Fun Maths for All Game Development Students

Diogo de Andrade

School of Communication, Arts and Information Tech.
Lusófona University
Lisboa, Portugal
diogo.andrade@ulusofona.pt

Nuno Fachada

HEI-Lab — Digital Human-Environment Interactions Lab
Lusófona University
Lisboa, Portugal
nuno.fachada@ulusofona.pt

ABSTRACT

We present an approach for teaching maths to game development undergraduates from a wide variety of backgrounds. We discuss the concepts behind an introductory maths course, its assessment and put forward some preliminary results on a practical implementation.

CCS CONCEPTS

• **Mathematics of computing**; • **Applied computing** → **Education**; *Computer games*;

KEYWORDS

Mathematics, Python, Game development

ACM Reference Format:

Diogo de Andrade and Nuno Fachada. 2020. Fun Maths for All Game Development Students. In *Proceedings of the 2020 ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE '20)*, June 15–19, 2020, Trondheim, Norway. ACM, New York, NY, USA, 2 pages. <https://doi.org/10.1145/3341525.3393992>

1 INTRODUCTION AND MOTIVATION

Mathematics is a core part of Computer Science (CS) education [7]. However, there are cases where although CS is an integral part of the curriculum, it is not the main focus. One such case is the Bachelor in Videogames at Lusófona University, a three year evenly interdisciplinary BA [3], open to students coming from various high school academic backgrounds, such as arts, sciences and even humanities. This large spectrum of backgrounds means that the motivation for each student to tackle maths is also diverse, and as such requires approaches well-suited for each group.

In previous years, students were lectured math subjects in a classical fashion, involving exposition of theorems and proofs. This generally led to poor results, as students did not fully grasp how the subject matter applied to game development. This problem was deepened by the fact that a lot of their role models in the industry are self-taught with no formal training, which further diminishes student motivation, even if current trends show that formal education in the field is on the rise [6].

2 COURSE METHODS AND GOALS

The rationale is to teach maths from a user's perspective. What is meant by this is that the proposed methods hinge on the idea that maths can be used without fully understanding the fundamentals and proofs that led to the theorems and equations, focusing more on creating recurrent mechanisms in the students.

For this we have two courses that work in tandem: Programming Fundamentals (PF) and Math and Physics for Games I (MPG1) – the latter being the focus of this presentation. In PF, we introduce students to the basics of programming, using Python and Pygame [4]. We then use that knowledge to build small applications and samples in MPG1, with the aim of not only solidifying the concepts behind the programming, but also of showing practical examples of the use of maths in real game development scenarios.

In terms of syllabus, we strive towards motivating the students to learn the more complex topics by themselves, showing them the fun of math and the potential satisfaction of seeing the pieces fit together if they delve a bit deeper into the mathematical proofs. We also do extensive use of video, by recording the whole classes and making them available through a YouTube channel, so that students can follow at their own pace in their homework.

Concerning the final assessment, evaluation is divided in theoretical and practical components. The former is carried out with two tests during the semester. The latter is done through a group assignment, described in the next section. The evaluation of this assignment follows a tiered approach, so that students that have reduced interest in math and/or do not have a science background are able to finish the project with some degree of success. The idea behind promoting group work is to open the possibility of students more comfortable with the subject matter to mentor others that are having more difficulties, thus establishing bonds that can be useful in the more group work-oriented later stages of the degree. We also require the use of Git as a tool for collaboration between group members and for downloading/cloning the project template. The delivery also has to be done in the form of a GitHub repository. Besides the code, students also have to submit a report on work breakdown and include a short postmortem of the assignment, so they can do some self-assessment and provide lecturers with information for improving future versions of the assignment.

