In at the Deep End: An Activity-Led Introduction to Creative Computing with Interactive Computer Graphics

Eike Falk Anderson[1] and Christopher E. Peters[1] and Fotis Liarokapis[1] and John Halloran[1]

[1]Interactive Worlds Applied Research Group, Coventry University, United Kingdom

Abstract

Misconceptions about the nature of the computing discipline(s) pose a serious problem to faculties that offer computing degrees, as students enrolling on their programmes come to realise that their expectations are not met by reality. This frequently results in the students' early disengagement from the subject of their degrees which in turn can lead to excessive 'wastage', i.e. reduced retention. In this paper we report on our academic group's attempts within creative computing degrees at a UK university to counter these problems through the introduction of a six week long project that newly enrolled students embark on at the very beginning of their studies. This group project provides a breadth-first, activity-led introduction to their chosen academic discipline, aiming to increase student engagement while providing a stimulating learning experience with the overall goal to increase retention. Having run in two iterations, we believe that this approach has been successful, with students showing increased interest in their chosen discipline and noticable improvements in retention following the first year of the students' studies.

1. Introduction

When applying for a university degree in computing, few potential students have an accurate conception of what their chosen degree entails. Many believe computing or computer science to be an extension of the use of office suites that they are familiar with from taking ICT (information and communication technologies) courses at school, confusing the degree programmes with basic computer literacy [BM05]. As a result, many students are disappointed when they enrol at university and – to their dismay – discover their mistake. This is reflected in the observable decline of retention in computing programmes, and to remedy this, it has been suggested to modify degrees to become "more fun" and to offer "multidisciplinary and cross-disciplinary programs" [Car06] that will keep students interested in the subject. Unfortunately, retention problems are not restricted to traditional computing courses, but also extend to some of the multidisciplinary and cross-disciplinary degree programmes, such as creative computing degrees. In these, one can find a completely new set of misconceptions and potential students often confuse programmes such as multimedia computing with more vocational training courses for content creation software packages and web design.

Furthermore, the complexity and the resulting difficulty of undergraduate computing degree programmes tends to be greatly underestimated, and once students become aware of this, they often disengage from the subject matter. This usually results in failure, or in some cases the students' withdrawal from their degree programmes. Consequently, retaining computing students remains a serious problem, one possible solution for which is to deepen the student's engagement with the subject.

In the wake of the adoption of a new pedagogic model by our faculty, located in Coventry University (UK), the solution of the Creative Computing subject group to address this problem has been the development of an integrative, interdisciplinary learning experience, providing new students with a breadth-first introduction to their chosen academic discipline. Newly enrolled students embark on a subjectspanning group project that encompasses the first six weeks of their first year at university, replacing the regular teaching schedule and combining various aspects of the courses that make up the first year of the creative computing degree programmes. This project, which is not formally assessed, not only engages the students closely with the subject matter of their degree programme, but as a side effect also increases cohort cohesion. First piloted at the start of the 2009/2010 academic year [SEA*10], the activity for our creative computing degrees – including Multimedia Computing and Games Technology pathways – integrates software and hardware development with usability evaluation, viral marketing techniques and academic writing. In its refined second iteration at the onset of the 2010/2011 academic year, the software development aspect focussed on computer graphics, resulting in the students' creation of a computer graphics application with a physical hardware interface. Our creative computing degree programmes are heavily reliant on modern multimedia concepts and technologies. "Multimedia – while embracing computer graphics – describes the foray of other disciplines into the digital realm" [Gon00] and through their projects our students not only "learn computer graphics", but also "learn through computer graphics", effectively making our students' learning experience a hybrid of both aspects of teaching computer graphics in context [CC09].

In this paper we first present our approach and the reasons for choosing this, followed by a discussion of the tasks that the students completed as part of their project, concluding with insights gained from this teaching 'experiment'.

2. Rationale

The Creative Computing subject group in the Department for Computing and the Digital Environment (CDE) of Coventry University's Faculty of Engineering and Computing (EC) has adopted Activity-Led Learning (ALL) [WM08, IJP*08, PJB*10] as the pedagogic model for improving student engagement and retention. In the belief that if challenged with an interesting problem the students will rise to the occasion, we confront the students with an ambitious task requiring them to take on a proactive role in problem-solving and to use their own initiative if they want their 'product' to succeed. The activity-led approach has its roots in problem-based learning (PBL). PBL has gained some acceptance as an effective approach within a variety of disciplines in higher education environments [YG96, Fel96]. This comes as no surprise, as according to Perelman [Per92] it satisfies three important criteria that promote optimal learning:

- It provides an environment where the student is immersed.
- Students receive guidance and support from fellow students.
- The learning process is functional.

