A Problem-Based Learning Approach of Teaching Mathematics to Media Technology Students Using a Game Engine

Triantafyllou, Evangelia; Misfeldt, Morten; Timcenko, Olga

Published in:
Global Research Community: Collaboration and Developments

Publication date:
2015

Document Version
Accepted author manuscript, peer reviewed version

Citation for published version (APA):
Triantafyllou, E., Misfeldt, M., & Timcenko, O. (2015). A Problem-Based Learning Approach of Teaching Mathematics to Media Technology Students Using a Game Engine. In E. de Graff, A. Guerra, A. Kolmos, & N. Arexolaleiba (Eds.), *Global Research Community: Collaboration and Developments: 5th International Research Symposium on Problem Based Learning* (pp. 155-166). Aalborg Universitetsforlag. http://vbn.aau.dk/files/217363939/Global_research_community_collaboration_and_development_final.pdf

A Problem-Based Learning Approach of Teaching Mathematics to Media Technology Students Using a Game Engine

Evangelia Triantafyllou[1], Morten Misfeldt[2], Olga Timcenko[3]

[1,3] Department of Media Technology, Aalborg University Copenhagen, Denmark;
[2] Research Lab: ICT and Design for Learning, Aalborg University Copenhagen, Denmark;
evt@create.aau.dk, misfeldt@learning.aau.dk, ot@create.aau.dk;

Abstract

In this article, we present our idea of using a game engine (Unity) to teach Media Technology students mathematics-related concepts. In order to observe how the introduction of a technological tool, namely the game engine, changes the practices in mathematical work, we adopted the anthropological approach in didactics. This theoretical framework defines the “atoms” of mathematical practice and discourse in terms of tasks, techniques, technologies and theories. We present a didactical scenario when Unity is used for introducing the calculation of reflection and refraction vectors and then we use the anthropological approach to present the practices in calculating these vectors with traditional mathematics and constructing them in Unity. Then, we discuss the differences between the two cases, when we argue that Unity can benefit Media Technology students, who use mathematics as a tool. However, the assumptions on the mathematical practice while using Unity will have to be confirmed in actual educational settings.

Keywords: mathematics education, game engine, problem-based learning, the anthropological approach, media technology

1 Introduction

Over the past years, engineering education has been challenged to embed creativity and innovation, in order to produce graduates who can easily adapt to societal changes (Badran, 2007; Jørgensen & Busk Kofoed, 2007; Zhou, 2012). As a result, a number of engineering programs have arisen that transcend the division between technical, scientific and creative disciplines. The teaching of mathematics to students of such disciplines represents a challenge to the education system because these disciplines are typically constructed in specific opposition to technology and science. This paper emerges from our research that explores the teaching of mathematics in such an engineering discipline, namely the Media Technology program at Aalborg University (Triantafyllou & Timcenko, 2013).

Regarding mathematical education in traditional engineering studies, it has been found that engineering students often have difficulties with understanding the mathematical concepts due to their lack of fundamental understanding of difficult concepts or due to their inability to perform deductive reasoning (Morgan, 1990). Moreover, it has been found that the conceptions of mathematical concepts are different for engineering students from those of mathematics students (Maull & Berry, 2000), and that engineering students see mathematics as a tool, and therefore wish to see the application side as part of the course (Bingolbali, Monaghan, & Roper, 2007). In our own research, we have confirmed the aforementioned findings for Media Technology students and we have also found that these students are reluctant to use technology in mathematics, since they believe that it adds to the complexity of such courses (Triantafyllou & Timcenko, 2014).

Inspired by the constructivist aspects of Problem-Based Learning, which Aalborg University applies to all its programs (Barge, 2010), we came up with the idea of substituting traditional mathematics assignments with mini-projects in a game engine (Unity). The main concept of this approach is that students get simple projects in Unity, where mathematics is used for game mechanics and they have to modify or further develop these projects. With this approach, we aimed at changing the mathematical practice of these students, by relating it with tangible objects in a virtual world. We chose Unity, since Media Technology students are familiar with it, and we wanted to avoid the learning effort of employing a new tool. In the following, we present a theoretical framework for analysing and comparing the practices when traditional mathematics and Unity is employed. Then, we present an example of a didactical scenario, where Unity is used for introducing light reflection and refraction to Media Technology students. We conclude that Unity offers new possibilities for such students, who wish to see the application part of mathematics.

