturing the interest and retaining them in order to acquire a profession in this area. Specifically, students will learn the fundamental concepts of procedural and object-oriented programming in Java by developing games in Alice 3.0 – a 3D interactive animation environment which uses text-based programming in order to create interactive and visual programming blocks. Our choice is motivated by the fact that, according to IT&C market research [3, 7], 89% of Desktops in the U.S.A run Java applications, there are more than 9 million Java developers worldwide that deploy video games and smart applications, use exciting programs and services, and over 3 billion mobile phones executes Java bytecode. By this paper, we aim to change the perception of Romanian society on education, namely education in Computer Science, to draw attention of students about the importance of IT&C, presenting computing technologies and programming through an applicative approach which differs by the traditional methods, based on promoting the development of software tools and educational games for all teaching topics from any science field. At the same time, we want to prove that anyone can learn programming in Java, starting with simple commands until creating games with 3D animations. Our game represents a pleasant learning alternative to the traditional education system. The advantage of this software application is that it is free, extensible and additionally, being created as a game, develops specific skills: ambition, desire to win, strategic thinking, etc. Furthermore, by a mechanism of reward, the students are motivated and want to play again in order to improve their performance, skills, understand solving methods and discover new things. The developed game is dedicated to any student or beginner learners, because the Alice environment is friendly, created on a tale background with fantastic characters, awakening interest both among boys, but especially for girls, who thus will be motivated in choosing their careers in IT&C domain. Moreover, a global requirement aims the learning, promoting and encouraging women to choose future jobs among STEM professions (Science, Technology, Engineering and Math) and are made many efforts to motivate women to succeed in these fields [8, 9].

The organization of the rest of this paper is as follows. In section 2 we shortly review the Related Work in the field of educational software and mainly of game-based learning. Section 3 describes a short background related to Alice environment, whereas section 4 presents the game's design: the storyboard, the graphical user interface and the software developer's vision. In section 5, we evaluate the satisfaction level of students regarding our game-based teaching approach. Finally, section 6 suggests directions for future work and concludes the paper.

II. RELATED WORK

There is a big project that aims to provide education to anyone for any topic. That project is KhanAcademy (http://www.khanacademy.org). There are also other papers that discuss about e-learning. One of them is an educational web game framework about a comparison game that helps young children to get a sense about quantity [10]. In [11] is presented a computer game for training approximate number system in order to improve math abilities of peoples suffering of dyscalculia.

Our previous experience about developing games is presented in [12] and consists in implementing an e-Learning platform for improving the teaching and learning process in somewhat abstract domains, such as computer architecture or object oriented programming, with the help of games. These games are time-dependent and are able to support collaboration between groups. There are two learning games implemented: a crossword puzzle and a collaborative jigsaw puzzle, the last one supporting multiplayer mode for up to 16 simultaneous players, being simple, fast, fun and reliable. The application allows geographically distributed students to concurrently and collaboratively play the same game. In [13] we have developed MiniGL, a tool used by students and teachers at Java and Games Programming laboratories from academia. We have implemented in Java under Eclipse framework some educational games for math learning and for testing the cognitive capacity of memory. The applications run on Android mobile phone or on desktop computers having Android emulator with a newer version than 2.2 (Froyo).

In [14] Dann et al. developed and tested teaching techniques and instructional materials for a college-level programming course using Alice 3 and Java. They applied the educational theory of mediated transfer to develop a new version of Alice. The student test scores of the experimental course sections had at least one grade improvement over the test scores of the traditional course sections.

In [15] Wang et al. investigated the feasibility of using Alice in teaching the programming concepts in high schools, compared to C++. The students' test scores showed that the Alice group outperformed the C++ group in the comprehension of the fundamental programming concepts, but the motivation was the same.

