

## **Rethinking and Redesigning an Image Processing Course from a Problem-Based Learning Perspective**

Reng, Lars; Triantafyllou, Evangelia; Triantafyllidis, George

*Published in:*  
Global Research Community: Collaboration and Developments

*Publication date:*  
2015

*Document Version*  
Accepted author manuscript, peer reviewed version

[Link to publication from Aalborg University](#)

*Citation for published version (APA):*  
Reng, L., Triantafyllou, E., & Triantafyllidis, G. (2015). Rethinking and Redesigning an Image Processing Course from a Problem-Based Learning Perspective. In E. de Graff, A. Guerra, A. Kolmos, & N. A. Arexolaleiba (Eds.), *Global Research Community: Collaboration and Developments: 5th International Research Symposium on Problem Based Learning* (pp. 178-186). Aalborg Universitetsforlag.  
[http://vbn.aau.dk/files/217363939/Global\\_research\\_community\\_collaboration\\_and\\_development\\_final.pdf](http://vbn.aau.dk/files/217363939/Global_research_community_collaboration_and_development_final.pdf)

### **General rights**

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal -

### **Take down policy**

If you believe that this document breaches copyright please contact us at [vbn@aub.aau.dk](mailto:vbn@aub.aau.dk) providing details, and we will remove access to the work immediately and investigate your claim.



# Rethinking and Redesigning an Image Processing Course from a Problem-Based Learning Perspective

Lars Reng<sup>1</sup>, Evangelia Triantafyllou<sup>2</sup> and Georgios A. Triantafyllidis<sup>3</sup>

<sup>1,2,3</sup> Department of Media Technology, Aalborg University Copenhagen, Denmark;  
[lre@create.aau.dk](mailto:lre@create.aau.dk), [evt@create.aau.dk](mailto:evt@create.aau.dk), [gt@create.aau.dk](mailto:gt@create.aau.dk);

## Abstract

Our experience at the Media Technology department, Aalborg University Copenhagen has shown that learning core concepts and techniques in image processing is a challenge for undergraduate students. One possible cause for this is the gap between understanding the mathematical formalism of such concepts and being able to use them for solving real-world problems. The Problem-Based Learning (PBL) pedagogy is an approach, which favours learning by applying knowledge to solve such problems. However, formulating an appropriate project for image processing courses presents challenges on how to appropriately present relevant concepts and techniques to students. This article presents our redesign of an image processing course at the Media Technology department, which focused on relevant concept and technique presentation and design projects and employed a game engine (Unity) in order to present such concepts and techniques. In a Unity environment, we developed visualizations of core concepts and basic image processing techniques. Unity was also used by students for developing projects (games) as assignments. The first offering of this new course format has been an intense learning experience for the instructional team. Media Technology students have welcomed the idea of using for image processing a tool they already use for other courses. Moreover, the visualizations and design projects in Unity have proved to increase student understanding compared to previous semesters, where other programming libraries were used. Since these preliminary results were very positive, we are planning to conduct a large scale quantitative study on the use of Unity and student understanding of image processing concepts during next year.

**Keywords:** image processing, game engine, problem-based learning, design projects, visualizations

## 1 Introduction

A thorough understanding of signal processing is paramount in many engineering courses, such as communications, sound or image processing. Various educational researchers and engineering educators have investigated ways to make such courses easy to understand. However, the abstract and complex mathematical concepts involved in signal processing and the disconnection of these concepts from real world continue to pose a challenge in conceptually understanding signal analysis (Fayyaz, Streveler, Iqbal, & Kamran, 2015). Fayyaz et. al proposed that difficulties in conceptually learning signal processing arise from insufficient understanding of the following concepts: (1) the difference between continuous and discrete domains, (2) discrete frequency, (3) units of Fourier series and Fourier transforms, (4) periodic/apperiodic or finite/infinite duration signals, (5) sampling, (6) aliasing and folding, (7) abstract mathematical concepts, and (8) advanced mathematical thinking ability. They have also identified the following possible explanations for these learning hurdles: phenomenological primitives, ontological miscategorization of

discrete and continuous domain signals, and the lack of ability among students for advanced mathematical thinking.