2 Background

Using a game engine for mathematical assignments involves students programming for solving these assignments. The idea that programming could be used to develop or enforce mathematical ideas is not new. Based on constructivism, Seymour Papert developed the programming language LOGO, where children can guide a small turtle around the screen. The turtle leaves a trace while moving around, allowing the child to create various geometrical figures (Papert, 1980). His suggestion was that children learn in a particularly efficient way when they are engaged in developing constructs such as beautiful patterns, interactive art, etc. Papert described LOGO as a "mathematical microworld" that allows children to engage in such projects. During the 1980s, there was great enthusiasm and confidence that LOGO and similar programming languages would radically reform mathematics teaching in primary schools. However, the results in mainstream implementation did not entirely live up to the expectations. There are a number of reasons for the disappointing results; for instance, students easily overlook the nuggets of mathematical knowledge, making their work in the microworld non-mathematical (Ainley, Pratt, & Hansen, 2006; Hoyles & Noss, 1992).

The idea that programming could be helpful in mathematics education in the late 1980s was also developed in the context of teaching mathematics at high school and college. Here the geometric and artistically framed LOGO program was less popular. On the contrary, teachers often utilized common programming languages such as BASIC, COMAL and PASCAL to support learning. One of the outspoken hopes was to create a process-oriented approach to abstract mathematics, basing abstract constructions in concrete numerical computations. Ed Dubinsky's work is probably the clearest description of the learning potential of programming (Breidenbach, Dubinsky, Hawks, & Nichols, 1992). His theoretical framework describes mathematical concept formation as beginning with performing actions on well-understood mathematical objects; these actions can be organized in processes and encapsulated into objects. These objects can be related to one another in schemas. This theoretical framework of mathematical concept formation was applied to improve the development of the process conception of function for university students and uses computers for empowering and enriching the concrete numerical calculations that are the necessary foundation for concept formation.

During the last years, digital games have been applied in many educational fields to enhance learning motivation (Prensky, 2001). Since game environments or engines allow users to customize their gaming experiences by building and expanding game behaviour, games offer new directions in relation to learning mathematics by programming, which have not been extensively explored. El-Nasr and Smith have proposed the use of modifying, or modding, existing games as a means to learn computer science, mathematics, physics, and aesthetic principles (El-Nasr & Smith, 2006). In two exploratory case studies, they presented skills learned by students as a result of modding existing games and they discussed the benefits of learning computer sciences skills, among others 3D graphics and mathematics. However, the literature has yet to discuss if and how programming in games can contribute to meaningful mathematics learning.

3 Theoretical Framework

The tools that we choose to bring to mathematics students do influence the learning of mathematics that becomes likely or possible (Ainley et al., 2006; Guin, Ruthven, & Trouche, 2006). And in that sense bringing programming into mathematics teaching does support certain types of learning. In order to observe these types of learning, we adopt the theoretical frameworks adopted and developed by Artigue (Artigue, 2002), namely the anthropological approach in didactics initiated by Chevallard (Chevallard, 1990), and the theory of instrumentation developed in cognitive ergonomics (Verillon & Rabardel, 1995).

3.1 The anthropological approach in didactics

The anthropological approach in didactics provides tools to model mathematical and didactical knowledge (Winsløw, 2012). This didactical theory views mathematics as the product of a human activity. Therefore, mathematical productions are framed by the social and cultural contexts where they develop and mathematical objects are entities which arise from the practices of given institutions (Artigue, 2002). These practices, also called "praxeologies", as described by Artigue, have four components: "…a type of task in which the object is embedded; the techniques used to solve this type of task; the "technology", that is to say the discourse which is used in order to both explain and justify these techniques; and the "theory" which provides a structural basis for the technological discourse itself and can be seen as a technology of the technology." (Artigue, 2002) Winsløw mentions that the tasks and the techniques define each other and calls the couple of a task and a technique as a practical block – the minimal entity of practical knowledge. Technologies explain how to apply and distinguish a whole set of techniques. At a higher level of discourse, technologies are developed, explained, related and justified in and by a theory (Winsløw, 2012). For a given set of practical blocks, we can define the theoretical block, which is formed by a technology and a theory. The anthropological approach describes the mathematical activity using the practical and the theoretical discourse. However, when a technique becomes routine in an institution, it tends to lose its connection to the theoretical discourse and becomes a simple, "de-mathematicised" act. Therefore, this approach helps to observe the changes that happen when technological tools are inserted into mathematical learning, since it offers a framework to observe if and how the practical and the conceptual work are interrelated.