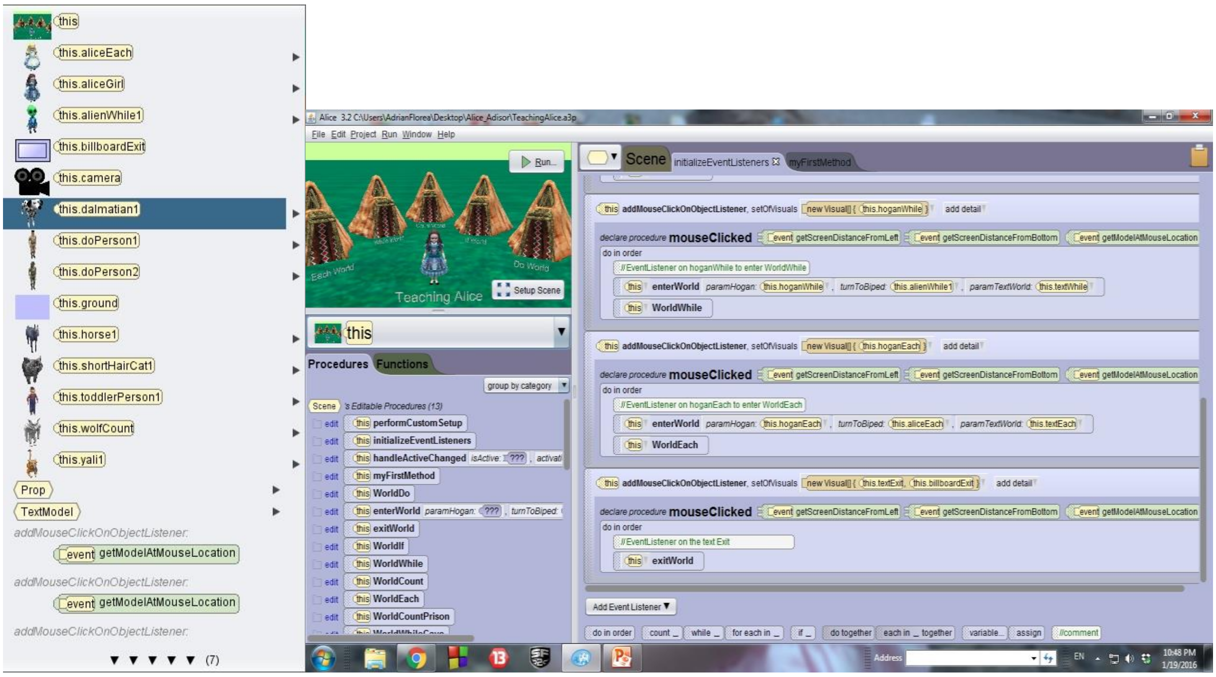
In [16], Magee and Han present a method to teach a university-level introductory computer science course by using Alice 3D animation-based programming environment followed by event-based graphical Java programming. The interactive and visual nature of these approaches has improved the retention in the follow-up courses.

III. ALICE: GENESIS, CHARACTERIZATION AND BENEFITS

Alice is an educational open-source environment used for teaching introductory Java programming. In accordance with ACM/IEEE Computing Curricula 2001 [17], Alice is the promoter of educational strategy "objects-first" or "early-objects" which considers that, teaching object oriented programming by discussing objects, methods, events from the very start represents a good thing. Its genesis was in 1998 at Carnegie Mellon University [17, 18]. In 2004 Alice 2.0 has been released and nowadays the version 3.2 is available with variants both for Windows 32-bit and 64-bit, Mac and Linux. In contrast to the other currently available tools for teaching Java, Alice is designed for beginner students who wish to learn object-oriented programming and Java, and to learn how to develop programs that animate 3D objects in a virtual world using fun, imagination and engaging tools. The virtual worlds contain three kinds of 3D objects: settings, actors and background template. Alice's smart interface helps you to drag-and-drop program elements (if/then statements, loops, variables, etc.) in a mouse-based editor in order to program the virtual world through "visible" actions like "*turn, moveTo, place, roll, resize*, etc." avoiding syntax errors. As all the modern programming environments, Alice supports methods, functions, variables, parameters, recursion, arrays, and events. After designing the scene based on one or many virtual worlds, Alice will render the animation.

