Mathematics Learning by Programming in a Game Engine

Development of Knowledge and Student Motivation

Triantafyllou, Evangelia; Timcenko, Olga; Misfeldt, Morten

Published in:
International Journal of Engineering Education

Publication date:
2017

Mathematics Learning by Programming in a Game Engine: Development of Knowledge and Student Motivation

EVANGELIA TRIANTAFYLLOU
Department of Architecture Design and Media Technology, Aalborg University Copenhagen, A.C. Meyers Vaenge 15, DK-2450 Copenhagen SV, Denmark, evt@create.aau.dk
MORTEN MISFELDT
Research Lab: ICT and Design for Learning, Aalborg University Copenhagen, A.C. Meyers Vaenge 15, DK-2450 Copenhagen SV, Denmark, misfeldt@learning.aau.dk
OLGA TIMCENKO
Department of Architecture Design and Media Technology, Aalborg University Copenhagen, A.C. Meyers Vaenge 15, DK-2450 Copenhagen SV, Denmark, ot@create.aau.dk

Abstract: This paper emerges from our research focusing on mathematics education in trans-disciplinary engineering programs and presents a case study in such an engineering discipline, namely the Media Technology program at Aalborg University Copenhagen, Denmark. In this case study, we substituted traditional mathematics assignments with a programming project in a game engine (Unity) when teaching the reflection and refraction vector calculation. The main concept of this approach was that students get simple projects in Unity, where mathematics is used for game mechanics, and they have to modify or further develop these projects. We conducted interviews with nine students who participated in this case study, and we analysed their mathematical work in Unity. For analysing students' mathematical practice, we employed the anthropological theory in didactics and the instrumental approach. The analysis of student responses and projects provided insights on how students apply knowledge from a mathematical model to implement a physical model. This study shed light on students' misconceptions and difficulties but also on opportunities for them to challenge their understanding. Moreover, it revealed that students do not always internalize the mathematical knowledge they acquire, and they may get correct results without understanding their mathematical meaning. We conclude that this type of activities is more beneficial for these students compared to mathematical exercises, because they challenge their understanding and confront them with their misconceptions.

Keywords: mathematics; the anthropological approach; instrumental genesis; game engine; programming; media technology

1. Introduction

Over the past years, a number of engineering programs have arisen that transcend the division between technical, scientific and artistic 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.

Mathematical education in traditional engineering studies has been extensively studied. Morgan [1] collected data on first year engineering students in a period of five years and expresses his concern at their overall mathematical ability. Moreover, he identified common errors and certain areas of mathematics, which appear to be difficult to many engineering students. Maull and Berry [2] found that engineering students hold different mathematical concept images compared to mathematics students. Bingolbali, Monaghan, and Roper [3] reported that engineering students see mathematics as a tool, and therefore wish to see the application side as part of the course.

This paper emerges from our research focusing on mathematics education in trans-disciplinary engineering programs and presents a case study in such an engineering discipline, namely the Media Technology program at Aalborg University Copenhagen, Denmark. In this case study, we substituted traditional mathematics assignments with a programming project in a game engine (Unity). We chose Unity because Media Technology students are familiar with this environment and we wanted to avoid the learning effort of employing a new tool. The main concept of this approach was that students get simple projects in Unity, where mathematics is used for game mechanics, and they have to modify or further develop these projects. In the following, we review research on mathematical learning by programming and we present a theoretical framework for analysing mathematical practice

when programming in Unity is employed as a learning method. Then, we present our case study, where we employed programming in Unity for teaching light reflection and refraction to Media Technology students. Moreover, we comment on data gathered by interviewing nine students, who participated in our case study, and we conclude this paper with a discussion of the affordances of Unity as a mathematics learning environment.

2. Background

The use of programming to develop or enforce mathematical ideas has been proposed by several researchers. In the late 1960s, Seymour Papert inspired by the principles of constructivism 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 [4]. His suggestion was that children can familiarize themselves with mathematics concepts while developing constructs or predicting turtle's movement in LOGO. Papert described LOGO as a "mathematical microworld" that allows children to engage in such projects. In the 1980s, researchers were convinced that LOGO and other programming languages would radically reform mathematics teaching. However, the reality did not live up to the expectations. There are various reasons for the disappointing results. For instance, student work in LOGO can become non-mathematical, as students easily overlook the pieces of mathematical knowledge [5, 6].

Programming in mathematics education has been also introduced in secondary and upper-secondary education. Instead of using LOGO, teachers have introduced common programming languages such as BASIC and PASCAL to support learning. In the 1990s, Dubinsky proposed that the use of programming helps in mathematical learning by making abstract concepts concrete and even by introducing students to a syntax involving arithmetic variables and arithmetic operations [7]. He developed also a theoretical framework (the APOS Theory), which views mathematical knowledge as gained when individuals perceive mathematical problem situations and construct mental actions, processes, and objects organized in schemas to make sense of these situations and solve the problems [8]. This theoretical framework was applied to improve the development of the process conception of function in undergraduate mathematics and employs computers for enriching the numerical calculations that constitute the necessary foundation for concept formation [9].