3.2 The instrumental approach

The instrumental approach addresses students' use of technology when learning mathematics from the perspective of appropriating digital tools for solving mathematical tasks (Guin et al., 2006). It views computational artifacts as mediating between user and goal (Rabardel & Bourmaud, 2003). It is an important aspect of this conceptualization that humans have goals on various levels, and hence that the goal of smaller actions can feed into larger plans (Nardi, 1996). Furthermore the approach presupposes a continuation and dialectic between design and use, in the sense that a pupil's goal-directed activity is shaped by his use of a tool (this process is often referred to as instrumentation), and simultaneously the goal-directed activity of the pupil reshapes the tool (this process is often referred to as instrumentalization). In students' work with technology the distinction between epistemic mediations and pragmatic mediations operationalize the difference between learning with technology and just using technology to solve tasks (Guin et al., 2006; Rabardel & Bourmaud, 2003). Epistemic mediations relate to goals internal to the user—affecting his or her conception of, overview of, or knowledge about something and pragmatic mediations related to goals outside of the user—making a change in the world. Finally, Rabardel and Bourmaud introduce sensitivity to a broader conception of the orientation of the mediation. Instrumented mediations can be directed towards (a combination of) the objects of an activity (the solution of a task), other subjects (classmates, the teacher), and oneself (as a reflective or heuristic process). Hence the theoretical framework consists of the concepts: instrumental genesis, as consisting of instrumentation and instrumentalization, the concepts epistemic and pragmatic mediations, as well as sensitivity towards the orientation of an instrumented mediation. The orientation of the mediation can be towards oneself, external objects, and other subjects.

4 The didactical scenario - reflection and refraction vectors

In this article, we use a didactical scenario and we examine how praxeologies change from traditional instruction to instruction with the use of Unity. This didactical scenario concerns light reflection and refraction and is taken from the computer graphics rendering course of the fifth semester at the Media Technology bachelor program. We selected the specific scenario because it involves mathematical work that can be visually represented. In this context, Unity (or any game engine) can greatly contribute to user understanding since it offers visualisation and interaction possibilities.

The didactical scenario took place during one lecture. The lecture started with the mathematical reflection vector calculation (Figure 1). The teacher (one of the authors of this article) reminded the students of the fact that according to the law of reflection, the angle of incidence equals the angle of reflection, and she explained the calculation of the projection of one vector to another. Then, students were asked to think how to calculate the reflection vector based on this information. On the whiteboard, the teacher then solved together with student contributions an exercise on calculating the reflection vector, given the coordinates of the vector of the incoming light and the angle of incidence.

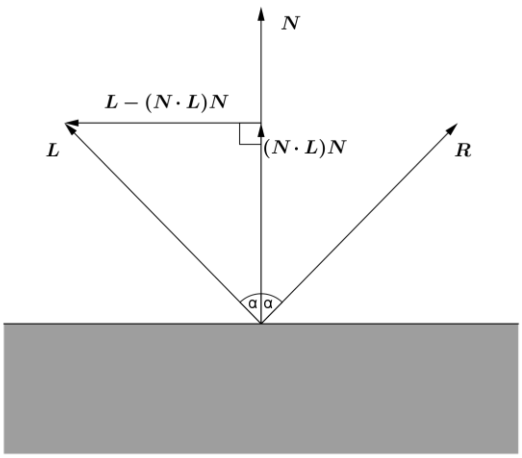

Figure 1: The direction of reflection **R** forms the same angle with the normal vector **N** as the direction **L** pointing toward the incoming light. It is found by subtracting twice the component of **L** that is perpendicular to **N** from **L** itself. (Lengyel, 2012)

Thereafter, the teacher calculation of refraction vector was explained by first introducing Snell's law, which describes the relationship between the angles of incidence and refraction, when referring to light or other waves passing through a boundary between two different isotropic media. Then, she went through the mathematical calculation of the refraction vector (Figure 2), and she discussed the conditions that invalidate the refraction formula and the physical phenomenon observed under these conditions (total internal reflection).