The Alice gallery consists in over 700 objects and characters. The current Alice world objects are displayed in an object tree. The developers can select objects from the gallery and put them into the scene, edit or change their properties. In our application, we use soldiers, aliens, cats, wolfs, dogs, rocks and obviously Alice, the main actor (see figure 1a). The scene editor allows students to set up objects in their 3D worlds. If the number of objects increases, more control statements must be defined in order to better control animation timing. Students use events to associate methods with mouse clicks, object collision, etc. For example, figure 1b shows several methods (*enterWorld*, *WorldEach*, *exitWorld*, *WorldWhile*, etc) that are triggered sequentially on Mouse Click event. Some of the visual effects produced after pushing the Mouse button are visible in figures 6 and 7. The details area displays methods, functions, procedures, and data for the selected object (see the down left corner of figure 1b). Students can build programs by dragging methods from the details area.

One advantage of creating games in Alice is based on the NetBeans plugin that allows the developer to import the Alice project (the TeachingAlice.a3p program) into the NetBeans IDE (https://netbeans.org/), and continue to code and run the program.



```
this addMouseClickOnObjectListener, setOfVisuals new Visual[] { this.hoganWhile } add detail
declare procedure mouseClicked  event getScreenDistanceFromLeft  event getScreenDistanceFromBottom  event getModelAtMouseLocation
do in order
    //EventListener on hoganWhile to enter WorldWhile
    this enterWorld paramHogan: this.hoganWhile , turnToBiped: this.alienWhile1 , paramTextWorld: this.textWhile
    this WorldWhile

this addMouseClickOnObjectListener, setOfVisuals new Visual[] { this.hoganEach } add detail
declare procedure mouseClicked  event getScreenDistanceFromLeft  event getScreenDistanceFromBottom  event getModelAtMouseLocation
do in order
    //EventListener on hoganEach to enter WorldEach
    this enterWorld paramHogan: this.hoganEach , turnToBiped: this.aliceEach , paramTextWorld: this.textEach
    this WorldEach

this addMouseClickOnObjectListener, setOfVisuals new Visual[] { this.textExit, this.billboardExit } add detail
declare procedure mouseClicked  event getScreenDistanceFromLeft  event getScreenDistanceFromBottom  event getModelAtMouseLocation
do in order
    //EventListener on the text Exit
    this exitWorld
```

```
Scene 's Editable Procedures (13)
edit  this performCustomSetup
edit  this initializeEventListeners
edit  this handleActiveChanged isActive: ??? , activati
edit  this myFirstMethod
edit  this WorldDo
edit  this enterWorld paramHogan: ??? , turnToBiped:
edit  this exitWorld
edit  this WorldIf
edit  this WorldWhile
edit  this WorldCount
edit  this WorldEach
edit  this WorldCountPrison
```

Figure 1 a) The object tree used in our game **b)** Associating methods by mouse clicks

IV. TEACHING ALICE: The Storyboard.

4.1 The Developer's Perspective

From the programmer's viewpoint, the first step aims to integrate research results about technical and pedagogical aspects in the development process of creative games used in formal and informal education, in order to allow software designers to effectively deploy games that improve learning process. This research consists in:

- Studying of interactive virtual environments for learning: opportunities and challenges;
- Introducing the new trends in digital design of educational games;
- The risk assessment in software implementation of digital gaming.

The next step will focus on shaping and developing the basic game components (in our case within the Alice software platform). From a practical viewpoint, writing an animation program supposes first composing / reading of a scenario (a description of the story, game, or simulation) and deciding how to continue to create the animation (design a storyboard, adding the characters and their dynamic movements, analysis of interaction between actors). Then, we implement the program code and finally we test the program by running the animation and correct the identified errors.

