Four years of sharing teaching practices within the French Computer Graphics community

C. Roudet[1] and F. Belhadj[2] and B. Sauvage[3] and M. Maria[4] and G. Gilet[5] and J-J. Bourdin[2]

[1]Université Bourgogne Europe - LIB - Dijon, France
[2]Université Paris 8 - LIASD - Paris, France
[3]Université de Strasbourg - ICUBE - Strasbourg, France
[4]Université de Limoges - XLIM - Limoges, France
[5]Université de Sherbrooke - BISOUS - Sherbrooke, Canada

Abstract

This paper describes and provides feedback on a Computer Graphics (CG) teaching initiative conducted by the French Association of CG (AFIG in French), as part of its annual national conference (called j.FIG). The AFIG, historically focused primarily on research and doctoral education, has been leading the French academic community in CG for 30 years. Since the beginning of 2021, it has launched a working group dedicated to CG teaching in the Bachelor's and Master's cycles. Its main action was to present panels during the j.FIG, to address issues related to CG teaching on a national scale. This is analyzed in detail in this paper. For each of the four organized panels so far, we present its main goals and the underlying discussions and repercussions, by comparing them with similar state-of-the-art initiatives. Possible actions and proposals to perpetuate the event are finally discussed. More broadly, our ambition is to obtain reactions and provoke necessarily enriching discussions, enabling everyone to escape a little from the teacher's solitude, alone in front of his class.

CCS Concepts

• Social and professional topics → Computing education programs; Model curricula; Adult education; • Computing methodologies → Computer graphics;

1. Introduction

The role played by visual communication in many everyday digital interactions means that innovation in this area requires both creativity and a broad spectrum of skills, many of which are related to Computer Graphics (CG). Thus, CG involves various domains (artistic, cultural, industrial, medical and research), skills in arts and sciences (mainly mathematics, computer science and physics) and technologies. Teaching and learning it in a scientific context requires being at the cutting edge of the latest technologies and programming improvements together with understanding the involved theories. To be able to master these in a given context (environment, framework and particular issues), a good portion of teaching time needs to be allotted to practical exercises. This is to be tempered given the needs of each particular curricula and their associated future prospects, to which the pedagogy must adapt. Developing and teaching such complete courses can be difficult and time consuming. Opening discussions at various geographic scales and/or giving possibilities of sharing online resources (courses, materials and codes) seem essential to create a community of practices and avoid becoming isolated.

The French Association of Computer Graphics (AFIG in French [AFI]) was created to federate, facilitate interactions and animate the French CG scientific community. One of its main mission is to co-organize each year (since 1993), the Computer Graphics conference (j.FIG in French). Historically, the French CG community was structured from the end of the 1980s with the first conference organized in 1988. Around a hundred of French experts in the field of CG (from research laboratories and industries) are present at this conference, half of whom being Ph.D. students working on the subject. It is also a special time for an award ceremony (best paper, best French Ph.D. thesis in CG), the organization of which is supported by the French Chapter of Eurographics. Through a three-day program, plenary conferences (with internationally recognized guest speakers) and summary presentations (by experienced researchers or industry players) alternate with a large space left for Ph.D. students to present their work. To supplement this program, workshops for reflections on scientific obstacles and new avenues of research also take hold.

Since November 2021, a special session (called teaching panel) was born with the main purpose of addressing issues related to teaching on a national scale. The organization of these panels is led by a Working Group (WG) within the association (called "teaching WG" and born in early 2021). Through a detailed analysis of these initiatives, along with comparisons with related studies, we aim to

DOI: 10.2312/eged.20251014

address in this paper, the following research questions: What are the views, interests, and experiences of the French CG academic community in teaching CG? What lessons have been learned from these national panels, organized to improve the educational practices of CG teachers?

The next sections are organized as follows. We first present the motivations behind the creation of these panels. We also detail their organization (ways of choosing speakers and to lead the discussion with the community). Next, we provide a historical overview of similar initiatives, led by various panels since 1983, in response to the numerous teaching recommendations or declarations that emerged in those years. We position all these initiatives in time and show how ours is complementary. When detailing the four panels, held between 2021 and 2024, we report the resulting discussions and debates, providing as far as possible, the most relevant direct quotes from speakers and the audience. We also relate each of these experiences with the current literature. The paper concludes by general feedbacks and perspectives for future panels.

2. Aims and organization of the panels

AFIG covers a variety of CG topics, such as image synthesis, 3D rendering, geometric modeling, physical simulation, animation, algorithmic and discrete geometry, visualization, or Virtual and Augmented Reality (VR/AR). Due to their specialization, these topics are rarely taught before the third year of the Bachelor's degree. They are more present in the Master's degree, with strong disparities from one university to another: the topics covered depend heavily on local research teams. The number of hours devoted to CG topics in the courses is sometimes modest, given the technicality and variety of the content. Also, the number of teachers within the same university is often limited. It thereby restricts the possibilities for exchanges that are nevertheless essential for sharing good practices and taking into account technical and pedagogical developments and innovations. For these reasons, we considered it useful to address issues related to teaching on a national scale. To this end, and as part of a new WG dedicated to CG teaching, we chose to create a session to discuss these issues at the annual national conference (j.FIG). We opted for a panel format because we believe it is conducive to exchanges and sharing of experiences, and it stimulates participation.