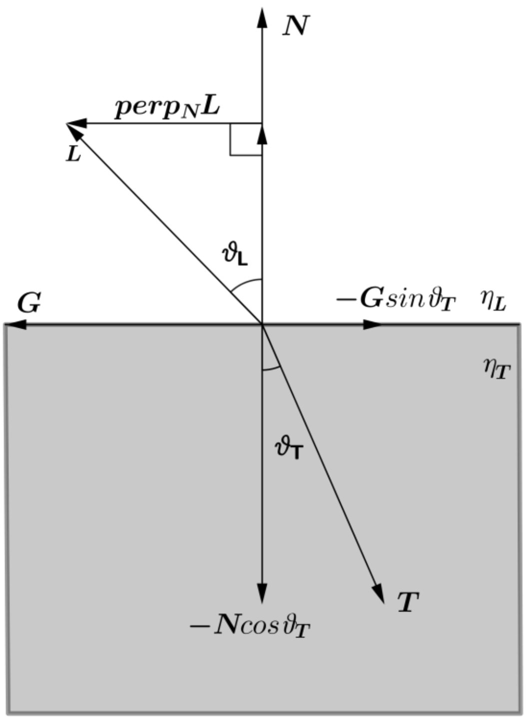

Figure 2: The angle of incidence **L** and the angle of transmission **T** are related by Snell's law. The refraction vector **T** is expressed in terms of its components parallel and perpendicular to the normal vector **N**. (Lengyel, 2012)

After these explanations, students were introduced to their homework, which it was given as a class activity (see Appendix). Students worked in their homework in class, but they could finish it and submit it up to ten days after the lecture. The homework involved using Unity for defining the reflection and refraction vectors, and gave the students the opportunity the formulas, which they were presented in class for changing game mechanics.

5 Calculation of reflection and refraction vectors using traditional mathematics and mathematics in Unity

In this section, we present the calculation of the reflection and the refraction vectors using traditional mathematics (as presented in students textbook) and using Unity. In order to understand the differences in mathematical practice in these two cases, we use the anthropological approach in didactics and describe the calculations in terms of theory, technologies, techniques and tasks involved.

5.1 Calculation of reflection vector

The task at hand here is to calculate the reflection vector **R**, given the direction **L** pointing toward the incoming light (Figure 1). This task has two steps (Table 1): the first is to use the dot product definition in order to calculate the projection of **L** to the normal direction **N** and then express the component of **L** that is perpendicular to the normal direction as the subtraction of two vectors, and the second is to express vector **R** as the subtraction between two other vectors.

For performing the aforementioned task, the techniques of the geometric definition of calculating the dot product and the definition of addition and subtraction of vectors in two dimensions are required. The task and techniques together form the practical block. The theoretical block consists of the technologies and the related theory. The technologies involved are the definition of the length of vectors, the definition of the dot product, the generic definition of addition and subtraction of vectors and the definition of unit vectors. The related theories are the law of reflection and vector spaces.

Table 1: Mathematical calculation of reflection vector

1) We first calculate the component of L that is perpendicular to the normal direction, as the subtraction of vector L and its projection on N (we use capital bold letters for the vectors): $$perp_N\boldsymbol{L} = \boldsymbol{L} - (\boldsymbol{N} \cdot \boldsymbol{L})\boldsymbol{N}$$ 2) The vector R lies at twice the distance from L as does its projection on the normal vector N. We can then express the vector R as: $$\boldsymbol{R} = 2(\boldsymbol{N} \cdot \boldsymbol{L})\boldsymbol{N} - \boldsymbol{L}$$

5.2 Construction of the reflection vector in Unity

When the reflection vector is constructed in Unity, the task is to graphically draw this vector by finding its direction from the reflection formula (Figure 3). In order to perform this task, the following techniques are required: normalization of vectors in Unity, calculation of the dot product of two 3D vectors in Unity, definition of vectors by two points (both in geometry and in Unity), and definition of rays by a point and a direction vector (both in geometry and in Unity).