Since the purpose of this work is the development and implementation of educational games with and for students, the next step involves applying games in classroom under teacher supervising, aiming:

- Learning by students of the fundamental concepts of procedural and object-oriented programming in Java language by running some games in Alice 3.0.
- Developing new applications by each student based on acquired knowledge.
- Assuming the programmer role of each student, depending on the level of study.

4.2 The User Guide Interface

Teaching Alice is an educational program that wants to teach the basics about Alice 3.0 and Java. To make this happen, it is necessary to understand what each instruction is doing. Alice girl will guide through the journey. The learners will have fun creating animated stories and games using characters and various objects from a rich gallery of 3-D models.

In the beginning, Alice is in the front of the camera and behind here are 5 hogans that represent the entry to each world, whose names are written on the ground, in front of them. By clicking on a hogan, Alice will enter the regarded world. A fog will cover the screen, simulating in this way the journey to that world. After entering the world, Alice will be taught about the selected instruction through an example.

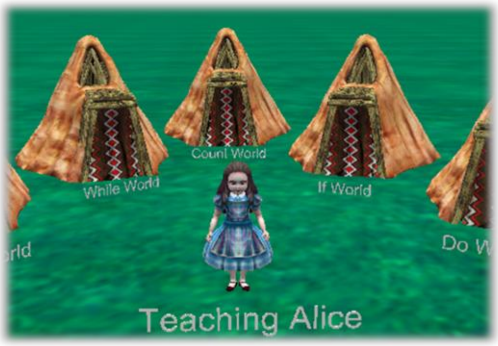
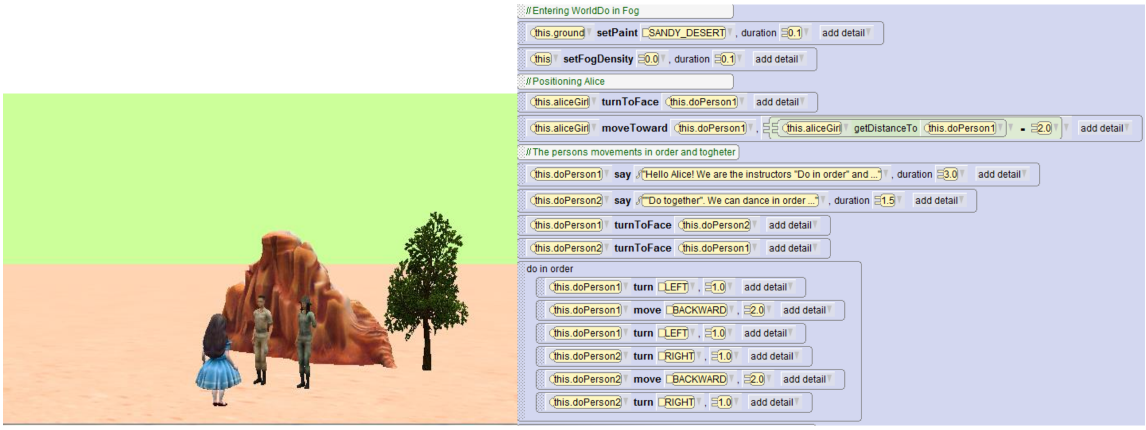