The session lasts between 1 and 1.5 hours and proposes each year a specific topic/concern. The chosen topics were selected from those suggested by members of the French CG community who responded to an online survey, sent in September 2021. It also allowed us to identify their teaching practices: the balance between theory and practice, the tools, languages, libraries, software on which they rely on, and their pedagogical and evaluation methods. We first noted that, among the 39 participants and in the preceding 5 years (i.e. from 2016 to 2021):

- 15 of them (38.5 %) had followed between 1 and 4 courses on pedagogy, 2 (5.1 %) to more than 5 (the remainder not having followed any),
- 11 of them (28.2 %) had already attended at least one conference session on CG pedagogy (e.g. education programs at EG), the remainder not having attended any.

Figure 1 reports the obtained proportions relative to the following question: what are your teaching interests? The additional re-

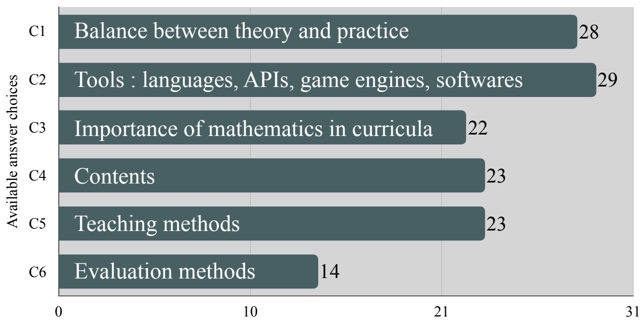

Figure 1: *What are your teaching interests? was the question asked to the French CG academic community in September 2021. Here are reported the distribution of the answers.*

ported teaching interests (by some of the 39 participants) were: generic programming of graphics cards; interactive 3D rendering on the Web; relations with companies; teaching and research.

Here is a list of the main areas of interest on which some of the 39 participants felt ready to intervene during the next educational panels (free field in the questionnaire), last 6 items (in italics) being reported as free comments:

- introduction to CG, CG practice with dedicated softwares,
- balance between theory and practice, contents to be taught,
- teaching VR/AR, ray tracing or real-time rendering,
- teaching and assessment methods,
- new course: Learning for Graphics, Graphics for Learning,
- library/framework: in physical simulation or animation,
- *sensitivity to the problem and project approaches*,
- *educational training for non-permanent staff*,
- *place of maths in CG and students' apprehensions*,
- *online tools/courses and practical work (e.g. ShaderToy for rendering, Colab for machine learning)*,
- *tool and curriculum sharing (for inspirations)*,
- *different existing French Master's degrees in CG and their attractiveness in terms of student numbers*.

To meet some of these last expectations, we have provided to the community (through the association's website [AFI]) some listings which identify all the companies, research teams, Master's degrees in France and course resources available online, in the field of CG.

This survey was also used to find panel speakers, chosen for their diverse practices and likely to fuel the debate. Our selection focused on those who had answered the questionnaire and felt ready to speak, or by consulting their list of CG teachings. At the start of each panel, the selected speakers begin with a brief personal feedback. A chairperson is chosen among the WG members, whose first task is to prepare the session with the panelists. During the session, they manage the timing of discussions and facilitate audience participation. The session is often scheduled at the beginning of the three-day conference to encourage discussions, reflections, and collaborations throughout the remaining time. Moreover, for each of the four past sessions, we chose to prevent participants from presenting remotely, to facilitate exchanges during the panel and avoid adding an additional constraint to the chairperson.

Our first panel focused on the unmissable theme: "Teaching CG, where to start? Or how to structure the introductory CG course". Since the previous survey listed many proposals around the use of tools in our teachings, the second edition focused on "software tools in CG teachings", to illustrate CG concepts and the associated pedagogy. "From provided skills to expected know-how: the case of CG" was the starting point of the third edition, bringing together both CG companies and academics in the panel. The fourth edition (in Oct. 2024) tackled the widely popular topic of "teaching mathematics for CG", intensely mentioned in the open-ended comments of the survey or at each panel final exchanges with the room.

After a general state of the art around these concerns (next section), we will detail and confront to the literature each of the themes of the four teaching panels.

3. Related work

Questions relative to teaching and curricular practices in CG started to emerge in 1983 in the form of panel during the SIGGRAPH conference [BBK*83]. The role and position of CG within future undergraduate Computer Science (CS) curricula were also debated during the Technical Symposium on Computer Science Education (SIGCSE) in 1986 [Ohl86] and 1998. It was more formally discussed at the "Year 1991 Model Curricula for Computing" [Tuc91], a joint IEEE Computer Society and ACM Task Force event (IEEE/ACM CC-1991). Its aim was to offer recommendations (in a core curriculum form) for all of the computing disciplines. In the late 90s and early 2000s and due to the substantive evolution in the discipline (availability of graphics software tools and cheaper hardware), the need for CG curriculum changes appeared simultaneously with the proposition of a new CS core curriculum [RSLD99] (during the IEEE/ACM CC-2001). At the 1999 SIGCSE panel, several members proposed a series of recommendations to modernize teaching programs [HCGW99]. The increased focus on intermediate and high-level graphical concepts, algorithms, and tools (like the 3D graphics APIs) [HS00] has been a consequence. During the SIGGRAPH 2006 educators panel, academics and industry players defined a CG knowledge-base [ALF*06] with seventeen core topics, combining both technical and artistic skills for undergraduate courses. The Bologna declaration (signed in 1999 by 29 European countries) was followed by reforms of higher education in curricula and grading systems all over the European Union. Considering its requirements, a Computer Graphics Education Workshop was held before Eurographics'06, to provide guidelines for CG teaching in European universities.