During the last years, digital games have been applied in many educational fields to enhance learning motivation [10]. Since game environments (i.e. 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 [11]. 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 science 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 technological tools for learning mathematics shape the development of mathematical meanings and mathematical knowledge and at the same time they are shaped by student conceptions [12]. Hoyles and Noss underlined the fact that technology per se does not influence significantly the development of mathematical concepts. It is the design of such environments and the activities that take place in these that can support mathematical development and specific learning objectives. However, pedagogic task design is not a trivial procedure. Ainley, Pratt, and Hansen [6] for instance have discussed the tension that teachers face when they design tasks (known as the planning paradox). If teachers plan guided by learning objectives, the tasks are likely to lack reward for students and mathematical richness. On the other hand, if teachers plan from tasks, they may increase students' engagement but then it is difficult to focus on student activity and assess their learning. In order to observe types of mathematical learning and knowledge transfer when students work in Unity, we adopted the theoretical framework initiated by Chevallard [13] and developed by Artigue [14], namely the anthropological approach in didactics, and the theory of instrumental genesis, which was initially introduced in cognitive ergonomics [15].

3.1 The anthropological theory in didactics

The anthropological theory in didactics provides tools to model mathematical and didactical knowledge [14]. This didactical theory views mathematics as the product of a human activity of study of types of problems. In this theory, two inseparable aspects of mathematical activity are identified [16]. On one hand, the *practical block* (or know-how) formed by types of problems (or *problematic "tasks[i]"*) and the by the *"techniques"* used to solve them. Thus, "tasks" consist of studying a mathematical problem of a given type, e.g. calculate the limit of a function. The term "technique" is used in this theory in a very broad sense to refer to what is done to deal with a problematic "task", e.g. there are different "techniques" to calculate the limit of a function. The anthropological theory assumes that the accomplishment of any "task" requires the existence of a "technique". Another assumption of this theory is that mathematical productions are framed by the social and cultural contexts where they develop [17]. Therefore, there is on the other hand the *knowledge block* of mathematical activity, which provides the mathematical discourse necessary to interpret and justify the practical block. This block is structured in two levels: the *"technology"* that is the discourse used in order to both explain and justify the "technique" used and explain how to apply and distinguish a whole set of "techniques", and the "theory", which provides a structural basis for the technological discourse itself and can be seen as a "technology" of the "technology" [14]. At a higher level of discourse, "technologies" are developed, explained, related and justified in and by a "theory" [16]. Types of problems, "techniques", "technologies" and "theories" are the basic elements of the anthropological model of mathematical activity and form what is called *mathematical praxeologies* [17]. The word praxeology indicates that practice and the discourse about the practice always go together (Fig. 1). 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.

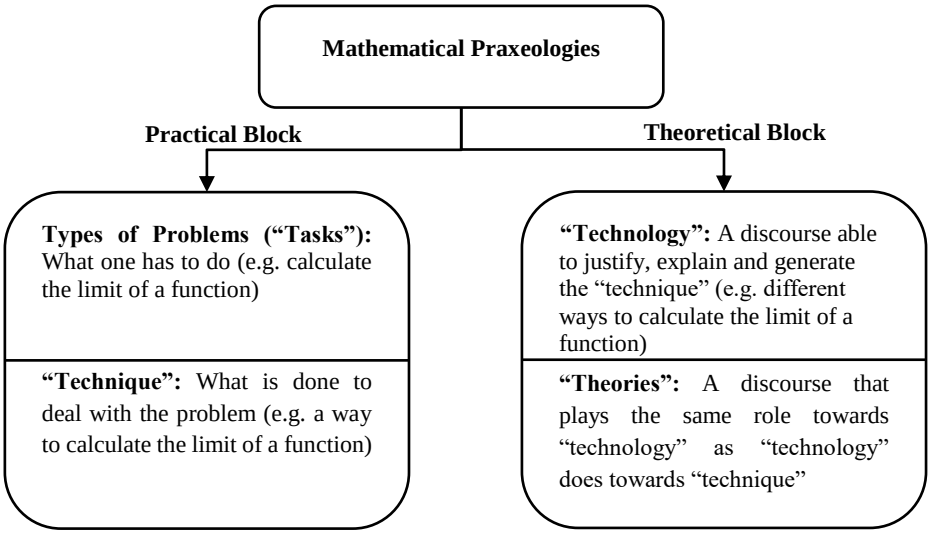


Fig. 1 The anthropological model of mathematical activity

3.2 The instrumental genesis

In order to address the purposefulness of technological tools for mathematical learning, researchers have adopted the instrumental genesis that originated in ergonomics. The instrumental genesis involves the process of instrumentation, i.e. when a person utilizes a tool for a goal-directed activity, which is shaped by the use of this tool, and the process of instrumentalization, i.e. when the goal-directed activity of the person reshapes the tool [18]. This approach has been used to discuss the integration of technology in mathematics classrooms and to distinguish between pragmatic mediations, i.e. using technology to solve tasks, and epistemic mediations, i.e. learning with technology [19]. Finally, Rabardel and Bourmaud introduced the concept of sensitivity referring to the orientation of mediations. Instrumented mediations can be directed towards the solution of a task, other subjects, and the person herself (as a reflective process).

In this article, we present a case study that was conducted in the Media Technology program at Aalborg University Copenhagen. This study addressed the following research questions:

- What kind of mathematical praxeologies exist in the mathematical domain and in the domain of using mathematics for programming for Media Technology students?
- Does the use of mathematical knowledge in a game development environment (Unity) support conceptual understanding for Media Technology students?
- How does the use of mathematical knowledge in a game development environment (Unity) affect self-reported conceptual understanding and motivation for Media Technology students?