Figure 2 The main entrance in *Alice World*

World Do

In "World Do", Alice will learn what the instructions "*do in order*" and "*do together*" are doing. In this world, we put two soldiers to dance separately first, and secondly, together. This way they will show that instructions can be executed in order (sequential) or together (concurrently, in parallel). Figure 3a illustrates the scene which appears when the user enters in World Do while Figure 3b shows the source code required to implement the scene.



```
// Entering WorldDo in Fog
this.ground setPaint SANDY_DESERT , duration 0.1 add detail
this setFogDensity 0.0 , duration 0.1 add detail
// Positioning Alice
this.aliceGirl turnToFace this.doPerson1 add detail
this.aliceGirl moveToward this.doPerson1 , this.aliceGirl getDistanceTo this.doPerson1 - 2.0 add detail
// The persons movements in order and togheter
this.doPerson1 say "Hello Alice! We are the instructors "Do in order" and ..." , duration 3.0 add detail
this.doPerson2 say ""Do together". We can dance in order ..." , duration 1.5 add detail
this.doPerson1 turnToFace this.doPerson2 add detail
this.doPerson2 turnToFace this.doPerson1 add detail
do in order
    this.doPerson1 turn LEFT , 1.0 add detail
    this.doPerson1 move BACKWARD , 2.0 add detail
    this.doPerson1 turn LEFT , 1.0 add detail
    this.doPerson2 turn RIGHT , 1.0 add detail
    this.doPerson2 move BACKWARD , 2.0 add detail
    this.doPerson2 turn RIGHT , 1.0 add detail
```

Figure 3 a) Alice in *Do World* – The Scene **b)** Source code for programming *Do World*

World If

In “World If”, Alice will learn what the instruction “if” is doing. The child “Baby If” can’t decide whether he is hungry or thirsty. Alice will explain him that if he’s hungry he can eat cake else, he can drink juice.

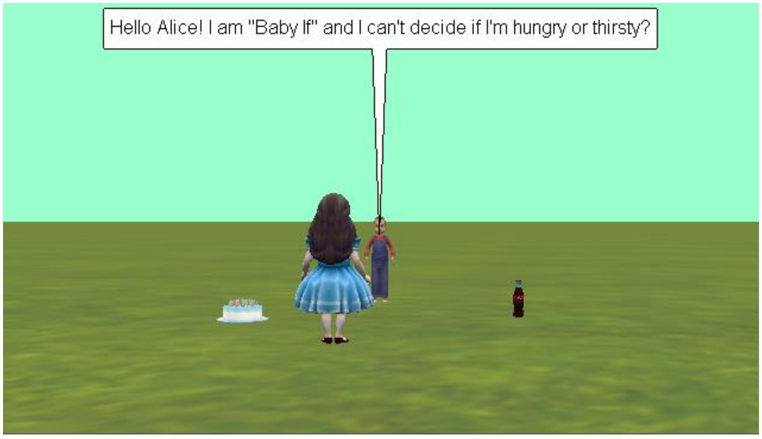

Hello Alice! I am "Baby If" and I can't decide if I'm hungry or thirsty?

Figure 4 Learning decision instruction in *World If*

World Count

In “World Count”, Alice will learn what the instruction “count up to” is doing. There, she will meet “Wolf Count” which will make Alice his prisoner, if she cannot answer correctly all its questions. Alice needs to follow the count up of the wolf and answer separately “5”, “13” and “32” if she wants to come back. If she answers wrong once, she will be a prisoner in the castle of the Wolf.

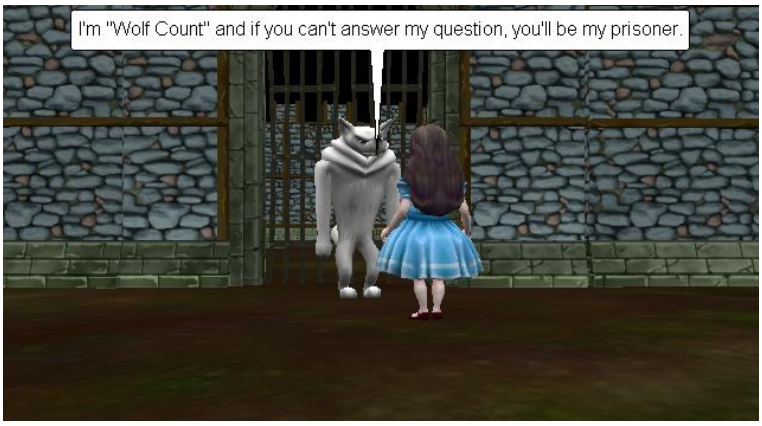

I'm "Wolf Count" and if you can't answer my question, you'll be my prisoner.