A few years later, benefiting from advances in CG and virtual reality applications, teachings made intensive use of games, computer generated imagery and animation techniques as powerful means to help students engage in the educational process [Ely12]. Observing this diversification (and based on the relevant literature in the field from 2000 to 2017), Suselo et al. [SWLR17] proposed in 2017 a categorization of the issues in teaching and learning CG, as well as approaches and methodologies to solve them. Four issues were identified: insufficient student prerequisites in mathematics and programming; difficulties with geometric concepts; difficulties in relating the encountered problem, the theoretical background and the actual solutions; student passivity in the face of an exciting but complex field. As a result, three teaching approaches and methodologies were outlined: the top-down approach (from moderately complex problems to foundations), the bottom-up approach and, finally, a hybridization of both. It's worth noting that, although the first and third approaches are the most widely described, with their own set of specific tools, it's the second that remains the most applied in practice.

4. Review/assessment of the four teaching panels (2021-2024)

The four following subsections (one per panel) describe:

- a brief overview of the questions surrounding the chosen topic,
- our precise expectations in the context of the j.FIG,
- our analyses of the impact of these panels on the community: what we observed and learned from the debates/discussions and how they influenced subsequent decisions,
- and a confrontation of our observations with the analyses of similar initiatives found in the literature.

4.1. The introductory CG course

In the mid 80s, the introductory CG course appeared at third or fourth year level within a CS major [Ohl86], to have the necessary programming and math skills. The SIGGRAPH 1994 panel has allowed for pioneering discussions and sharing between protagonists of this course (professors in arts, engineering, and computer science) [BLL*94]. The late 90s and early 2000s evolutions have led to various proposals and recommendations as a basis that people can adapt to their local needs [Wol00]. The emphasis was on the inclusion of inherently visual content and interactivity. Nowadays mixed reality seems to be increasingly favored in some introductory CG courses as a visual tool to facilitate interaction [HGS23].

Because of the interdisciplinary potential of CG, Cunningham also argued in 1999 [Cun99] and 2000 [Cun00b, Cun00a] for a wider audience (sciences, mathematics, or engineering students) to the beginning CG course. In these same years appeared the Bologna statement, which included a recommendation for European universities to have at least one CG course required for graduation. Examples of harmonization efforts in that context are discussed in [GIC08] for the specific case of the introductory CG course. In setting up a CG curriculum, the authors suggest to take into account the real constraints (institutional resources, targeted skills), objectives and profile of the targeted students rather than creating a unique one that meets all the needs at once. In the same spirit, Cunningham [Cun08] advocated in 2008 the need to adapt the teaching of the beginning course to a particular context, to attract and motivate students.

In 2017, Dodgson et al. [DC17] designed a technical introductory CG course accessible for undergraduates who had only taken one undergraduate programming course and no math courses beyond high school. The goal of this first-year course was to introduce

Proceedings published by Eurographics - The European Association for Computer Graphics.

the rigour of CG with minimal prerequisites, to attract students to the major in CG that the authors promote within their Bachelor of Science degree. In 2018 Balreira et al. proposed an update of the first survey (realized in 2000 [Wol00]) of the possible contents in the introductory CG courses [BWF18]. It lists a much higher number of topics than the previous one, due to the considerable evolution of the field over the 17 years separating them. Another difference is the way of collecting data, the first one being based on a small survey conducted during SIGGRAPH 1998 (among 20 educators from various institutions around the US), whereas the data of this newest one come from online resources (from 28 courses from around the world).

Our first panel dedicated to teaching aspects in higher education took place in Sophia Antipolis, and was focused on how to structure the introductory CG course. Starting with very narrow considerations (in the 80s), we see that there are currently a plethora of ways to introduce CG and we wanted to shine a light on several of them, chosen to highlight this diversity. The following observations, questions and comments emerged during this one-hour session:

- CG was considered by all 4 speakers as image synthesis (not as geometry processing, for example),
- 5 courses were presented, one of which concerned a Master's degree and the remaining 4 a Bachelor degree. The first is included in a second year of Master in modeling and optimization (for math students). It represents more than 30 hours. The remaining undergraduate courses take also more than 30 hours, with almost half of them for lab.
- various ways of approaching the first CG course were highlighted though these 5 curricula and two competing approaches emerged: production-ready software used as "black box" opaque tools (like Unity, Blender, WebGL, ThreeJS, Shadertoy, ...) that allow for a quick visual approach to the main theoretical concepts of CG, without requiring strong implementation skills; homemade controllable 3D engines to be enriched, that foster mastery of low level concepts and algorithms by implementing them (almost) from scratch.