The theoretical block in this case is very similar to the theoretical block of the mathematical calculation with the only addition of the definition of rays in the set of related technologies.

5.3 Calculation of refraction vector

The task is to calculate the direction of the refraction vector **T**, given the direction **L** pointing toward the incoming light (Figure 2). The angle of incidence **L** and the angle of transmission **T** are related by Snell's law. The task has five steps (Table 2).

The first step is to express the vector **G** in terms of the perpendicular part of vector **L** while the second is to express the refraction vector **T** in terms of its components parallel and perpendicular to the normal vector **N**. The following three steps aim at eliminating the trigonometric numbers in the refraction formula and replacing them with the refraction indices. The techniques to perform these steps include the geometric calculation of the dot product of vectors in 3D space, the addition and subtraction of vectors in 3D space, the decomposition of vectors in parallel and perpendicular components, the normalization of a vector, and using the fundamental trigonometric identity ($sin^2x + cos^2x = 1$) to replace $cosx$ with $sinx$ and vice versa.

The theoretical block contains the relevant technologies and theory. The technologies contain the definition of length of vectors, the definition of the dot product, the definition of addition and subtraction of vectors, the definition of unit vectors, the definition of sine and cosine, the decomposition of vectors in components, and the fundamental trigonometric identity. The theory, which is needed in order to perform the specific task, is the Snell's law, the Pythagoras' theorem, trigonometry and theory of vector spaces.

Table 2: Mathematical calculation of reflection vector

1) We express the vector **G** in terms of the perpendicular part of **L** on **N**: $$\boldsymbol{G} = \frac{perp_N\boldsymbol{L}}{sin\theta_L} = \frac{\boldsymbol{L} - (\boldsymbol{N}\cdot\boldsymbol{L})\boldsymbol{N}}{sin\theta_L}$$
2) We then express the refraction vector **T** in terms of its components parallel and perpendicular to the normal vector: $$\boldsymbol{T} = -\boldsymbol{N}cos\theta_T - \boldsymbol{G}sin\theta_T = -\boldsymbol{N}cos\theta_T - \frac{sin\theta_T}{sin\theta_L}[\boldsymbol{L} - (\boldsymbol{N}\cdot\boldsymbol{L})\boldsymbol{N}]$$
3) Using Snell's law, we can replace the quotient of sines with the quotient of refraction indices: $$\boldsymbol{T} = -\boldsymbol{N}cos\theta_T - \frac{\eta_L}{\eta_T}[\boldsymbol{L} - (\boldsymbol{N}\cdot\boldsymbol{L})\boldsymbol{N}]$$
4) Using the fundamental identity and Snell's law: $$\boldsymbol{T} = -\boldsymbol{N}\sqrt{1 - \frac{{\eta_L}^2}{{\eta_T}^2}sin^2\theta_L} - \frac{\eta_L}{\eta_T}[\boldsymbol{L} - (\boldsymbol{N}\cdot\boldsymbol{L})\boldsymbol{N}]$$
5) Replacing $sin^2\theta_L$ with $1 - cos^2\theta_L = 1 - (\boldsymbol{N}\cdot\boldsymbol{L})^2$ finally yields: $$\boldsymbol{T} = \left(\frac{\eta_L}{\eta_T}\boldsymbol{N}\cdot\boldsymbol{L} - \sqrt{1 - \frac{{\eta_L}^2}{{\eta_T}^2}[1 - (\boldsymbol{N}\cdot\boldsymbol{L})^2]}\right)\boldsymbol{N} - \frac{\eta_L}{\eta_T}\boldsymbol{L}$$

5.4 Construction of the refraction vector in Unity

When the refraction vector is constructed in Unity, the task is to graphically draw this vector by finding its direction from the refraction formula (Figure 4). In order to perform this task, the following techniques are required: normalization of vectors in Unity, calculation of the dot product of two 3D vectors in Unity, definition of vectors by two points (both in geometry and in Unity), definition of rays by a point and a direction vector (both in geometry and in Unity) and calculation of the square root of a number.

The theoretical block in this case is very similar to the theoretical block of the mathematical calculation of refraction with the only addition of the definition of rays and the definition of the square root of a number in the set of related technologies.