Figure 5 Learning repetitive instruction in *World Count*

World While

In “World While” (see figure 6), Alice will learn what the instruction “while” is doing. There, she will meet “Alien While” who will play a game called “Find the Alien”. While Alice is turned around, the Alien will hide behind a rock and the rocks will switch positions. Alice needs to answer where she thinks he is hidden. The Alien is hidden “left”.

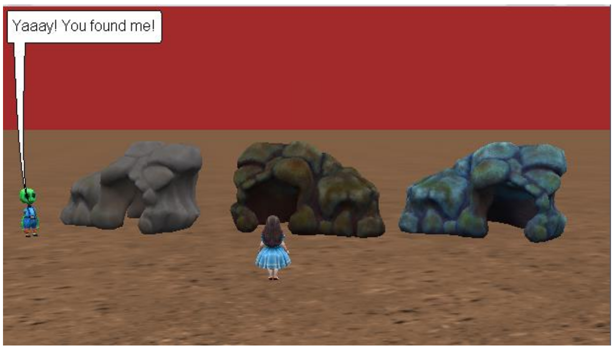


Figure 6 The *World While* scene

World Each

In “World Each”, Alice will learn what the instructions “for each in …” and “each in … together” are doing. This way she will understand their meaning. In this world, Alice will meet “Alice Each”. She will make each Quadruped resize, make a step backward and come to its original size again and step forward.

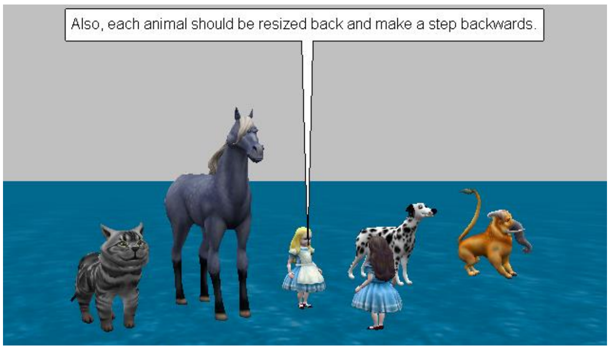

Also, each animal should be resized back and make a step backwards.

Figure 7 The *World Each* scene

V. THE IMPACT ON LEARNING OBTAINED BY DEVELOPING EDUCATIONAL GAMES

Table 1 Important feedback from students

IMPORTANT FEEDBACK FROM STUDENTS				
Question	Always	Almost always	Almost never	Never
Were the Alice platform, the new technology and the developed games helpful to learn object oriented programming basics?	61.67%	30%	8.33%	0%
Was easier and more pleasant to perform the laboratory activities, with this form of work supported by software technology for creating games? Does motivated you to perform more comfortably the activities at this lab?	45%	55%	0%	0%
Do you think that Alice and the created educational games represent a reliable learning environment?	60%	35%	5%	0%

To achieve higher levels of motivation, meeting the professional interests of students is required but, in the same time, the teacher must guide them towards new knowledge of that he is aware that must be learned. In order to evaluate the satisfaction level of students regarding our teaching approach and for quantifying the impact on learning by developing educational games, we conducted a survey (see table 1) among 60 students from our departments: Multimedia Systems

Engineering Sibiu and from “Samuel von Brukenthal” National College Sibiu. The questions were focused on student’s perception related to practical experience in laboratory where they built scenarios and have developed games. Specifically, we tried to find out if they understand the OOP principles, if they love to code and, whether the “learning by gaming” represents a more suitable teaching solution than traditional methods. According to the students feedback we may say that their perception on learning by own developed educational games is positive, representing a good alternative to the classical teaching methods.