On one hand, getting started to CG with a visual approach (made easier by one of the black boxes previously mentioned) provides the advantage of not having to delve too deeply into mathematical concepts. According to the target audience, it can be a good option as a way to reconcile and not discourage students with this demanding area. But as one of the speakers pointed out, "it's important to combine the use of these black boxes with a practical approach, to expand the students' creativity and/or artistic side", which generally goes hand in hand with motivation. On the other hand, homemade 3D engines with much less black box effects enforce a better understanding of fundamental concepts, provide smaller code bases with easier access, but require significant efforts to produce rich synthetic images. "Having to code mathematical concepts (traditionally taught in an abstract way) in the context of concrete applications has changed their view of mathematics", according to another speaker.

Many questions that arose focused mostly around which tool (high or low-level one), language or engine to use when choosing the second option. From there, the following divergent opinions emerged:

- "choosing Vulkan without having started by OpenGL is not a good option",
- "requiring students to use low-level platforms can be an effective way to help them understand how images are generated from scratch, thereby demystifying the underlying principles of 3D graphics",
- "grouping together several teachers to build a common tool with a certain number of primitives is better than building your own tool",
- "open-source game engine, such as Godot, or a library that progressively introduces to modern OpenGL is recommended for introductory and advanced CG courses."

One of the speakers also promoted the interest of "setting up a national programming contest which encourage the usefulness and possibilities offered by image synthesis and digital creation" (university Paris 8, annually since 2014 [PBB19]). This initiative has generated a lot of positive feedback, mainly in terms of involvement and desire to continue in CG. All speakers, when asked about how they have measured if their initial objectives were achieved (attract students, provide solid CG foundations, etc.), the majority confessed their difficulty in evaluating it. A long and costly process of quizzing at the beginning and end of the course would be necessary, to be sure to integrate the students' motivation beforehand. In accordance with the findings of Galvez et al. [GIC08] and Cunningham [Cun08] about harmonization efforts, when asked about the desirable number of hours dedicated to this course, the 4 speakers agreed that "it depends on the intended objective". They also agreed that "30 hours is too short for an introductory course, when starting from scratch in CG".

4.2. Tools

To illustrate one or more concepts of CG, various types of tools (integrated, online, remote, easy to share or not) can be thought of. In 2019, a literature review of the technologies and tools used to support teaching and learning CG [SWLR19] listed a large number of them (from APIs to modelling or custom tools), grouping them in periods of barely 10 years each, over the past 35 years. Hence 3 periods have been identified from 1991 to 2018, without any standard technology being privileged (as for other subfields of CS). Improving students' motivation and understanding of the underlying CG concepts were reported by the reviewed articles, but little convincing evidence has been provided.

For our second panel, dedicated to "software tools in CG teachings", each of the five chosen speakers was asked a 5-minute presentation of her/his tool (both accessible to everyone and as specific as possible to reach the most specialists). They also specified whether their course is intended to attract students and/or to give them a solid foundation and to justify their tool choice regarding the underlying CG learning and pedagogy. We considered both black boxes and homemade libraries. Two general production ready software were considered (for people with minimal programming experience): Unity [Uni] as a support for VR/AR (Master 2 course in CS) and the open-source Processing language and environment [Pro] for a computational geometry course (as a way to attract undergraduate students to the image field). For those two black boxes, we found papers praising their strengths for teaching the introductory CG course [PBTF09, SBG10, Los22]. Their benefits for learning fundamental graphic algorithms are: ease of quickly creating interactive graphical applications, access to lots of examples/resources (that can be easily modified or improved) and debugging tools thanks to a huge and active global community.

On the other hand, three custom tools (developed by the academics themselves) were presented:

1. coupled programming tools to attract students (gcc and OpenGL together with github for its documentation, versioning and continuous integration features; accessed by millions of users),
2. a single personal, low-level and multiplatform, open-source library (using C, OpenGL3.3+ and SDL2), designed for learning from the fundamentals that make up CPU-based real-time rendering pipeline, then sliding towards an introduction to modern OpenGL aided by facilitators, and finally concluding with an opening to the whole software architecture, covering in detail the multiple CPU-GPU interactions. The library is thus used in a series of undergraduate courses with aims ranging from introduction to computer graphics to demomaking and video game design,
3. a personal high-level and Open-Source C++ library [CGP] for creating, animating and interacting with 3D scenes. Designed to be as simple/light as possible (learning-oriented) for heterogeneous backgrounds and familiarities (from beginners to advanced) with C++. Good appropriation was observed with a feeling of "self-accomplishment" on attractive examples.

None of the above custom tools fall into the "conceptual" category (with minimal programming or math skills required, for a top-down teaching approach) that Suselo et al. [SWLR19] have listed as being the most common. One of the speakers recommends "getting students started from scratch" without using graphics APIs. Once they have understood the fundamental algorithms and associated mathematics, "they move on to shaders and/or graphics APIs". This approach, which "also requires a mastery of debugging tools and a minimum of algorithmic", seems more suited to computer science students. Others prefer mixing low and high level languages for variety, without getting used to a unique solution. A good balance might be the hybrid approach [WHS*19], to develop simultaneously practical, programming and knowledge base skills.

Of our speakers, one from those who do not encourage the exclusive use of black boxes for the introductory course (as expressed by Chen et al. in [CXR18]), believe that "whatever the choice made, what motivates teachers is to train their students' ability to adapt, on the assumption that it is difficult to predict precisely what will be useful in the years ahead". Another one argued that "assisting students or rebuilding with them the fundamental blocks (such as rasterization or the practical use of linear algebra concepts) prepares them to tackle any low-level library requiring a good knowledge of the entire chain from CPU to GPU." But as already noted by Suselo et al. [SWLR19], convincing evidences of the effectiveness of these tools relative to the desired purposes are still missing. More rigorous evaluations are needed.