Our experience in the Media Technology department of Aalborg University, Denmark confirms the aforementioned findings. Our teaching experience has shown that learning core concepts and techniques in image processing is a challenge for undergraduate students. As literature has discussed and our experience has shown a cause for this is the gap between understanding the mathematical formalism of such concepts and being able to use them for solving real-world problems. In order to alleviate such learning hurdles for students, we introduced a new course design for the image processing course at Media Technology.

This article presents our redesign of the image processing course at the Media Technology department, which focused on relevant concept and technique presentation and design projects. This novel course format was offered during the autumn 2014 term at the Aalborg University Copenhagen, Denmark. During this term, we used a game engine (Unity) in order to present course concepts and techniques (Reng, 2012). In a Unity environment, we developed visualizations of core concepts (e.g. image representation in pixels) and basic image processing techniques. Our hypothesis was that by using Unity to implement various image processing techniques (e.g. point processing, neighbourhood processing, histograms, blob detection, etc), students could grasp more easily fundamental concepts (such as colour spaces, filters, object detection etc), since the students work on a familiar and popular environment which is not difficult to use it (compared to a programming library) and is also connected to game development which attracts a lot of students' attention. Unity was also used by students for developing projects (games) as assignments. We got inspiration for introducing this learning approach by the Problem-Based Learning (PBL) pedagogy, which is applied to all programs at Aalborg University. In the following, we mention the core characteristics of PBL and then present the new course design. We conclude this article by discussing our experiences with this new learning approach and proposing actions for future work.

## **2 Related research**

Our search in the SCOPUS database revealed no articles describing the introduction of a game environment or engine to image or signal processing undergraduate courses. By using the keywords "game AND image processing course" and "Unity AND image processing course", we got 38 and 5 results respectively, but none of them was relevant. We have also used the keywords "Unity AND image processing", which returned 308 results but they did not refer to the educational field, while the keywords "game engine AND image processing" revealed 94 results but they were from the game architecture field. By omitting the image processing part, we have found a few approaches where game environments have been employed for software engineering or computer science courses and some other related approaches, which have applied other PBL-inspired learning methods in signal processing. In the following, we present research from those fields.

Albu and Malakuti investigated a hybrid instructional format that combined traditional lectures with a PBL-based component (Albu & Malakuti, 2009). Their approach aimed at providing a framework for solving multimedia-related digital signal processing problems, where students were encouraged to formulate their own problems. The PBL component was added to the lecture-centric course and custom-designed software was introduced for system design and analysis. The authors observed student-formulated problems in approximately 15% of the total number of handed-in projects. Interesting applications included image segmentation via edge detection, creating audio effects via digital filtering, and segmentation of piano

sounds based on harmony analysis. However, some students expressed the desire to receive a more structured project assignment formulated in closed-form and the authors mentioned improvements that have to be made to the software they introduced in terms of stability, flexibility, and efficiency for the implementation of user-formulated filters.

Arévalo et al. describes introduced a PBL approach in a computer vision course (Arévalo et al., 2011). This approach focused on the practical block of the subject and consisted of collaboratively developing a software application to solve a central computer vision problem: detecting and classifying objects in images. The aim of their initiative was twofold: getting the student to assimilate and put the acquired knowledge into practice (specific skills); and to develop generic skills such as planning and conducting their learning, performing individual and group works, coordination, etc. Arévalo et al. evaluated this collaborative PBL approach according to the degree of students' satisfaction and the students' academic outcomes. They evaluated the produced material (final reports, forum contributions, etc.) and students' self-assessments and the results indicated a high degree of satisfaction and involvement of students, better academic outcomes (compared to previous years) and solutions to the problem, in some cases, really creative.

In the field of employing game environments to promote student participation, enable variation in how lectures are taught, and improve student interest in higher education, Wang and Wu used a game development framework as a learning aid in a software engineering course (Wang & Wu, 2009). In their article, they described a case study where a game development framework (XNA) was applied to make students learn software architecture by developing a computer game. They provided a model for how game development frameworks can be integrated with a software engineering or computer science course and they described important requirements to consider when choosing a game development framework for a course. In their approach, they made some extensions to the existing game development framework to let the students focus more on software architectural issues than the technical implementation issues. The responses from the students were overall very positive compared also to previous years. Students felt they learned a lot from the game project and they liked the practical approach of the project. However, there were some students who felt that there was lack of XNA and technical support during the project and that there was too much focus on XNA, and games and too little on software architecture.