This type of student-centred education is not without problems, however, and has been severely criticised [KSC06], and the success of PBL approaches may depend upon the ability of students to work together to identify and analyse problems and/or to generate solutions [Cam96]. However, learning is multidirectional, including other students in mutually supporting roles as well as tutors and faculty [AM93].

Computing curriculum recommendations state that "the breadth of the discipline should be taught early in the curriculum" [Tuc96]. This is realised in a breadth-first computing curriculum, where students are exposed to the computing domain through a "broad introduction to the major areas of Computer Science" [VW00], allowing students to gain a more comprehensive understanding and appreciation of the discipline. They gain "a holistic view of a topic before they learn about more complicated details" [DG06] that empowers them. Important concepts are touched upon early on to provide students with the basis for a much larger range of activities than would be possible in more traditional/conservative teaching sequences. This is because students experience the tasks that they embark upon in the wider context of the computing discipline, rather than as isolated subject matter. While to many students this may seem intimidating at first, it nevertheless tends to result in much deeper understanding.

ALL or PBL not only lend themselves to the teaching of computer graphics [MGJ06], but the use of computer graphics itself offers the possibility of defining interesting problem-based learning scenarios while also enabling collaborative or mediated learning activities that could lead to better learning [Tud92]. Learning occurs through multiple interactions within the learning environment [SD95, Cam96] and thus a potential added benefit of using computer graphics in combination with PBL scenarios is that learners engage with these using different senses, helping them to fully immerse in the learning situation [Csi90] which should result in learning gains [CGSG04].

2.1. Finding a Challenge

To meet our goal of engaging the students with the creative computing discipline we had to face our own challenge of finding a suitable activity for the students to embark on. In the development of such activities it is important that they "are meaningful to the student" [Cun99], as well as appropriate for the intended student group, which in our case are absolute beginners who are taking their first steps in higher education. The activities designed for the six week group project would have to be related to the degree programmes of the students, complex enough to appear challenging, yet achievable within the set timeframe. After some consideration, we settled on the idea of tasking the students with the development of a graphics application based on the popular *Etch A Sketch*® drawing toy by the Ohio Art Company (http://www.etch-a-sketch.com), the computer implementation of which would not only involve graphics, but would also provide an interesting exercise in user interface design and evaluation [Bux86]. To provide students with an additional challenge we then extended the basic concept of a 2D drawing toy with turnable knobs as inputs for the two drawing axes into the third dimension.

The development environment chosen for the task was *Processing* [RF06] (`http://www.processing.org`), a Java-derivative language for computer arts creation, which lends itself well to introductory programming and computer graphics education [PBTF09] and also interfaces with the Arduino micro-controller [Sto09] (`http://www.arduino.cc/`) that we chose for the development of the hardware interface. The Arduino is an Open Hardware design that has been successfully employed as an educational tool [FW10], which allows the easy creation of input devices for computers, which in the case of the kits we used does not require any soldering but allows the hardware to be simply slotted together, making it ideal for our purposes.

3. Learning by Doing

For this group project the cohort of 54 students was split up into groups of 6 to 7 students each. The project, dubbed the "six week challenge" of creating a 3D etch-a-sketch-like graphics application with a dedicated hardware interface was broken down into six sub-challenges – or themes – that each added new elements to the overall project and that each could be completed within one week, including:

- 2D graphics programming
- 3D graphics programming
- hardware design & interfacing
- usability evaluation
- viral marketing
- academic communication & reporting

We employed an activity-led instruction cycle [AP09] (Figure 1) in which students were first – usually at the start of each week – introduced to the sub-challenges, with important subject related information covered in a short introductory lecture. Students were then left to work out how to solve each of the sub-challenges. Teaching staff were available throughout to provide encouragement and additional guidance when requested and, depending on the student groups' progress, to run additional sessions to cover subject areas that the students discovered while working on their projects.