Many people in our audience agreed that a motivated teacher is the main thing, while not forgetting that teaching involves a lot of repetition. Finally, regarding the difficulties of teaching these tools remotely, it seems to depend a lot on the level of autonomy and expectations. "Difficulties and discouragements mainly arise when installing certain devices or libraries, especially for those who lack IT skills".

4.3. Provided skills and employability

This one-hour session (same duration as before) took place in a slightly different context from previous years, with more time than usual devoted to industry presentations, over the three days of j.FIG 2023. The opportunity to have a rich CG industrial network in Montpellier explains this consideration. The retained topic dealing with the attractiveness and quality of the French CG Master's degrees, from the point of view of companies and recruiters (mostly from game studios), we had scheduled:

- two professors heading a Master's degree in game development (covering especially CG and Image Processing programming), with a good recruitment rate upon graduation,
- a senior engineer (who spoke on behalf of his company but also of a dozen studios he had previously questioned on the issues addressed here),
- two technical or R&D directors coming from two different studios.

Before this panel (and within the context of the j.FIG) two of the three studios had already presented in detail their activities and opportunities, the third only flew over them at the beginning of this session. Brief summaries presented by the two Master's heads highlighted the need for scientific or general culture in a wide range of fields (CS and CG fundamentals, social sciences, literature, cinema, modeling, physics, biology, optics, mechanics, etc.) with the emphasis on open-mindedness in research and soft skills such as teamwork and critical ability. For these two Master's degrees, a large space is devoted to stakeholders, particularly in the field of level design, to discuss concrete issues and highlight industry needs. The primary aim is to provide students with a solid academic foundation capable of abstraction, while companies will coach them in other aspects.

During the 30-minute debate that followed, the main questions revolved around what profiles these studios need and expect most, knowing that there is no single path that leads to game designer and developer. Since the questioned engineering-oriented studios are more focused on applied rather than fundamental research, their recruiters are mainly looking for "engineers with a fairly versatile profile and a technical background, particularly in programming, without necessarily requiring a specialization". One of the industrial speakers emphasized that "software infrastructure profiles, such as information systems administrators, are also in demand, since network performance, security issues and user support are occupying a significant position." All spoke of the importance for their colleagues of "being curious, able to adapt, speaking other languages, and knowing how to communicate and exchange ideas with a critical sense." The observation of the low feminization of the field requires an upstream awareness. And finally, when it comes to appropriating Deep Learning, the industrial speakers reported that they are "mostly cautious, but nonetheless attentive, and do not want to get started before a technological watch has been carried out, from an engineering point of view."

Proceedings published by Eurographics - The European Association for Computer Graphics.

Some of these observations concur with the diagnosis recently made by the Push Start association [Pus23]. This report focuses on the French video game industry, and took into account the vast diversity of professions and multifaceted companies involved in video games, with the aim of understanding how to structure it, make it sustainable, and internationally influential. The part that interests us the most concerns employee education in line with the needs of companies, faced with rapid developments in professions, skills and technologies in the sector. A first observation is that the recruiters have difficulty evaluating our degrees and what to expect from our students. A large disparity is observed between schools fully dedicated to video game and animation, and university curricula, which focus on scientific skills and computer graphics. In particular, Technical Artists are difficult to recruit, due to their transverse profile, and to the scarcity of appropriate initial curricula. This diagnosis proposes avenues for improvements, such as: developing new curricula and making them accessible; developing an official reference framework for jobs and skills; strengthening the links between studios and educational organizations; creating a disciplinary field of research integrating video games.

The main conclusion is that the panel was fruitful, and we would benefit from pursuing the discussions. A clear limitation is that only video game industry and animation studios were represented in our panel. Further reflection would benefit from the presence of a wider range of industrial players, and covering other areas of CG such as geometry processing, acquisition, VR/AR, offline rendering, etc.

4.4. Teaching mathematics for CG

Since 2021, the question of the place of mathematics in each CG course, and according to the target audience, has always arisen at the end of the panels. Consequently, the last teaching session, at the end of 2024, has been scheduled to be longer, *i.e.* 90 minutes, allowing time for a special guest speaker. His experience as a mathematics teacher and his interest in CG, a consequence of the opportunity to teach mathematics in CG courses at university, seemed appropriate to give him a longer time. Thus, he presented an overview of the mathematics used and the mathematical skills acquired for each of the main CG areas: rendering, shading, procedural textures, animation, geometry processing and modeling, and so on. His first observation is that "CG is catchy, involves beautiful maths, but is poorly known and would require means to attract people from high school". He also advocates teaching maths by benefiting from the graphic and visual potential of CG [Tan91], to "help make abstract concepts concrete and accessible".

Three other speakers also took part in the panel, all of them university professors, each head of a Master's degree in CS with a CG coloration. The main question was how mathematics fits into their curricula, whether as a modeling or support tool, in which courses it is needed and with what pedagogical teaching methods.