In our approach, we got inspired by the working with real-world artifacts approach of PBL and we combined it with the introduction of a game environment (Unity). In this sense, our approach is novel, especially because it is applied in an image processing course. In the following section, we describe the main characteristics of PBL, which build the theoretical framework of our approach.

### **3 Problem-Based Learning**

PBL is a student-centered pedagogy in which students learn through the experience of problem solving (Kolmos, Krogh, & Fink, 2004). Learning begins with a problem to be solved, posed in such a way that students need to gain new knowledge before they can solve the problem, and thereby learning both thinking strategies and domain knowledge. The goals of PBL are to help the students develop flexible knowledge, effective problem solving skills, self-directed learning, effective collaboration skills and intrinsic motivation (Hmelo-Silver, 2004).

PBL may also support group work (Kolmos, 1996). Working in groups, students identify what they already know, what they need to know, and how and where to access new information that may lead to resolution of the problem. This procedure enhances content knowledge while simultaneously fosters the

development of communication, problem-solving, critical thinking, collaboration, and self-directed learning skills. PBL may position students in a simulated real world working and professional context, which involves policy, process, and ethical problems that will need to be understood and resolved to some outcome. By working through a combination of learning strategies to discover the nature of a problem, understanding the constraints and options to its resolution, defining the input variables, and understanding the viewpoints involved, students learn to negotiate the complex sociological nature of the problem and how competing resolutions may inform decision-making.

Additionally, PBL represents a paradigm shift from traditional classroom/lecture teaching. The role of the instructor in PBL (known as the tutor) is to facilitate learning by supporting, guiding, and monitoring the learning process. The tutor must build students' confidence to take on the problem, and encourage the students, while also stretching their understanding. Therefore, the role of the teacher is to guide and challenge the learning process rather than strictly provide knowledge.

PBL was first introduced in the medical school program at McMaster University in Hamilton, Ontario, Canada in the late 1960s (Neville, 2008). Since then various universities and other educational institutes have adopted PBL as a model of teaching and learning. Since its establishment in 1974, Aalborg University, Denmark bases all its university programs on PBL, also referred to as "PBL - The Aalborg model" (Barge, 2010). The PBL - Aalborg Model has become both nationally and internationally recognized and a trademark for Aalborg University. Through research in PBL, Aalborg University continues to develop and adapt the PBL-model as a learning model for students as well as teaching staff, and ensures that the model responds well to the demands and changes posed by the surrounding society and changes in the education area.

The PBL – Aalborg Model shapes the institution's program curricula. The program curriculum at Medialogy is mapped onto academic terms (semesters) according to an appropriate progression with regard to depth and breadth of content as well as sophistication of project work. Each program consists of an appropriate balance of courses, which accompany the students' project work. In each semester, a theme is selected to serve as the context, in which central theme related courses and semester projects address the learning objectives. Within the theme and the overall learning objectives, problems and project proposals are to be chosen. Apart from their semester projects, students have often to work on projects for their semester courses.

#### **4 The new course design**

The most common approaches for running Image Processing (IP) courses is by using a specific IP library (e.g. Matlab Image Processing Tollbox, OpenCV, OpenFrameworks, etc). Such tools are very powerful for conducting research and developing IP applications, but our observations during IP courses and student performance on IP have shown that Media Technology students find it difficult to understand and use the IP concepts. Moreover, student evaluation of the IP course has shown that a pure programming platform (e.g. Matlab, OpenCV, etc) is not very attractive for such students. Therefore, we decided to redesign the Introduction in IP course for Media Technology students.

Taking into account the aforementioned observations, we identified two challenges. On one hand, how to better communicate the IP concepts, and on the other hand how to motivate students to start experimenting with these concepts. We decided to resolve these challenges by employing the PBL context of our university and our background and previous experience in game engineering. Therefore, we

introduced the use of a game engine (Unity) as a means for efficient visualization of the IP concepts, and at the same time as a platform for game, development where the IP concepts are used.

The syllabus of the Introduction in IP course contains the following topics:

- Introduction to IP / imaging
- Image acquisition
- Color spaces
- Point processing / pixel operations
- Neighborhood operations / Filtering
- Morphological operations
- Color detection/tracking
- Blob analysis
- Segmentation in video and Geometric transformations
- Frequency analysis of images

In the new course design, we used Unity based visualizations for introducing abstract mathematical concepts used in IP. Moreover, students were given the possibility for their design projects to create games within Unity, where they can use IP concepts.