3 GROUP ASSIGNMENT

The main goal of the assignment is to take an existing simple 3D renderer and add some functionality to it. This renderer, as well as all the required files and instructions, are made available via GitHub [2]. The renderer is built using Python and Pygame for the graphical side, with NumPy [5] and the quaternion package [1] used

---

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
ITiCSE '20, June 15–19, 2020, Trondheim, Norway
© 2020 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-6874-2/20/06.
<https://doi.org/10.1145/3341525.3393992>

Figure 1: Screenshot from one of the submitted projects, showing a lighted 3D object in the centre of the screen.

for matrix and quaternion handling, respectively. The assignment has several goals, namely:

- (1) Build a viewer application with the following functionality:
  - Display a 3D object in the centre of the screen, as shown in Figure 1. The user has to be able to control the visualization, rotating and moving the object around.
  - Create a model other than a cube for this display. The model can be loaded from a file or can be built in code. The model has to include sub-objects.
- (2) Build a first-person shooter (FPS)-like application, namely:
  - Create an environment where the player can roam using standard FPS controls. The environment can be just a series of cubes with different scales and positions.
  - Implement backface culling.
  - Implement filled geometry, replacing the wireframe.
  - Stop objects behind the camera from being rendered.
  - Implement very simple point lighting.

The minimum viable delivery is the viewer application, which should be easy to create, even for students that struggle with the mathematical concepts. The items in the second part of the project will improve the grade. Students have about one month to develop the project.

4 PRELIMINARY ASSESSMENT

The proposed approach has been implemented at the MPG1 course during the 2019/20 academic year. A total of 28 students were evaluated. With respect to the practical evaluation component, there were 16 assignments handed in (corresponding to 22 students), 15 of which were rated positive. Grading was done on a 0 to 20 scale. Most grades were around the 10 to 13 mark (mean 11.8, median 12), corresponding to students that did the most basic part of the assignment with at most one extra assignment goal. Most of the higher grades (>15) were done individually.

After the end of the semester, students of the last 3 years were asked to evaluate, on a scale of 1 to 5, their motivation with the project. The current year students ($n = 24$) evaluated the improved project, while students from the previous two years ($n = 40$) evaluated the classic math project. Results show an improvement on the

Figure 2: Student grade relation between Math and Physics for Games I (MPG1) and other courses in the same semester.

motivation, with the mean score going up from 2.55 to 3.25. Median and standard deviation stayed at around the same values (3 and 1, respectively).

Figure 2 shows the relation between the final MPG1 grades (with the theoretical and practical components) and three other courses in the same semester, namely Game Art I (GA1, art), Interactive Narratives (IN, game design) and PF (programming). Interestingly, we could not find any inverse grade correlation between MPG1 and the other courses, particularly the art classes, which leads us to think that the proposed approach is working even for students of art backgrounds. Of course, these results are very preliminary, as the degree-wide effects of these methodological changes will only be clear in a few years.

5 CONCLUSIONS

In this paper we presented an approach for teaching maths to game development undergraduates coming from a wide range of backgrounds. Although it is early to say if the proposed methodology will have degree-wide positive results, this field trial provided some interesting pointers which we believe are worth sharing with the intended audience for this Tips, Techniques and Courseware presentation. More specifically, even if the grades were not the ones we were hoping for, there was a clear growth in student motivation with the course, with students positively acknowledging the importance of maths in the context of game development.

REFERENCES

- [1] Mike Boyle. 2018. The quaternion package: Add support for quaternions to python and numpy (Version v2.0). Zenodo. <https://doi.org/10.5281/zenodo.1220425>
- [2] Diogo de Andrade. 2019. Final Project IMEJ1 2019/2020. GitHub. https://github.com/VideojogosLusofona/imej1_2019_projecto
- [3] Nuno Fachada and Nélío Códices. 2020. Top-down design of a CS curriculum for a computer games BA. In *Proceedings of the 2020 ACM Conference on Innovation and Technology in Computer Science Education (Trondheim, Norway) (ITiCSE '20)*. ACM, New York, NY, USA. <https://doi.org/10.1145/3341525.3387378>
- [4] Will McGugan. 2007. *Beginning game development with Python and Pygame: from novice to professional*. Apress.
- [5] Travis E Oliphant. 2006. *A guide to NumPy*. Vol. 1. Trelgol Publishing USA.
- [6] Suzi Stephenson. 2019. Number of graduates in computer games continues to rise. TIGA. <https://tiga.org/news/number-of-graduates-in-computer-games-continues-to-rise>
- [7] The Joint Task Force on Computing Curricula. 2013. *Computer Science Curricula 2013*. ACM and the IEEE Computer Society. <https://doi.org/10.1145/2534860>