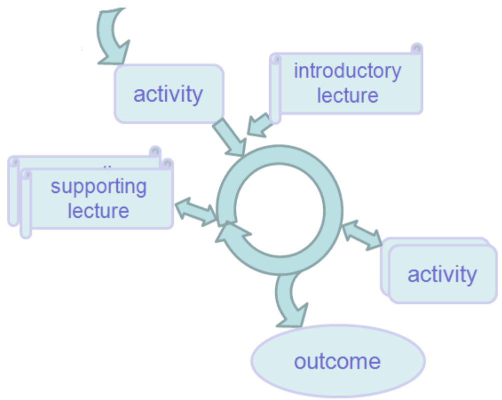

Figure 1: *The Activity-Led Instruction cycle. Starting from an initial* activity*, students receive an introductory lecture to subject-specific aspects of their task, which they then solve independently; additional lectures are based on students' needs/demands and may lead to further* activites.

3.1. Making Pictures

The first set of tasks for the student groups concentrated on the development of the graphics application. This required from each team the writing of code in the – new to them – *Processing* language, the creation of graphical elements for providing a user interface with their program and dealing with 2D and 3D geometry. Initially, the basic functionality of a 2D drawing tool was implemented that would function with basic keyboard interaction, after which the 2D application was extended into a 3D version. The actual implementation details of the keyboard controls were decided by the student groups. However, the minimum implementation specifications that were required from the groups included the following features:

- Implementation of a drawing environment (initially using a square and possibly different 2D shapes).
- The ability to performing some basic affine transformations in 2D and 3D (i.e. translation, scaling and rotation).
- Definition of a changeable camera/view.
- User interaction based on predefined keyboard keys and/or mouse control (i.e. controls that allow a user to limit movement to X, Y and Z).
- An appropriate graphical user interface design.

All of the student groups achieved at least a basic implementation of the features and demonstrated prototypes capable of drawing to the screen in 2D and 3D, and allowing the screen to be cleared afterwards. Most groups exceeded the basic requirements (see for example, Figure 2) and included additional features. Many of these related to the selection of different drawing colours from a predefined palette, either by manual selection or, in some cases, automatic schemes that accounted for the drawing depth by changing some of the colour characteristics. A number of implementations also featured the use of 2D shapes as brushes with which to draw. As students experimented with shapes and drawing in 3D, important questions arose. For example, technical issues relating to camera set-up, object and scene rotation, and 3D object positioning using transformations, all arose naturally as the task was feature-driven. In this way, students discovered for themselves the need to understand these concepts, which might otherwise have seemed obscure or unimportant.

3.1.1. Control

Students were also encouraged to investigate different interaction schemes, for example, by mapping different keys onto controls and considering mouse movement. In particular, they were tasked with attempting to control the application using the minimum number of keys possible and to create novel mouse-keyboard methods for control. This added an extra challenge beyond the more obvious 1:1 mapping between keys and functions, and additionally helped raise important issues for consideration during the modelling of the physical controller (see Section 3.2) and in the usability evaluation (see Section 3.3). A number of groups succeeded in enabling more advanced interaction control by combining both the mouse and the keyboard. This proved to be very useful when the groups were subsequently asked to design user interaction tasks, for example drawing a 3D cube, for usability studies.