In the following, we present the set of basic illustrations and exercises that we used to support this new approach of teaching IP and their connection with the PBL pedagogy.

#### *Grayscale pixel representation*

In this visualization, a Unity scene is created, where the pixels are represented as 3D bars, and their grayscale value (0-255) is used to define the height of the bars in the z-axis. By creating this scene, the students can use the top point of view where the scene is shown as a 2D grayscale image (Figure 1a). Then by simply rotating it, the students can see it from a different point of view (Figure 1b and 1c), where they can see the height of the bars, which correspond to their grayscale values.

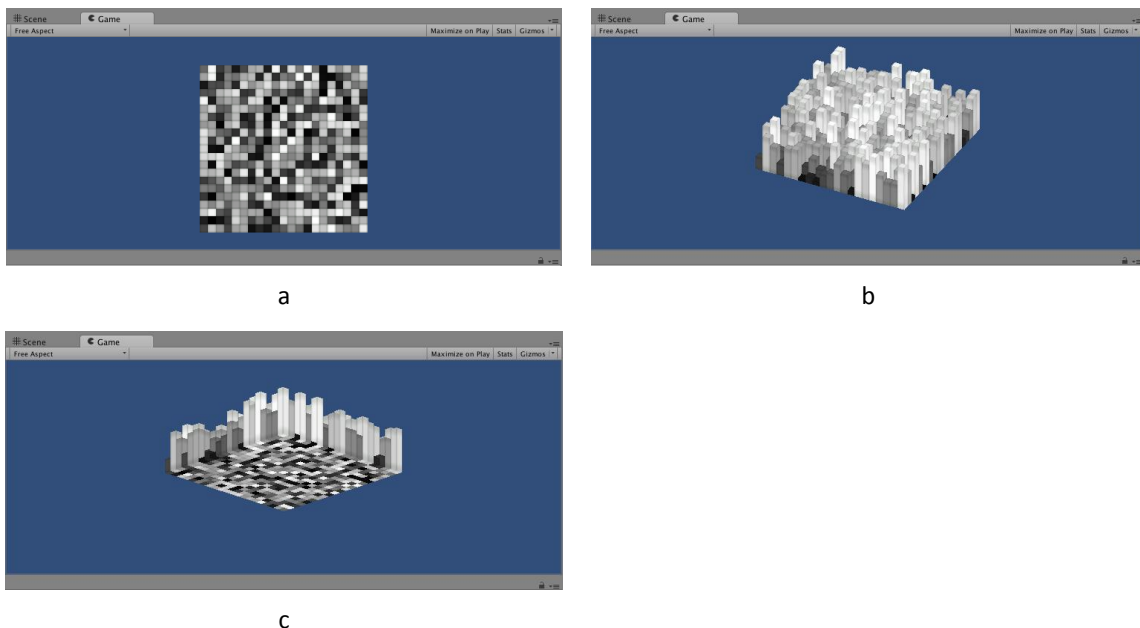


Figure 1: Grayscale pixel representation

In this exercise, we aimed at creating a connection between the grayscale values of the pixels in an image with the histogram of the image. The bars representing the values relate to the bars in a histogram and this connection creates a real-world representation of an abstract concept as the grayscale value of a pixel.

#### *Color representation*

This is an exercise, where the students were asked to create a cube in Unity and try to paint it in different colors by using different values for the variables red, green and blue (Figure 2).

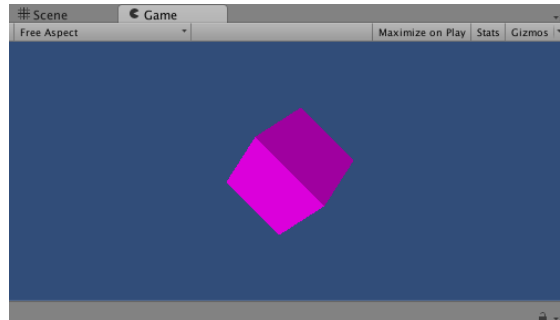


Figure 2: Coloring a cube

For this exercise, students are required to use the RGB (red, green, blue) values in order to change the color of the cube and by doing so, they are experimenting with this color space. *Point processing*

This is an exercise where the students were asked to attach a texture to a Unity game object (e.g. cube) and then do some point processing operation on this texture, like change contrast or brightness, produce inverted images, etc (Figure 3).