6 Discussion

In the previous section, we have analysed the tasks, techniques, technologies and theories involved when calculating the reflection and the refraction vector using traditional mathematics and the Unity game engine. From this analysis, we can see that although the theoretical block remains almost the same, the practical knowledge needed for performing the same task using mathematics and Unity differs. Leaving aside the technical implementation aspects, the students have to consider how a mathematical formula is connected with game objects. For example, the direction of the vector **L** in the mathematical model is pointing toward the incoming light, while the ray in Unity representing the incoming light is drawn using a vector with the opposite direction. Therefore, the students have first to create a vector with the opposite direction as L, in order to be able to correctly apply the reflection and refraction formulas in Unity. Moreover, the students can see what these formulas define in the real world (i.e. direction of the reflected/refracted light).

Another aspect that comes up when working in Unity is the specificity of the refraction formula. This formula contains the square root of an expression. When working with this formula with pen and paper, students can easily ignore the fact that the expression under the square root cannot be negative. When this happens in Unity, there is an exception called, and the students can at least see that something is wrong in their program. Realizing that this formula cannot be applied when this happens is important for understanding, since this the condition for total internal reflection to happen. Students can also experiment with different values of the incident angle (since they have the opportunity to manually change the direction of the laser pen in Unity), in order to see for which angles the light is reflected instead of refracted.

Regarding the mathematical concepts involved, the calculation of the reflection and refraction vectors requires a higher level of mathematical discourse. However, we argue that this higher level of mathematical thinking is not necessarily important for Media Technology students. The construction of these vectors in Unity still calls for understanding of the basic related mathematical concepts (e.g. direction of involved vectors, angles, unity vector on a surface), leaving aside the procedure of forming the reflection and refraction formulas. Finally, Unity offers possibilities of interaction with and visualization of game objects, and verification of self-assumptions. Such possibilities can help exploring different conditions and how they affect the game objects and the observed physical phenomena.

Finally, the use of a game engine can support mathematical work where students generate actual objects (in a virtual world), interact with them, change their properties, and observe how different objects interact with each other. This approach in learning stems from a constructivist approach to learning and is well aligned with the PBL approach implemented at Aalborg University, Denmark. Therefore, we believe that it can also be used for other subjects (e.g. image processing, sound computing) for enhancing their PBL character.

7 Conclusion

In this article, we presented our idea of using a game engine (Unity) to teach Media Technology students mathematics-related concepts. In order to observe how the introduction of a technological tool, namely the game engine, changes the practices in mathematical work, we adopted the anthropological approach in didactics. This theoretical framework defines the "atoms" of mathematical practice and discourse in terms of tasks, techniques, technologies and theories (Winsløw, 2012). We presented a didactical scenario when Unity is used for introducing the calculation of reflection and refraction vectors and then we used the anthropological approach to present the practices in calculating these vectors with traditional mathematics and constructing them in Unity. Then, we discussed the differences between the two cases, when we argued that Unity can benefit Media Technology students, who use mathematics as a tool. However, the aforementioned assumptions on the mathematical practice while using Unity will have to be confirmed in actual educational settings. The instrumental approach presented in section 3.2 can be used to verify that this tool is used as intended and results in authentic epistemic mediations. We are currently performing observations and interviews with students at Media Technology in order to verify these assumptions. So far, student reaction to this new approach has been positive and anecdotal feedback shows that the PBL oriented approach is preferable from the traditional way of teaching mathematics.

References

Ainley, J., Pratt, D., & Hansen, A. (2006). Connecting engagement and focus in pedagogic task design. British Educational Research Journal, 32(1), 23-38.

Artigue, M. (2002). Learning mathematics in a CAS environment: The genesis of a reflection about instrumentation and the dialectics between technical and conceptual work. International Journal of Computers for Mathematical Learning, 7(3), 245-274. doi:10.1023/A:1022103903080

Badran, I. (2007). Enhancing creativity and innovation in engineering education. European Journal of Engineering Education, 32(5), 573-585. doi:10.1080/03043790701433061

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

Bingolbali, E., Monaghan, J., & Roper, T. (2007). Engineering students' conceptions of the derivative and some implications for their mathematical education. International Journal of Mathematical Education in Science and Technology, 38(6), 763-777.