VI. CONCLUSIONS AND FURTHER WORK

Sometimes the information presented within the lectures are not transferred completely to, and well understood by, students because of the age and mentality differences. From pedagogical viewpoint, the game-based teaching approach improves the learning process, since it permits students to notice, to be creative, rather than learn through classical methods, discovering the fact that studying can be fun as well. As a conclusion to our teaching solution of programming based on game development in Alice, we remark increasing the cognitive performance of the students and their interest and motivation to the content of the subject using this kind of laboratory activity, and also enhancing firmness, initiative, creativity especially for multimedia systems engineering students. By engaging teachers beside students to these training activities (playing, extending games or, developing new ones) helped teachers to better understand the learning process (providing students more accurate advices) because they saw which issues were easy to solve and, which were difficult and prevented students to achieve the learning goals.

As a further work we intend to create a collection of games that models and simulates relevant scientific topics from biology, chemistry, physics, etc. using visual programming. Also, importing our Alice projects (.a3p programs) into the NetBeans IDE and further software development, with emphasize on concepts such as inheritance and polymorphism, will be another issue.

References

[1] Wing J., A Vision for the 21st Century: Computational Thinking, Communications of the ACM Magazine, vol. 49, no. 3, March 2006, pp. 33-35.

[2] Sengupta P., Dickes A., Farris A.V., Karan A., Martin D., Wright M., Programming in K-12 science classrooms, Communications of the ACM Magazine, Volume 58 Issue 11, Pages 33-35, November 2015.

[3] https://academy.oracle.com/en/oa-web-overview.html

[4] http://kidsintech.adfaber.org/proiecte/programezi-cu-alice/

[5] https://code.org/curriculum/science

[6] Vintan L., Educaţia universitară în ingineria calculatoarelor: spre o abordare cultural-ştiinţifică, Revista de Politica Ştiinţei şi Scientometrie, Vol. 4, No. 3, pp. 204-208, Sept. 2015.

[7] https://www.java.com/en/about/

[8] Meerman T., Providing Support Networks for Women in STEM, ACM News, September 29, 2015.

[9] Stober D., Needed: More women in data science, ACM TechNews – Stanford Report, November 12, 2015.

[10] Chirilă, C.B., Educational resources as web game frameworks for primary and middle school students, The 10th eLearning and Software for Education Conference, Romania, 2014, Issue 1, pages: 207-214.

[11] Gliga, F., Ivanescu, A., Computer training of approximate number system could improve math abilities, The 10th eLearning and Software for Education Conference, Romania, 2014, Issue 3, pages: 218-222.

[12] Anghel, T., Florea, A., Gellert, A., Florea, D., 2014. Developing Online Collaborative Games for e-Learning Environments. In *New Horizons in Web Based Learning* (pp. 221-230). Springer Berlin Heidelberg.

[13] Florea A., Burghelea E., Gellert A., Florea D., 2015. MiniGL: Game and Learning, The 11th "eLearning and Software for Education" (eLSE) Conference, 2015, Issue 1, pages: 180-187.

[14] Dann W., Cosgrove D., Slater D., Culyba D., Cooper S., Mediated transfer: Alice 3 to Java, Proceedings of the 43rd ACM Technical Symposium on Computer Science Education, pp. 141-146, Raleigh, NC, USA, February 2012.

[15] Wang T.-C., Mei W.-H., Lin S.-L., Chiu S.-K., Lin J.M.-C., Teaching Programming Concepts to High School Students with Alice, 39th ASEE/IEEE Frontiers in Education Conference, San Antonio, TX, October 2009.

[16] Magee J., Han L., Integrating a Science Perspective into an Introductory Computer Science Course, 3rd IEEE Integrated STEM Education Conference, Princeton, NJ, March 2013.

[17] Dann W., Cooper S., Pausch R., Learning to Program with Alice (3rd Edition), Prentice Hall, 2011.

[18] Pierce J.S. et. Al, Alice: easy to learn interactive 3D graphics, CHI 98 Conference Summary on Human Factors in Computing Systems, pp.26-27, April 18-23, 1998, Los Angeles, California, USA.