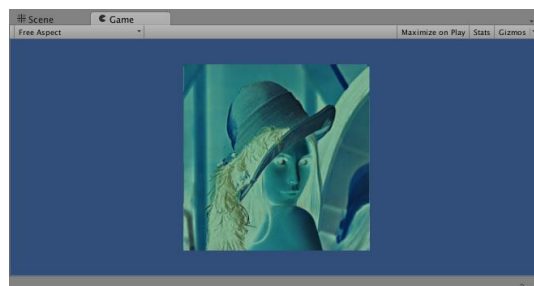


Figure 3: Inverting an image

#### *Point processing: Thresholding*

Like in the previous exercise, the students were asked to do some thresholding to a 2D texture (Figure 4).

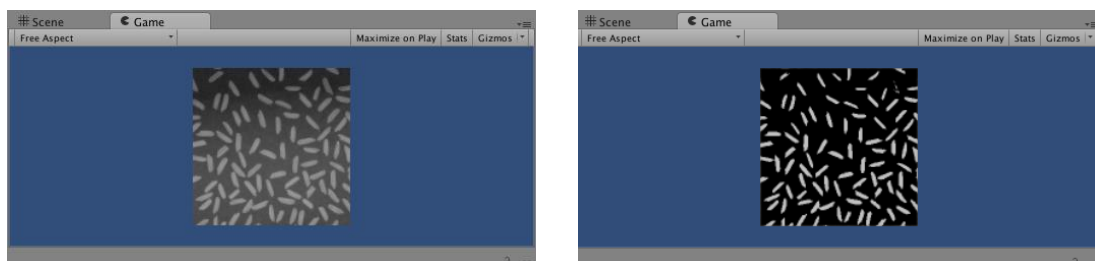


Figure 4: Thresholding an image

#### *Neighborhood processing*

In such exercises, the students were asked to apply neighborhood processing operators such as mean filters, rank filters or edge detectors (Figure 5).



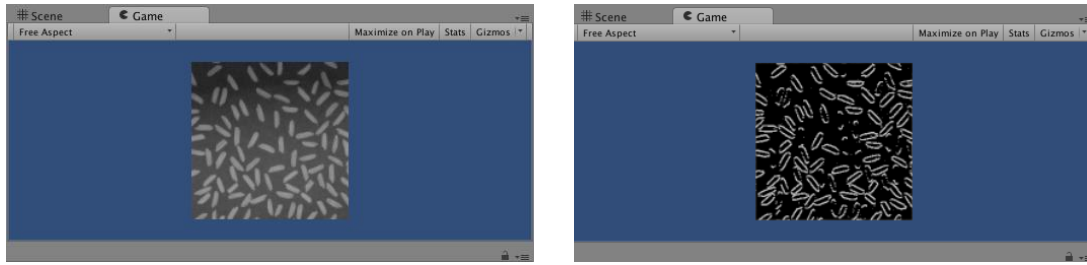


Figure 5: Edge detection (Prewitt kernel)

### *Morphological operators*

In these exercises, the students were asked to apply erosion, dilation, closing and opening in a 2D texture (Figure 6).

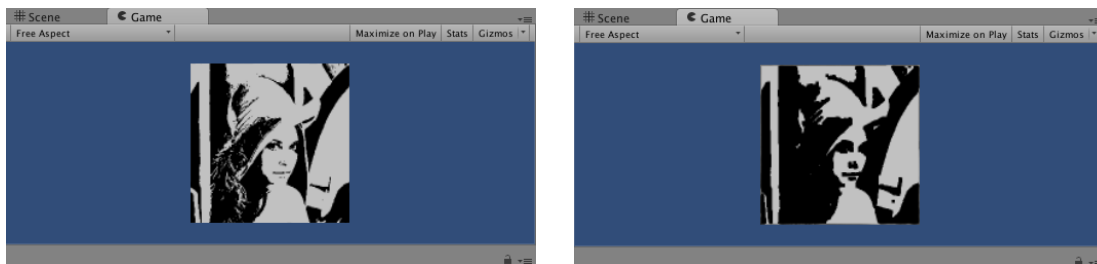


Figure 6: 5x5 erosion

In all the aforementioned exercises, students engaged with game objects that are similar to real world ones and they used a tool, which they have used in other fields of their study. The introduction of real world problems and the interconnection between different courses are two of the main characteristics of PBL.