Breidenbach, D., Dubinsky, E., Hawks, J., & Nichols, D. (1992). Development of the process conception of function. Educational Studies in Mathematics, 23(3), 247-285. doi:10.1007/BF02309532

Chevallard, Y. (1990). On mathematics education and culture: Critical afterthoughts. Educational Studies in Mathematics, 21(1), 3-27. doi:10.1007/BF00311013

El-Nasr, M. S., & Smith, B. K. (2006). Learning through game modding. Computers in Entertainment (CIE), 4(1), 7.

Guin, D., Ruthven, K., & Trouche, L. (2006). The didactical challenge of symbolic calculators: Turning a computational device into a mathematical instrument Springer Science & Business Media.

Hoyles, C., & Noss, R. (1992). Learning mathematics and logo MIT Press.

Jørgensen, F., & Busk Kofoed, L. (2007). Integrating the development of continuous improvement and innovation capabilities into engineering education. European Journal of Engineering Education, 32(2), 181-191. doi:10.1080/03043790601116964

Lengyel, E. (2012). Mathematics for 3D game programming and computer graphics, third edition Delmar Cengage Learning.

Maull, W., & Berry, J. (2000). A questionnaire to elicit the mathematical concept images of engineering students. International Journal of Mathematical Education in Science and Technology, 31(6), 899-917.

Morgan, A. (1990). A study of the difficulties experienced with mathematics by engineering students in higher education. International Journal of Mathematical Education in Science and Technology, 21(6), 975-988.

Nardi, B. A. (1996). Context and consciousness: Activity theory and human-computer interaction Mit Press.

Papert, S. (1980). Mindstorms: Children, computers, and powerful ideas Basic Books, Inc.

Prensky, M. (2001). Digital game-based learning New York: McGraw-Hill.

Rabardel, P., & Bourmaud, G. (2003). From computer to instrument system: A developmental perspective. Interacting with Computers, 15(5), 665-691. doi:http://dx.doi.org/10.1016/S0953-5438(03)00058-4

Triantafyllou, E., & Timcenko, O. (2013). Applying constructionism and problem based learning for developing dynamic educational material for mathematics at undergraduate university level. PBL Across Cultures : Proceedings from the 4th International Research Symposium on PBL 2013, Kuala Lumpur, Malaysia. 335.

Triantafyllou, E., & Timcenko, O. (2014). Opportunities and challenges of using technology in mathematics education of creative engineering studies. In C. Stephanidis (Ed.), HCI international 2014 - posters' extended abstracts (pp. 171-176) Springer International Publishing. doi:10.1007/978-3-319-07854-0_31

Verillon, P., & Rabardel, P. (1995). Cognition and artifacts: A contribution to the study of though in relation to instrumented activity. European Journal of Psychology of Education, 10(1), 77-101. doi:10.1007/BF03172796

Winsløw, C. (2012). Mathematics at university: The anthropological approach. Lecture, ICME-12, 887-901.

Zhou, C. (2012). Fostering creative engineers: A key to face the complexity of engineering practice. European Journal of Engineering Education, 37(4), 343-353. doi:10.1080/03043797.2012.691872

Appendix

Class activity on light reflection and refraction: Open the Unity project “Math”, which contains two scenes: the Reflection scene (Figure 3) and the Refraction scene (Figure 4). When in play mode, you can use your arrow keys to rotate the pen in space. Use your mouse scroll keys in order to zoom in and out in the scene and your left mouse key in order to rotate the camera (that means your own view on the scene).

Open the Reflection scene:

1. This scene contains a pen, which emits a beam of light. The beam of light is represented by a red line. In order to draw this line and its reflection on the plane (assume that the plane is a mirror, on which the light reflects), there is a script attached on the pen. Open the script in order to see the code.