In this study, we adopted motivation as defined by the self-determination theory [20]. The self-determination theory distinguishes between two types of motivation: intrinsic and extrinsic. Intrinsic motivation refers to those actions that individuals engage in because they are inherently interesting and enjoyable while extrinsic motivation refers to those actions that individuals engage because they lead to separable outcomes [21]. However, our aim was not to measure the effect on specific types of motivation but rather to investigate if and how students perceived that this intervention affected their motivation, without explicitly mentioning the intrinsic or extrinsic terms.

4. Methods

In this study, we employed qualitative methods, since the purpose was to investigate how students approach the two domains, what kind of conceptions appear and what is the effect on student motivation [22]. We aimed at understanding how students perceived the two domains and how they make meaning of them. Moreover, we conducted a case study because we were interested in documenting the process of introducing a new mathematical learning domain along with the mathematics domain [23].

4.1 Context of the study

In order to explore if and how students develop mathematical concepts and transfer knowledge from the mathematical domain to an application-related domain, we have conducted a case study in fall 2014. In the context of teaching Media Technology students attending the 'Computer Graphics Rendering' course the mathematical calculation of reflection and refraction vectors, we designed an in-class activity in Unity (Appendix). The students who were not able to complete this activity in class or were absent could submit their answers online (in Moodle) up to one week after the lecture. The goal of this activity was that students apply the mathematical model of calculating these two vectors in order to 'construct' the reflection and the refraction of a ray in a given Unity model. Our hypothesis was that the connection between the programming tasks and the visualizations in Unity, and students' interest in game development would provide a meaningful context and minimize the need for instrumentation.

The activity had two parts. In the first part, students had to complete a Unity script in order to find the direction of a reflected ray by using the reflection formula. The direction of the ray to be reflected was defined by a vector pointing toward the reflecting surface and in opposition with the direction assumed in the reflection formula (vector **L** in Fig. 2 points toward the incoming light). We did this for two reasons: 1) the definition pointing toward the reflecting surface is compatible with ray definition in Unity, and 2) we were interested to see if students would realize this difference. Unity has a built-in function for calculating the reflection of a vector and by using this function we showed students what the reflected ray should look like. Moreover, we asked students to elaborate on reflection properties, e.g. dependency on reflecting surface's unit vector, consecutive reflections etc.

The second part concerned the construction of the refraction vector. The students had to use the refraction formula and add the appropriate code in order to calculate the direction of the refracted ray. In this part, the indices of refraction of the two materials were chosen in order for total internal reflection to be possible to happen. Therefore, this formula could not be applied when the incidence angle was larger than the critical angle. The students should first check the angle between the incoming ray and the surface's normal vector. If this angle (incidence angle) was smaller than the critical angle, then the refraction formula should be used for the direction of the refracted ray. In

case it was larger, the reflection formula should be used and the ray would be reflected instead of refracted. In this part, there was also a question regarding refractive indices of materials and how they affect refraction. Finally, students had to solve an exercise from their textbook. In this exercise, students should calculate the critical angle between two given materials. The students were then asked to verify the solution in Unity, i.e. change accordingly the refractive indices and check for which angles the ray is reflected instead of refracted. If their implementation was correct, they should observe that for angles equal or larger than the one they calculated, the ray is reflected. In this part of the exercise, we wanted to see if students could apply the solution to a mathematical problem for predicting the behaviour of the refraction model in Unity.

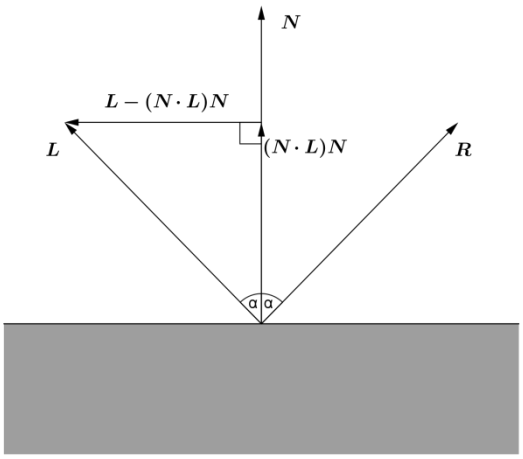


Fig. 2 The direction of reflection **R** forms the same angle with the normal vector **N** as the direction **L** pointing toward the incoming light. It is found by subtracting twice the component of **L** that is perpendicular to **N** from **L** itself [24]

4.2 Data acquisition and analysis

To investigate the students' ability to deal with problems in both domains, we conducted individual interviews with nine (N=9) students – six males and three females – from the ones who submitted a project on reflection and refraction in Unity. These students volunteered for the interview and served as individual cases. According to literature on qualitative research this number of individuals is sufficient for obtaining insights into process-related phenomena [25]. The interviews were carried out one to two weeks after the delivery of the project and focused mainly on how the students approached this project, their reasoning for solving it and if and how they transferred knowledge from one domain to another. Finally, we asked students to share their reflections on the experience of applying mathematical knowledge in Unity.

During the interviews, we first asked the students to describe the mathematical model of reflection and refraction and to clarify related concepts (angle of incidence, Snell's law, total internal reflection, etc.). Then, we asked them what specific steps they had taken in dealing with the project in Unity, how they interpreted the results, what challenges they experienced during the process and how they dealt with those challenges, and how they experienced the use of mathematical knowledge for programming in a game engine. Finally, we asked them if and how they find this experience motivating. The interviews were carried out in a semi-structured way and in some cases we used probing or elaboration questions, in order to clarify student answers. All interviews were video recorded and transcribed.