In combination with these basic exercises, we used group work, which is also aligned with the Aalborg PBL model. We asked the students to work in groups in order to implement in Unity the following small (and more advanced) games, which use IP concepts.

*Project 1:* Illustrate the grassfire algorithm for BLOB detection by showing all the steps of the blob detection pixel by pixel for the sequential and recursive methods. This illustration is using an island (as an image) and some grass regions (as the BLOBS) that need to be burnt according to the grassfire algorithm.

*Project 2:* Create a game quiz where different point processing operations are selected (e.g. thresholding, inversion, gray level mapping, brightness, contrast) and the game player should find out which one has been applied to the texture

*Project 3:* Create a game where the player should use a specific series of operations (point and neighborhood) in order to find out a hidden message within the texture.

*Project 4:* Create an application where different edge detectors are applied to illustrate the different results.

## **5 Discussion**

The introduction of Unity for visualizing abstract IP concepts has been proved to be beneficial for student understanding. The results (pass/fail) on student performance showed that there was a 12% improvement compared to the last year (59% passed last year, while 71% passed this year). Moreover, students showed better understanding during class discussions and in their design projects. We have also gathered student feedback during lectures, and it was very positive. Students reported that using a tool they already knew

allowed them to focus on the comprehension of the new concepts. Furthermore, they mentioned that applying IP in a field they are aware with (game development) increased their motivation to work and their sense of meaningful learning.

Regarding design projects in Unity, the student feedback was also very positive. Students showed enthusiasm on the idea of creating games for an IP course and felt that it contributed to their learning. We believe that these design projects contributed also to introduce a PBL aspect to our course. This approach of letting students build real life objects and artefacts and interact with them is in line with the core aspects of PBL. We have experienced that students were able to submit high quality projects and approached the problem solution in various ways. However, we noticed that they needed much more assistance while conducting group work. This assistance was provided by the teaching assistants assigned to this IP course.

## 6 Conclusion

The first offering of this new course format has been an intense learning experience for the instructional team. Media Technology students have welcomed the idea of using for IP a tool they already use for other courses. Moreover, the visualizations and design projects in Unity have proved to increase student understanding compared to previous semesters, where other programming libraries were used. Since these preliminary results were very positive, we are planning to conduct a large scale quantitative study on the use of Unity and student understanding of IP concepts during next year.

## References

- Albu, A. B., & Malakuti, K. (2009). Work in progress-problem-based learning in digital signal processing. *Frontiers in Education Conference, 2009. FIE'09. 39th IEEE*, 1-2.
- Arévalo, V., Luque-Baena, R., Gómez, G., Nieto, J. G., Sánchez-Sánchez, J., & Velasco-Álvarez, F. (2011). Collaborative Problem-Based Learning. Experience in the "Computer Vision" course (Computer Engineering Degree). *ICERI2011 Proceedings*, , 6614-6619.
- Barge, S. (2010). *Principles of problem and project based learning, the Aalborg PBL model*. Aalborg: Aalborg University. [http://www.aau.dk/digitalAssets/62/62747\\_pbl\\_aalborg\\_modellen.pdf](http://www.aau.dk/digitalAssets/62/62747_pbl_aalborg_modellen.pdf)
- Fayyaz, F., Streveler, R. A., Iqbal, A., & Kamran, M. (2015). Category mistakes, knowledge in pieces, or something else? problems in conceptually learning signal analysis. *International Journal of Engineering Education*, 31(1), 58-71.
- Hmelo-Silver, C. E. (2004). Problem-based learning: What and how do students learn? *Educational Psychology Review*, 16(3), 235-266.
- Kolmos, A. (1996). Reflections on project work and problem-based learning. *European Journal of Engineering Education*, 21(2), 141-148.
- Kolmos, A., Krogh, L., & Fink, F. K. (2004). *The aalborg PBL model: Progress, diversity and challenges* Aalborg University Press Aalborg.
- Neville, A. J. (2008). Problem-based learning and medical education forty years on. *Medical Principles and Practice*, 18(1), 1-9.

Reng, L. (2012). Development of an artificial intelligence programming course and Unity3d based framework to motivate learning in artistic minded students. *International Conference on Engineering Education 2012*, 824-830.

Wang, A. I., & Wu, B. (2009). An application of a game development framework in higher education. *International Journal of Computer Games Technology*, 2009, 6.