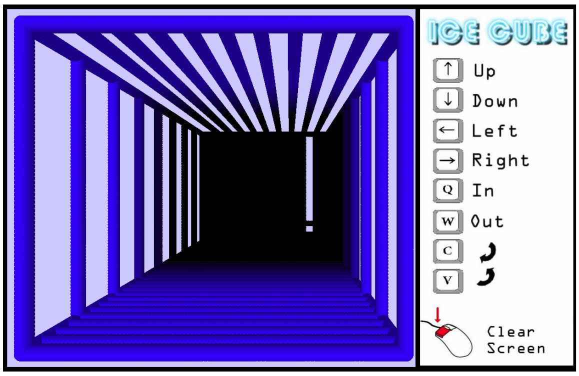

Figure 2: *One of the student-created 3D drawing applications.*

3.2. Getting Physical

Once the basic graphics application was developed, teams were asked to integrate their application with a dedicated hardware interface and then to evaluate the hardware prototypes. For the hardware task, students used the Arduino prototyping platform, which allows users to quickly construct devices ranging from simple flashing lights to autonomous spyplanes and handheld consoles. Online resources were provided to the student groups, including ebooks and hardware tutorials. In addition, students were given instructions on how to create a blinking LED device using resistors and potentiometers. Resistors were used to protect the circuit and the potentiometer to control 'knight rider' (LED Scanning light effect) speed. At the end of the task, all of the groups had created circuit diagrams for their etch-a-sketch-like applications and many students created sulutions with three pontentiometers for controlling the drawing, similar to the "Digital Airbrush" by Batagelj et al. [BMTM09]. Some of the most important keyboard functions that were assigned to hardware buttons included: change of colour, drawing speed change, background colour change, clearing the screen, restoring the screen, precision mode, camera movement, zoom in/out, and the provision of a help screen. Some groups also decided to provide a combination of two or more button pushes to perform a particular action, solving the problem of having too many key assigned features.

3.3. Usability Evaluation

The usability component of the six week challenge involved students in designing a simple usability study focussing on one or two key tasks for their etch-a-sketch, running the study on four or five users, collecting data, and analysing it to develop an informed view on whether the interface to their graphics application was usable in terms of the tasks tested.

The main difficulties in teaching usability usually are that it is highly conceptual and often abstract. Typically it is taught by asking students to run studies on interfaces they may not have a personal interest in. The six week challenge meant that students had a strong motivation to show their designs were usable. Personal investment in the work helped leverage engagement in many issues which can be a challenge to teach, in particular the forming of a research question for a usability study, the collection and analysis of different types of data, realistic and relevant scoping of user tasks, and the correct setting up and running of user sessions.

The embedding of advanced usability material within the six week challenge increased its accessibility: there was impressive work within a short period. Our activity-led approach in general can be claimed to ease the transition from pre-degree to degree education, particularly helping to ameliorate the feelings of dismay and difficulty we identified in the introduction (Section 1).

3.4. Dissemination

An important further aim beyond developing students' technical abilities and team work was to develop their awareness of the importance of dissemination and how dissemination should be tailored to both target audience and goal. Students disseminated their work both internally and externally through group demonstrations, a viral marketing campaign and academic communication methods. The aims of dissemination were to inform the work of other groups, to provide them with the experience of presenting work to different external target-groups (e.g. academics, consumers), and to think about ways in which quantitative and qualitative feedback could be collected.

3.4.1. Group Demonstrations

At the end of every week, a special ‘show and tell’ session consisting of a gathering of all of the students and lecturers involved in the six week challenge was organised so that students could demonstrate their work to the whole cohort, as well as to members of the faculty. This was primarily a student-driven activity: while lecturers had the opportunity to provide feedback on the work of the students, the student demonstration sessions focused on students commenting on the work of others. Most importantly, it allowed groups to demonstrate any innovative features that they had implemented over the course of the previous week. We believe that the fostering of this type of constructive competition between groups was a major contributing factor in motivating them to seek new and interesting features to be demonstrated the following week.

In addition to internal demonstration, students also had to disseminate their work externally in two contrasting ways: through a viral marketing campaign (Section 3.4.2) and in the form of an academic manuscript (Section 3.4.3).

3.4.2. Viral Marketing Campaign

A challenge based on a viral marketing scenario was chosen as it involved a number of concepts that first year students could relate to, particularly the use of social networking and video sharing sites, while also serving to highlight the necessity of differing dissemination methods based on the target audience, e.g. for comparison with academic communication methods (Section 3.4.3). Students published their work by placing interactive demonstrations of their graphics applications on their web-pages (Figure 3) and loading prerecorded videos on YouTube. The Processing system provided the necessary facilities for allowing students to do this themselves, as it allows interactive graphics programs to be embedded in websites. Most of the student groups successfully completed the website integration of their applications and interfaces, while some groups chose to provide downloadable executables of their applications instead. The graphical nature of the work seemed particularly amenable to such sites, as a means for attracting interest from peers and potential employers, and also serving as a starting point for the creation of a graphics programming portfolio. It also served to raise the question of feedback, by looking at ways in which qualitative and quantitative data could be collected. This involved accounting for simple metrics, such as tracking the number and types of comments and views that their work attracted. The issue of feedback is sometimes underestimated from the students’ point of view. Graphics work published on the web may be a very useful way for attracting comments from more skilled graphics practitioners from around the world, as a way for students to obtain broader formative feedback on their portfolio work from a diverse audience. The viral marketing campaign helped raise an awareness that the dissemination method must account for the target audience, which may also include potential employers.