In order to gain insight on how the rest of the students enrolled in the 'Computer Graphics Rendering' course dealt with this project, we went through all the individual submissions for this assignment (N=60). We checked both the code for programming reflection and refraction and their text answers to the other questions. During this check, we did not find any approaches to this project that were different to the individual cases presented in this article. Therefore, we argue that the data presented in this article are representative for this group of students.

The data was analysed using an inductive approach for qualitative analysis [26]. During this data analysis, consensus on findings was sought among all authors of this article in order to ensure a deep reflexive analysis and to strengthen the validity of the findings. Furthermore, two of the authors were actively involved in the course, which greatly assisted in interpreting students' answers and experiences.

5. Results

A short description of individual cases' answers to the project in Unity is given in **Table 1**. The question numbers are the numbers given in the project description in the Appendix. For referring to the case students, we use pseudonyms to ensure anonymity. In the following, we discuss individual students' responses on reflection and refraction separately.

Table 1 Short description of individual cases' answers to the project in Unity

	1.Reflection	2.Surface	3.Refraction	4.Indices	5.Media	6.Cr. angle
John	Correct	Correct (Sphere)[a]	Wrong direction and no TIR[b]	Correct	Wrong (impossible)	Result (not Unity)
Bob	Correct	″	″	Correct	NA[c]	″
Lara	Correct	″	″	NA	NA	″
Anna	Correct	Correct (normal)	″	Correct	Wrong (impossible)	Correct (no explanation)
George	Correct	″	″	Correct	″	Result (not Unity)
Jim	Correct	″	Correct	Correct	Correct	Correct
Mary	Correct	Correct (plane)	Correct	NA	NA	NA
Nick	Correct	Correct (normal)	Correct but no TIR	Correct	Correct	Correct
Tom	Wrong direction	Correct (terrain)	Wrong direction	Correct	NA	Result (not Unity)

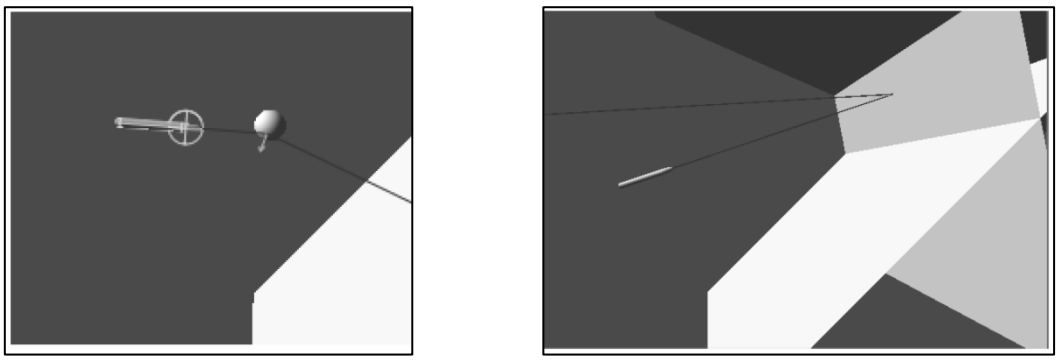
[a]In parentheses, keywords of student answers are given, [b]ITR=Internal Total Reflection, [c]NA= No Answer

5.1 The reflection vector

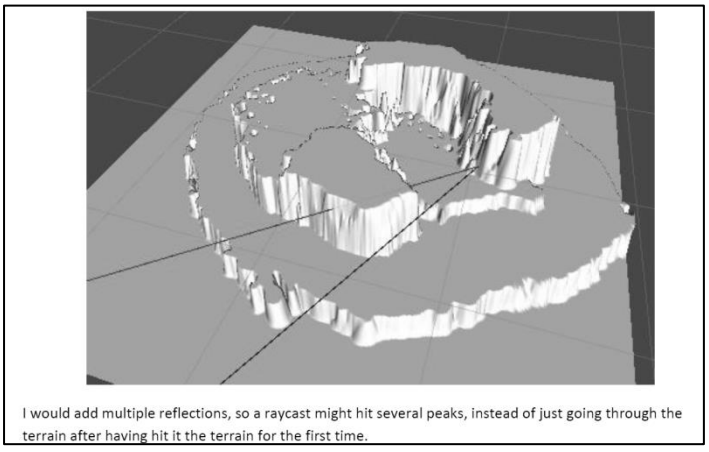
The course textbook contains the mathematical calculation of the reflection vector **R** given the direction **L** pointing towards the incoming light, which is shown in Fig. 2. We started the interviews by asking students to briefly explain the steps to complete this "task". Among the nine students, only two (Jim and Nick) could explain these steps and therefore were aware of the practical block of this praxeology. The other students could not recall the "technique" of calculating geometrically the dot product of two vectors, and Anna and Tom were even challenged by the "technique" of adding or subtracting vectors for finding other vectors. Regarding the theoretical block, all students were aware of the majority of the related "technologies" (length of vectors, addition and subtraction of vectors) but George, Anna and Tom had problems defining what a unit vector is and how the dot product of two vectors is defined. As far as the related "theories" are concerned, Tom and Mary found it difficult to explain the law

of reflection when given the diagram of Fig. 2. Mary suggested that according to this law, the incidence and the reflection angle are always complementary angles while pointing to the diagram of Fig. 2. Tom used the right articulation but he suggested that the incidence angle is the complementary angle of angle α in Fig. 2.