Since these three Master's degree mainly focus on computational issues, admission is mainly gained with a bachelor's degree in computer science, or more rarely in mathematics (often with planned upgrades, particularly in C++ programming). This may explain why, in practice, some students seem to be apprehensive about mathematics (sometimes even rejecting it) and some do not choose CG for fear of not having a sufficient level in maths. Most of the speakers admitted having to reassure them and/or quickly introduce video games, VR/AR, AI, medical imaging, etc. as motivation. In some places, a double bachelor degree in mathematics and computer science exists and seems to be well appreciated and well suited to pursuing in the considered Master's degrees, even if it is complicated to set up.

These Master's degree having a specialization in CG, and sometimes also in computer vision and AI, the courses related to these specializations are generally those where mathematics is most present. We know that a good background in mathematics, as well as in physics, is essential for understanding CG, but in the mentioned cases, the mathematics requirements are mainly adapted to the prerequisites acquired in the associated bachelor's degree (in the same department/university). Otherwise, the necessary updates are taught by CS teachers, with extremely practical and applied approaches (even for the evaluations). One of the speakers said that "even for mathematical aspects already learned in the bachelor's degree, students have difficulty applying them and need reminders in the targeted application contexts and even to code them, to fully understand."

According to Elyan [Ely12], even if "CS students are not expected to be mathematicians, it is necessarily for them to develop good understanding of the underlying mathematical concepts". Already in 2001, Tucker et al. [TKB01] pointed out that CS curricula "were drifting away from a fundamental commitment to theoretical and mathematical ideas", both in introductory and core courses. He said this causes a disconnect between theory and practice for a vast majority of CS graduates (intended to become software engineers/developers and considered as experts in techniques and artifacts that reflects current technologies). By initiating a dialogue among educators, the authors suggested to find a way to restore the mathematical cohesion to the undergraduate CS curriculum, considered at that time as "(one of) the least mathematical among the science and engineering disciplines".

Some of the interviewed teachers believe that "projects with a well-defined problem to solve, i.e. where the mathematics can be hidden, are a good way to motivate students to tackle CG problems, while reassuring them about their mathematical skills". "Through programming, which they generally prefer to mathematics, they are forced to start thinking, identifying and understanding how to apply the concepts they have seen previously". Video or educational games are identified as good examples to improve student's engagement and interaction [MGBMO*08]. This practical approach (or at least a presentation of key applications) can also be considered when introducing mathematical concepts in CG courses. Intuitive, practical and non-mathematical approaches, with simple and short activities using an appropriate software tool, were also investigated by Elyan [Ely12] in 2012. His wish was to teach students challenging mathematical concepts like B-Splines, Bezier Curves, FFD, hierarchical modeling, and inverse and forward kinematics, for an interactive 3D animation course. He noticed that the engagement and thinking students put into these hands-on activities helps them better understand some of the most difficult mathematical theories.

Finally, what seems important is to offer an introduction to CG at the end of the Bachelor's degree, so that students can discover it through practice, while at the same time providing the necessary mathematics, and see if they are interested before continuing. We indeed know the importance of having motivated students, in one of the most demanding areas of CS.

5. Observations and feedback

As expected, panels resulted in a lot of discussion and exchanges between speakers and with the audience too. This experience revealed significant disparities in the teaching offer in France, due to the number of possible approaches, techniques and surrounding industrial network. In 2022, within the AFIG teaching WG, we have started to share a pool of educational resources already available online, as a community github page linking courses, materials and codes. The main goal is to capitalize on the amount of work required to create such software contexts, adapted to our learning, in order to create a community of practices. This base could serve as a guide to enrich existing courses or create new ones. In addition, it would be interesting to keep up-to-date the prerequisites related to the skills sought after graduation, as this would be as useful for companies as it would be for colleagues looking for Ph.D. candidates.

The feedback from these four panels highlighted the fact that many innovations have been implemented, with the aim of motivating students on sometimes demanding topics. Nevertheless, what emerges is a common desire to establish the basic principles of CG, often associated with sufficient mathematical foundations, while using tools to make a large place for practice. Thus, with the idea that the impact on student motivation is always at the heart of our concerns, we have identified three major points of attention at disciplinary level:

- the discovery and immersion of graphic culture in a fun way, using software such as 3D modelers (Blender) or game engines (Unity),
- the use of libraries dedicated or specially designed by teachers, the latter being either low-level oriented, for an algorithmic understanding linked to the most complete graphic approaches possible, or much more specific and high-level, in order to focus learning on a particular point,
- the use of the possibilities offered by the visualization of various scientific phenomena as a motivation for programming and learning mathematics. In our opinion, there are still projects to be undertaken in terms of teaching tools and methods related to assessment processes which are not sufficiently shared.

6. Conclusion and perspectives

We presented the way in which we have organized panels at j.FIG over the past four years and the feedback we have received from the French CG community. Each educational theme and the corresponding reactions were compared with other similar initiatives found in the literature. We would like to point out that in many cases, the debates elicited similar reactions, comments and conclusions.

We note that animation, in the form of a panel, has brought various opportunities, like:

- opening discussions between associate professors, professors (and also temporary teachers) and industrial actors among the French community,
- it's a way for everyone attending these sessions each year, to keep focused on the pedagogical part of our job and even to discover things that each of us had not necessarily thought of,
- and everyone comes out a little from the solitude of the teacher (alone in front of the class).