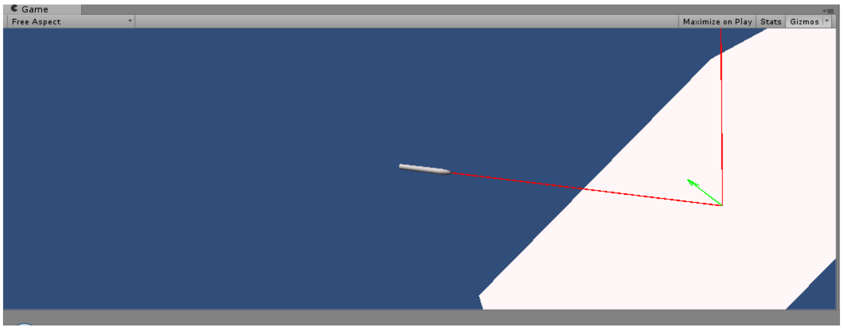

Figure 3: The reflection scene in Unity

2. The code uses the Unity method Reflect(); in order to calculate the direction of the reflected light. Delete this line of code (or make it a comment by adding // at the beginning of the line) and then calculate the direction of the reflected line by using the formula of the reflection:

$$\boldsymbol{R} = 2(\boldsymbol{N} \cdot \boldsymbol{L})\boldsymbol{N} - \boldsymbol{L}$$

3. Suppose (or actually try to do it!) that we substitute the plane with a rough surface (e.g. a terrain with mountains). What adjustments (if any) do you have to do in the code of the pen script for calculating the reflected line?

Open the Refraction scene:

1. This scene contains again a pen, which emits a beam of light. The beam of light is represented by a red line. In order to draw this line and its refraction on the plane (assume that the plane is the interface between two media, e.g. air and water), there is a script attached on the pen. Open the script in order to see the code.

2. Complete the code for drawing the refracted line, by using the formula of the refraction:

$$\boldsymbol{T} = \left(\frac{\eta_L}{\eta_T}\boldsymbol{N} \cdot \boldsymbol{L} - \sqrt{1 - \frac{{\eta_L}^2}{{\eta_T}^2}[1 - (\boldsymbol{N} \cdot \boldsymbol{L})^2]}\right)\boldsymbol{N} - \frac{\eta_L}{\eta_T}\boldsymbol{L}$$

Keep in mind that in some cases total internal reflection can happen instead of refraction!

3. What changes do you have to make in the scene/code if you want to change the materials (e.g. instead of air and water, water and glass)

4. What happens if $\eta_L < \eta_T$?

5. Solve the following exercise by hand and then verify your answer in Unity. In order to check if total internal reflection happens on the critical angle you calculated, print the value of the angle in the console when total internal reflection occurs. Use the command Debug.Log(); for printing.

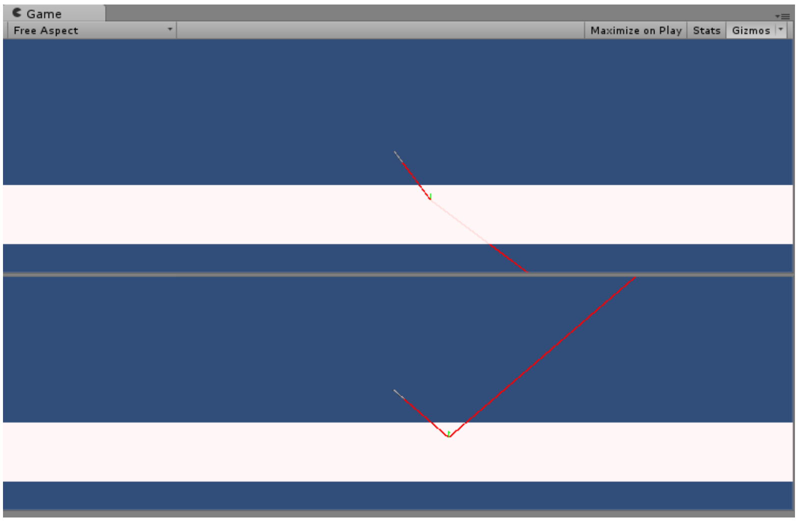

Figure 4: The refraction and the total internal reflection scene in Unity

Exercise:

The critical angel at the interface between two media is the smallest angle of incidence at which total internal reflection occurs. Determine the critical angle for a beam of light traveling upward through water toward the surface where it meets the air. The index of refraction of water is 1.33, and the index of refraction of the air is 1.00.

Tips for Unity programming:

If you want to see the details for one method or command in Unity, highlight the word you are searching for and then press Ctlr and ' in windows or Cmd and ' in mac. A browser window should open with details from the Unity API.

It is better to run your project having "Maximize on play" deactivated. This way you can observe better what happens in the scene.