Regarding the part of programming the reflection of a ray in Unity, only Tom did not succeed to construct the right reflection. This was due to the fact that the ray was defined in the opposite direction of the direction of vector **L** in the mathematical model. The other case students defined a new vector either by inverting the ray direction vector or by using two points between the plane and the source of light, and then inserted this to the reflection formula. Tom used directly the direction vector of the ray in the reflection vector and therefore he drew a reflection vector pointing to a wrong direction. All students could explain how vectors **R**, **L**, and **N** are translated in Unity code. Regarding the required "techniques", all case students could use the appropriate functions in Unity in order to normalize vectors, calculate the dot product of two vectors, and define a ray by a point and a direction vector. In the second question on reflection (see Appendix), all case students suggested that since the normal vector of the surface is used for calculating the reflection vector, no changes have to be made in the code if the plane is substituted by another surface. The students provided different arguments for justifying their answers. Jon, Bob and Lara replaced the plane with a sphere in order to show that their program works also fine in this case, Mary inserted a vertical plane to the initial one, and Tom replaced the plane with a terrain with gradual changes in height and noted that successive reflections would happen in that case (Fig. 3). Anna, George, Jim and Nick noted correctly that the normal vector is calculated every time based on the current reflection surface without providing any visual explanation.



(a) (b)



(c)

Fig. 3 John's (a), Mary's (b), and Tom's (c) answers to the second question of the project

5.2 The refraction vector

The course textbook contains the mathematical calculation of the refraction vector **T** given the direction **L** pointing towards the incoming light, which is shown in Fig. 4. During the interviews, we asked the students to explain the steps of this calculation (practical block). Again, only Jim and Nick could explain all steps of this "task". The other case students faced the same difficulties as in the reflection part but in this case they were also challenged by the decomposition of a vector to its parallel and perpendicular components (e.g. for vector **T**) and by the use of trigonometric identities to simplify the refraction formula. Regarding the theoretical block, they all were aware of the definition of the trigonometric functions and the Pythagorean Theorem, which is used for getting the fundamental trigonometric identity. As for the Snell's law relating the angle of incidence and the angle of transmission with the refractive indices of the materials, Bob and Tom reported that they haven't seen it before since they only read the part on the refraction formula from their textbook.

In the part concerning Unity implementation, students' responses during the interviews showed that they were aware of all the related "techniques". However, only Jim and Mary checked the condition for distinguishing between refraction and total internal reflection in their projects (Fig. 5). The other students attempted to draw the refracted ray even when this was not possible, since there was no specific warning by the compiler for attempting to place a negative number under a square root in Equation (2). However, no ray was drawn in this case. All students apart from Jim, Mary and Nick had also difficulties in drawing the refraction vector in the right direction. Even students, who had successfully drawn the reflection vector, forgot in this case to invert the direction vector of the ray before inserting it to the refraction formula (Fig. 6). The students mentioned in their projects that the refraction did not look right but they could not find a way to correct it. The students had again no problem during the interviews in matching the vectors and angles of the mathematical model with objects and variables in the Unity model and code.

During the interviews, all students could answer the question on how to change the materials in the Unity model, noting that only the values of the refractive indices have to be changed. Lara and Mary were also able to answer although they did not submit any answer for this question, and reported having overlooked it. However, question 5 (see Appendix) was more challenging for students, since only Jim and Nick mentioned in their projects that in this case total internal reflection never happens. From the rest, the ones, who answered, argued that it is not possible for the ratio of the refractive indices to be less than one.

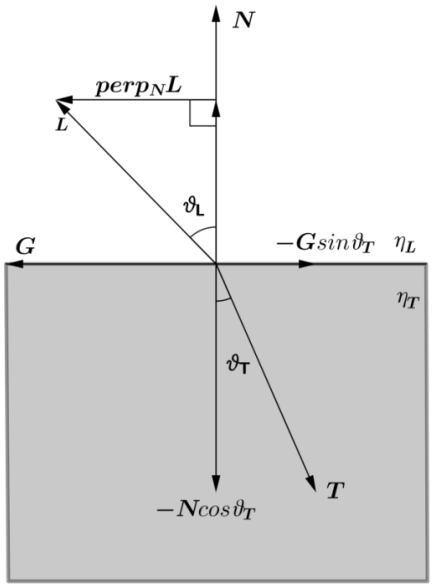


Fig. 4: The angle of incidence **L** and the angle of transmission **T** are related by Snell's law. The refraction vector **T** is expressed in terms of its components parallel and perpendicular to the normal vector **N** [24]

The final question of the project (question 6 - Appendix) required students to find the critical angle between two materials using the appropriate mathematical formula, and then adjust their Unity code in order to validate this answer, i.e. students should observe total internal reflection to happen instead of refraction for angles equal or larger that the critical angle they calculated. All case students but Mary (who argued that she overlooked all last three questions of the project) could find the formula to use from their textbook and calculated the critical angle. Nevertheless, when the case students were asked what the critical angle is during the interview, five of them could not give a right answer. Mary suggested that the critical angle is always equal to 90º, while John, Bob, George, and Lara said that they could not remember what the critical angle represents.