It will certainly be reproduced in the years to come. We are thinking about other levers to promote more in-depth exchanges, in term of partnerships and collaborations. Surveys could also be submitted to the panel participants, in order to support each panel's findings, enrich the data and improve the analysis of community practices.

Brainstorming ideas and new themes that arouse interest are our main objectives, together with evidence to know if these exchanges of ideas and practices lead or not to real concrete actions. At the end of the 2022 panel, a new list of suggested topics and concerns had emerged among: green IT, asynchronous teaching, video games, 3D project types, low level vs. high level. Questions of this kind have also arisen: are fundamental algorithms (Bresenham, scan line, etc.) still relevant?, at what level should we introduce CG?, what algorithms should we suggest to beginners (first year)?

In terms of educational innovations or strategies in connection with CG, we could think of:

- the Problem Based Learning approaches [MGJ06] to improve the student's learning capability or focusing on learning the necessary tools to correctly find a solution to concrete problems,
- wondering which educational online environments or tools to consider for a remote attendance or an hybrid mode, knowing that CG is nowadays mainly taught in a hands-on manner [WLD*21, RMC*21],
- flipped classrooms and their impacts in teaching CG. In 2021, Indu made an initial assessment in Arts and Science Colleges [Ind21], but what about other contexts?

More recently, questions around AI for teaching and learning CG arised strong interest and could be the main theme of a future panel. At least, the students' use of generative AI as a programming assistant (already explored by extensive research) should be investigated, emphasizing that there remains little consensus on implementing curricula and assessment changes [FLRWD25]. It would finally be necessary to question their integration regarding the past, present, and future of computing education research as a field of science [ALPS23].

References

[AFI] AFIG:. `https://www.asso-afig.fr`. 1, 2

[ALF*06] ALLEY T., LAXER C., FLAXMAN T., GEIGEL J., GOLD S., HITCHNER L., ORR G., POLLACK B. W.: A knowledge base for the emerging discipline of computer graphics. In *ACM SIGGRAPH Educators Program* (2006), Association for Computing Machinery, p. 32–es. 3

[ALPS23] APIOLA M., LÓPEZ-PERNAS S., SAQR M. (Eds.): *Past, Present and Future of Computing Education Research: A Global Perspective*. Springer, 2023. 7

[BBK*83] BORK A., BROWN M., KING R., DAM A. V., WOZNY M., FOLEY J. D.: Computer graphics in higher education (Panel Session). *ACM SIGGRAPH 17*, 3 (1983), 31–33. 3

[BLL*94] BRESENHAM J., LAXER C., LANSDOWN J., OWEN G. S., LARRONDO-PETRIE M. M.: Approaches to teaching introductory computer graphics. In *ACM SIGGRAPH* (1994), Association for Computing Machinery, pp. 479–480. 3

[BWF18] BALREIRA D., WALTER M., FELLNER D.: A Survey of the Contents in Introductory Computer Graphics Courses. *Computers & Graphics 77* (2018), 88–96. 4

[CGP] CGP:. https://imagecomputing.net/cgp. 5

[Cun99] CUNNINGHAM S.: GVE: Report of the Eurographics/SIGGRAPH workshop on graphics and visualization education. *ACM SIGGRAPH 33*, 4 (1999), 96–102. 3

[Cun00a] CUNNINGHAM S.: Powers of 10: the case for changing the first course in computer graphics. In *SIGCSE: ACM Technical Symposium on Computer Science Education* (2000), Association for Computing Machinery, pp. 46–49. 3

[Cun00b] CUNNINGHAM S.: Re-inventing the introductory computer graphics course: providing tools for a wider audience. *Computers & Graphics 24*, 2 (2000), 293–296. 3

[Cun08] CUNNINGHAM S.: Computer graphics in context: an approach to a first course in computer graphics. In *ACM SIGGRAPH ASIA 2008 Educators Programme* (2008), Association for Computing Machinery. 3, 4

[CXR18] CHEN M., XU Z., RIPPIN W.: On the pedagogy of teaching introductory computer graphics without rendering APIs. In *Eurographics Education Papers* (2018), The Eurographics Association, pp. 47–50. 5

[DC17] DODGSON N. A., CHALMERS A.: Designing a Computer Graphics Course for First Year Undergraduates. In *Eurographics Education Papers* (2017), The Eurographics Association, pp. 9–15. 3

[Ely12] ELYAN E.: Enhanced interactivity and engagement: Learning by doing to simplify mathematical concepts in computer graphics and animation. *EDUCON: IEEE Global Engineering Education Conference* (2012), 1–8. 3, 6

[FLRWD25] FENG T. H., LUXTON-REILLY A., WÜNSCHE B. C., DENNY P.: From Automation to Cognition: Redefining the Roles of Educators and Generative AI in Computing Education. In *ACE: Australasian Computing Education Conference* (2025). 7

[GIC08] GÁLVEZ A., IGLESIAS A., CORCUERA P.: An Introductory Computer Graphics Course in the Context of the European Space of Higher Education: A Curricular Approach. In *ICCS: International Conference on Computational Science* (2008), Springer Berlin Heidelberg, pp. 715–724. 3, 4

[HCGW99] HITCHNER L., CUNNINGHAM S., GRISSOM S., WOLFE R.: Computer graphics: the introductory course grows up. *SIGCSE: ACM Technical Symposium on Computer Science Education 31*, 1 (1999), 341–342. 3