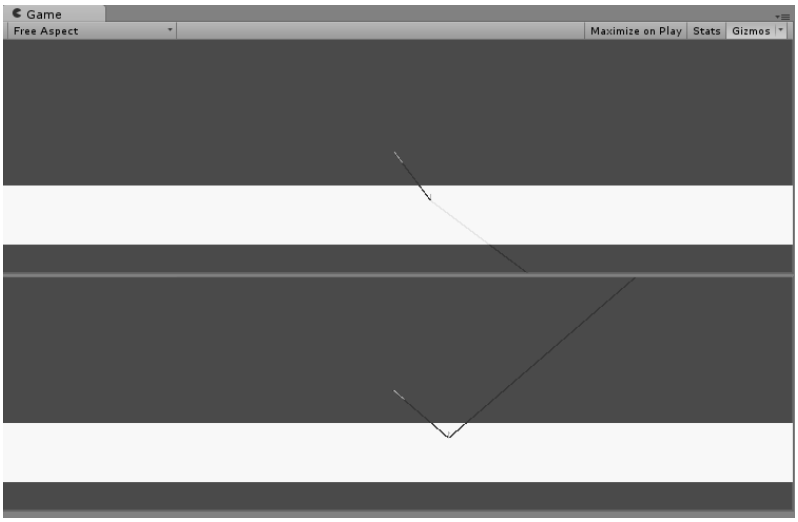


Fig. 5 Jim's correct implementation of refraction and total internal reflection

For question 6, Jim, Nick and Anna tried to validate their answers in the Unity model. Jim and Nick successfully did that but Anna got confused during this attempt, because she hadn't implemented total internal reflection in her project and also did not realize that Unity angles are given in radians instead of degrees. This confrontation made her question her overall approach in this part of the project. She wrote:

> '*For some reason my program does not work, as it should. When I use the Debug.Log(); to print out Asin((hT/hL)); [Asin=inverse sine function in Unity, hT and hL the refractive indices] it does not give the right answer. It says that the angle is 0.8509° [the answer she got (48.75°) in radians] each time no matter how I change the hT and the hL value. It might be because the formula to begin with has been written wrong. I cannot find a way to fix it. I also think that my refraction looks wrong in the scene so it is very much possible that I made it wrong*'.

5.3 Motivational aspects

During the interviews, the case students were prompted to reflect on their experience with using Unity and programming in a mathematics-related topic. Firstly, we asked them what they did to match the mathematical model with the 3D model in Unity. All students reported that they used the visual representation to guide them to this translation and that it was quite straightforward to assign to Unity objects their mathematical meaning. Moreover, they argued that this translation increased their understanding on the mathematical model.

Secondly, we asked them to compare this project with mathematical exercises they had to solve during previous lectures of the same course. The case students reported that they found this project more challenging but at the same time more motivating. According to the students, the aspects that increased motivation were the visual feedback in Unity and the use of a tool associated with computer game development, which they had used before. By visual feedback, the students meant the ability to observe visually their implementation and how it was affected by changes they were making to it and the ability to move the camera (point of view) in Unity in order to observe the scene

from different angles. Finally, students mentioned that applying mathematics in Unity provides them with an experience of how mathematics is applied outside of the university by professionals in this field.

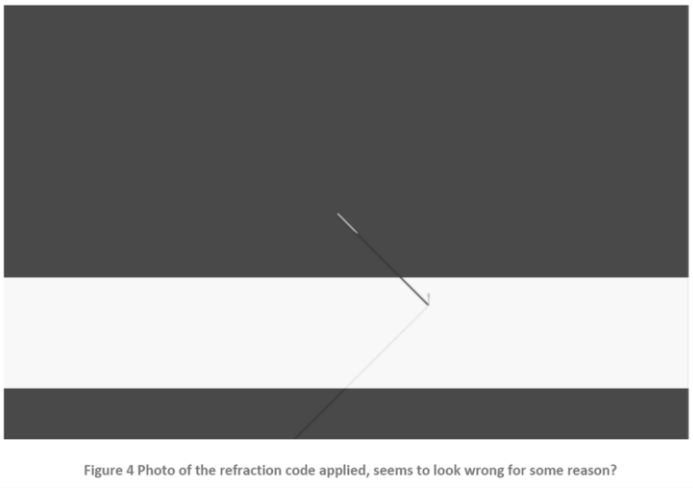


Fig. 6 Anna's wrong refraction and her comment

6. Discussion

The project given to students and discussed in the previous section required knowledge from three different domains: physics, mathematics, and programming. Students should understand the physical laws of the reflection and refraction and based on them study the mathematical calculation of the reflection and refraction vectors respectively. Then, they had to apply this understanding for constructing these vectors in a programming environment. Therefore, we will refer to these distinct domains while discussing the results in the following paragraphs.

The results presented in the previous section indicate that most of the Media Technology students, who participated in our study, were challenged when they had to use their general mathematical knowledge for understanding the mathematical model of reflection and refraction. The students for instance could add or subtract two random vectors, but they found it difficult to understand how one vector is expressed as the addition or subtraction of two other vectors in this specific case. The same applies for most of the "theories" and "technologies" and their related "techniques" and "tasks" in the mathematical domain. Nevertheless, we observed that not all students were aware of the "theories" regarding reflection and refraction that steam from physics. We hypothesize that these students belong to one of these two categories: 1) students who did not study their textbooks before attempting to work in the project, or 2) students who studied and used these "theories" while working in their project but they did not internalize them in order to be able to recall them a few weeks later.

In the Unity programming part, most of the students exhibited the opposite behaviour. They were well acquainted with "tasks" and "techniques" in Unity, but in some cases they were missing the theoretical block that would make these "tasks" contextualized and meaningful. There were for instance students like Mary, who successfully programmed the reflection and refraction vectors but failed to elaborate on this implementation. In such cases, instrumentalization was impeded by the lack of the conceptual understanding of the related concepts.

There were also students who used creatively the affordances of this environment for testing their assumptions (e.g. insert other reflection surfaces) or validating their answers (critical angle), like Jim. Out of sixty submissions, fourteen successfully addressed all questions of the project (about 23%). In these submissions, we observed that students used many screenshots to enrich their responses. We argue that the Unity environment helped students to describe and visualize their activity.

On the other hand, we experienced students like Anna, who realized that their implementation was faulty by checking the visual characteristics of it, but were unable to find the means to correct it (e.g. the refracted ray disappears for some angles, the refracted ray looks weird). Anna understood the physical laws of reflection and refraction, so she knew how the final result should look like. We believe that this visual confrontation was beneficial for students because it provoked at least some reflection on their practice. It may have not increased their conceptual understanding but it created a kind of a cognitive conflict that could challenge their understanding [27].

As far as the last part of the project is concerned, the majority of students were able to apply the right formula and calculate the critical angle between two given materials. However, only a few were able to interpret the result and connect its meaning to their Unity implementation (i.e. for angles larger than this angle, total internal reflection should happen instead of refraction). We therefore argue that such students do not benefit by their routine assignments, i.e. mathematical exercises, where they are just asked to apply specific mathematical formulas. Since they are lacking conceptual understanding, they very easily overlook the meaning of the "tasks" they perform. Our results showed that the same may apply to programming tasks, as the case of Mary shows. However, the individual cases reported such mathematical problem solving to be easier compared to this Unity project, and we believe that this feeling comes from the fact that this project challenged their knowledge and their understanding.

In regard to motivational aspects, all individual cases reported that they felt more engaged while working in Unity. They felt that this project was a way to integrate domain-specific applications to mathematics and to create a connection between student practice and professional practice. Moreover, they argued that the effort of matching a mathematical model with a Unity model increased their mathematical understanding. Finally, they appreciated the visual feedback of such environments and their interactive nature regarding adjustments.

7. Conclusion

In this paper, we presented a case study where programming in a game engine is employed for mathematical learning. The case study took place in the Media Technology program at Aalborg University Copenhagen, Denmark and in the context of the 'Computer Graphics Rendering' course. While teaching undergraduate students the calculation of the reflection and refraction vectors, we substituted class activities and homework, consisting of traditional mathematical exercises, with a programming project in Unity (game engine). The students had to add the appropriate code in a given Unity project in order to 'construct' the reflection and refraction of a ray. Sixty students submitted their answers to this project and we conducted interviews with nine of them. The interview data and the projects submitted shed light on students' misconceptions and difficulties but also on opportunities for them to challenge their understanding. Moreover, they revealed that students do not always internalize the mathematical knowledge they acquire, and they may get correct results without understanding their mathematical meaning. The students reported increased motivation and engagement while working in this project. Therefore, we argue that this type of activities is more beneficial for these students compared to mathematical exercises, because they challenge their understanding and confront them with their misconceptions. Finally, we would like to mention that it was not possible at this stage to evaluate the impact on learning of this intervention quantitatively. A future study spanning a longer period of time and covering more mathematical topics should address this matter.

References

1. A. Morgan, 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), 1990, pp. 975-988.

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

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

4. S. Papert, *Mindstorms: Children, computers, and powerful ideas,* Basic Books, Inc.,1980.

5. C. Hoyles and R. Noss, *Learning mathematics and logo,* MIT Press,1992.

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

7. E. Dubinsky, ISETL: A programming language for learning mathematics, *Communications on Pure and Applied Mathematics,* **48**(9), 1995, pp. 1027-1051.

8. E. Dubinsky and M.A. McDonald, APOS: A constructivist theory of learning in undergraduate mathematics education research, *The teaching and learning of mathematics at university level: An ICMI study,* 2001, pp. 273-280.

9. D. Breidenbach, E. Dubinsky, J. Hawks, D. Nichols, Development of the process conception of function, *Educational Studies in Mathematics,* **23**(3), 1992, pp. 247-285.

10. M. Prensky, *Digital game-based learning,* New York: McGraw-Hill,2001.

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

12. C. Hoyles and R. Noss, The technological mediation of mathematics and its learning, *Human development,* **52**(2), 2009, pp. 129-147.

13. Y. Chevallard, On mathematics education and culture: Critical afterthoughts, *Educational Studies in Mathematics,* **21**(1), 1990, pp. 3-27.

14. M. Artigue, 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), 2002, pp. 245-274.

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

16. C. Winsløw, Mathematics at university: The anthropological approach, *Selected Regular lectures from the 12th international congress on mathematical education*. S. J. Cho (ed.). Springer Publishing Company, 2015, pp. 859-875.

17. J. B. Farré, M. Bosch, L.B.E. Salfate, J. Gascón, Didactic restrictions on the teacher's practice, *Educational Studies in Mathematics,* **59**(1-3), 2005, pp. 235-268.

18. P. Rabardel and G. Bourmaud, From computer to instrument system: A developmental perspective, *Interacting with Computers,* **15**(5), 2003, pp. 665-691.