[HGS23] HEINEMANN B., GÖRZEN S., SCHROEDER U.: Teaching the basics of computer graphics in virtual reality. *Computers & Graphics 112* (2023), 1–12. 3

[HS00] HITCHNER L., SOWIZRAL H.: Adapting computer graphics curricula to changes in graphics. *Computers & Graphics 24* (2000), 283–288. 3

[Ind21] INDU A.: Impact of Flipped Classroom in teaching Computer Graphics. *Psychology (Savannah, Ga.) 58* (2021), 9888–9895. 7

[Los22] LOSCOS C.: Introduction to computer graphics: a visual interactive approach. In *Eurographics Education Papers* (2022), The Eurographics Association, pp. 1–8. 5

[MGBMO*08] MORENO GER P., BURGOS D., MARTINEZ-ORTIZ I., SIERRA J., FERNÁNDEZ-MANJÓN B.: Educational game design for online education. *Computers in Human Behavior 24* (2008), 2530–2540. 6

[MGJ06] MARTÍ E., GIL D., JULIÀ C.: A PBL experience in the teaching of computer graphics. *Computer Graphics Forum 25* (2006), 95–103. 7

[Ohl86] OHLSON M. R.: The role and position of graphics in computer science education. In *SIGCSE: ACM Technical Symposium on Computer Science Education* (1986), Association for Computing Machinery, pp. 232–237. 3

[PBB19] PALUS J.-P., BELHADJ F., BOURDIN J.-J.: Do contests improve students skills in Computer Graphics? The case of API8. In *Eurographics Education Papers* (2019), The Eurographics Association, pp. 45–51. 4

[PBTF09] PELLICER J. L., BLANES J. S., TORMOS P. M., FRAU D. C.: Using processing.org in an Introductory Computer Graphics Course. In *Eurographics Education Papers* (2009), The Eurographics Association, pp. 23–28. 5

[Pro] PROCESSING:. https://processing.org/. 4

[Pus23] PUSH START ASSOCIATION: *Diagnostic Ambition Jeu Vidéo - L'industrie du Jeu Vidéo en France*. Tech. rep., National Consortium bringing together the 8 regional professional associations of the video game sector and the national video game union, 2023. 6

[RMC*21] RODRIGUES R., MATOS T., CARVALHO A., BARBOSA J., ASSAF R., NÓBREGA R., COELHO A., SOUSA A.: Computer Graphics teaching challenges: Guidelines for balancing depth, complexity and mentoring in a confinement context. *Graphics and Visual Computing 4* (2021), 200021. 7

[RSLD99] ROBERTS E., SHACKELFORD R., LEBLANC R., DENNING P. J.: Curriculum 2001: interim report from the ACM/IEEE-CS task force. *SIGCSE Bull. 31*, 1 (1999), 343–344. 3

[SBG10] SCHWEITZER D., BOLENG J., GRAHAM P.: Teaching introductory computer graphics with the processing language. *Journal of Computing Sciences in Colleges 26* (2010), 73–79. 5

[SWLR17] SUSELO T., WÜNSCHE B., LUXTON-REILLY A.: The Journey to Improve Teaching Computer Graphics: A Systematic Review. In *ICCE: International Conference on Computers in Education* (2017), APSCE, Christchurch, New Zealand, pp. 361–366. 3

[SWLR19] SUSELO T., WÜNSCHE B. C., LUXTON-REILLY A.: Technologies and Tools to Support Teaching and Learning Computer Graphics: A Literature Review. In *ACE: Australasian Computing Education Conference* (2019), Association for Computing Machinery, pp. 96–105. 4, 5

[Tan91] TANIMOTO S. L.: Mathematics Education Using Computer Graphics (panel summary). *ACM SIGGRAPH 25*, 3 (1991), 165–167. 6

[TKB01] TUCKER A. B., KELEMEN C. F., BRUCE K. B.: Our curriculum has become math-phobic! In *SIGCSE: ACM Technical Symposium on Computer Science Education* (2001), Association for Computing Machinery, pp. 243–247. 6

[Tuc91] TUCKER A. B.: Computing Curricula 1991. *ACM/IEEE-CS Joint Curriculum Bull. 34*, 6 (1991), 68–84. 3

[Uni] UNITY:. https://unity.com. 4

[WHS*19] WÜNSCHE B. C., HUANG E., SHAW L., SUSELO T., LEUNG K.-C., DIMALEN D., VAN DER MARK W., LUXTON-REILLY A., LOBB R.: CodeRunnerGL - An Interactive Web-Based Tool for Computer Graphics Teaching and Assessment. In *ICEIC: International Conference on Electronics, Information, and Communication* (2019), pp. 1–7. 5

[WLD*21] WÜNSCHE B. C., LEUNG K., DIMALEN D., VAN DER MARK W., SUSELO T., ALEX M., SHAW A., LUXTON-REILLY A., LOBB R.: Using an Assessment Tool to Create Sandboxes for Computer Graphics Teaching in an Online Environment. In *CSERC: Computer Science Education Research Conference* (2021), ACM, pp. 21–30. 7

[Wol00] WOLFE R. J.: Bringing the introductory computer graphics course into the 21st century. *Computers & Graphics 24* (2000), 151–155. 3, 4