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

20. D. Edward and R. Ryan, Intrinsic motivation and self-determination in human behavior, *New York, Pantheon,*1985.

21. L. Abeysekera and P. Dawson, Motivation and cognitive load in the flipped classroom: Definition, rationale and a call for research, *Higher Education Research & Development,* **34**(1), 2015, pp. 1-14.

22. M. Borrego, E.P. Douglas, C.T. Amelink, Quantitative, qualitative, and mixed research methods in engineering education, *Journal of Engineering Education,* **98**(1), 2009, pp. 53-66.

23. S.B. Merriam, *Qualitative Research and Case Study Applications in Education. Revised and Expanded from" Case Study Research in Education.",* ERIC, 1998.

24. E. Lengyel, *Mathematics for 3D Game Programming and Computer Graphics, Third Edition,* Delmar Cengage Learning, 2012.

25. A. J. Onwuegbuzie and N.L. Leech, A call for qualitative power analyses, *Quality & Quantity,* **41**(1), 2007, pp. 105-121.

26. M.B. Miles and A.M. Huberman, *Qualitative data analysis: An expanded sourcebook,* Sage, 1994.

27. R. A. Rahim, N.M. Noor, N.M. Zaid, Meta-analysis on element of cognitive conflict strategies with a focus on multimedia learning material development, *International Education Studies,* **8**(13), 2015, pp. 73-78.

Biography

Evangelia Triantafyllou is a postdoc fellow in the Department of Architecture Design and Media Technology at Aalborg University Copenhagen, Denmark. She obtained the M.A. degree in Electrical and Computer Engineering at Aristotle University of Thessaloniki, Greece, in 2000 and the P.D.Eng (Professional Doctorate in Engineering Design) degree in ICT at Eindhoven University of Technology, The Netherlands, in 2004. She subsequently worked as a computer science teacher on various educational levels and received her Ph.D. on ICT-based teaching methods for improving university mathematics learning at Aalborg University Copenhagen, Denmark in 2015. Evangelia has authored/co-authored more than 25 scientific publications. Her research interests include technology-enhanced learning in mathematics, active learning and university mathematics education.

Morten Misfeldt, professor, is research manager of the research lab for ICT and design for learning, Aalborg University Copenhagen, Denmark. He holds a Ph.D. from the Learning Lab, Department of Education in Aarhus University, Denmark. His research interests include technology enhanced learning, university mathematics education, the use of ICT in primary level mathematics education, the influence of ICT on mathematics curriculum and mathematical practices in various areas of society. Morten is also research manager of the Danish GeoGebra Institute and has authored/co-authored more than 100 scientific publications.

Olga Timcenko, associate professor, holds a Ph.D. in robot control, but has researched for many years in virtual environments and user interfaces for educational purposes, as a member of the LEGO Systems, Billund (Denmark) Business development unit, where she worked on projects like LEGO Mindstorms and LEGO Universe. She was the LEGO team lead in the FP6 EU Network of Excellence KALEIDOSCOPE: Concepts and Methods for Exploring the Future of Learning with Digital Technologies (2004-07), where she participated in several JERPs. After joining the Department of Architecture Design and Media Technology, Aalborg University Copenhagen in 2006, she focused her research on technology enhanced teaching and learning, especially mathematics at university level. She was a member of LEAF - Nordic research network for Learning Ecosystems and Activities of the Future (2010-2012), and now she is a member of the ReCreate research centre. She is author / co-author of 50+ conference and journal papers in the field of robotics and children and technology, and of four international patents in the field of virtual 3D worlds / 3D user interfaces.

Appendix

Class activity on light reflection and refraction:

Open the Unity project "Math", which contains two scenes: the Reflection scene and the Refraction scene. 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.

This scene contains a pen, which emits a beam of light. The beam of light is represented by a ray (Fig. 7). In order to draw this ray 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.

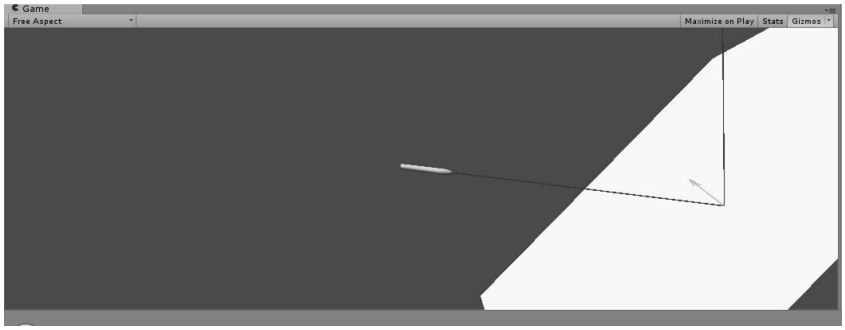


Fig. 7 The reflection scene in Unity

1. 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:

$$\mathbf{R} = 2(\mathbf{N} \cdot \mathbf{L})\mathbf{N} - \mathbf{L} \qquad (1)$$

2. Suppose (or even 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 ray?

Open the Refraction scene.

This scene contains again a pen, which emits a beam of light. The beam of light is represented by a ray. In order to draw this ray and its refraction while passing through 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.

3. Complete the code for drawing the refracted ray by using the formula of the refraction:

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

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

4. 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)?

5. What happens if $\eta L < \eta T$?

6. 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.

Exercise from (Lengyel, 2012):

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.

List of figures

List of tables

[i] Since the words task, technique, technology, and theory may have ambiguous meaning, we use quotes to refer to these words, when they are used to describe the elements of the anthropological model of mathematical activity.