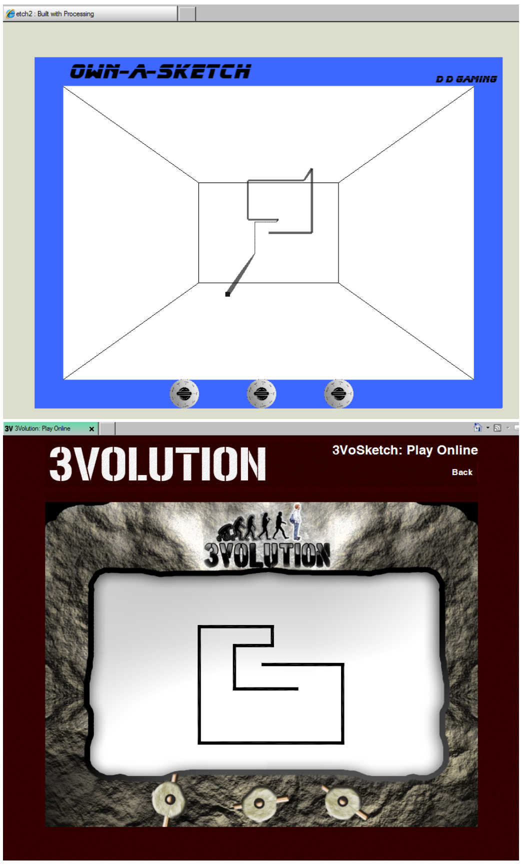

Figure 3: *Examples of student groups’ graphics applications embedded in the groups’ websites.*

3.4.3. Academic Communication

In contrast to the viral marketing campaign, students also had to consider how they should disseminate their work in an academic manner. The task for the final week of the project was to prepare a short paper (3-4 pages of collaborative academic writing) providing background information to their projects and stressing the relevance of this research to their product. Each student group were set a specific research question – many of which were related to the graphics techniques used – that they were asked to answer with their short paper. For this the groups had to:

- engage with a number of academic texts, providing them with a basic understanding of academic writing (language and style), some of which [LR88, Lar09], originating from the computer graphics community, were provided to them;
- adopt appropriate strategies for finding and evaluating relevant textual sources [Gri09], including the use of citation databases;
- learn to organise information in a logical manner, suitable

for presentation in written form, as well as for oral presentation [Ger04].

After an introduction to literature search strategies in the introductory lecture, students were also introduced to the LaTeX document preparation system [Lam86] to ease them into the practice of preparing consistently formatted documents. Students were then directed towards the compilation of a comprehensive reading list of academic articles that appeared relevant to their set research questions, providing the basis for their short review/survey paper. Throughout this activity, students were repeatedly briefed on the principles of academic honesty to prevent problems like plagiarism.

The resulting short papers showed an unexpected level of maturity, rarely seen in students in their first year at university. The students also developed a much greater appreciation for the academic writing style, contrasting it to the much more informal communication forms they were familiar with before (see Section 3.4.2).

4. Discussion

Our student-centered, activity-led introduction to creative computing through the development of a simple, yet intriguing interactive computer graphics application, certainly appears to have been a success. The inclusion of different modes of communication, and the inclusion of 'academic writing' in particular seems to have been especially beneficial and assessments submitted by the students so far appear to be of a better quality than previously observed. Nevertheless, while our approach seems to have improved retention within the first student cohort experiencing this educational experiment, we believe it is too early to generalise our findings, which at this stage are primarily anecdotal. For example, one factor that may have influenced this apparent success may be the early use of group-based activities, which can provide a social support structure that helps to retain students who might otherwise consider leaving the course.

Our mode of delivery has very much followed the concept of activity-led instruction, which in this context refers to the instruction of students on how to embrace the activity-led learning process. At the introduction for every sub-challenge (see Section 3), exemplar-based activity sessions were organised with the primary purpose of familiarising students with the process, rather than the task's content *per se*. Students were thus provided with a concrete, real-world example of the processes involved in addressing the challenges, eventually turning them into pro-active problem solvers who were not 'afraid' to face new problem domains. In this respect the weekly 'show and tell' sessions were also highly useful, as the competition they instilled between the different student groups prompted many students to independently investigate different techniques, which they then disseminated among their peers – effectively students themselves took on the role of instructors.

One important observation that should be taken seriously is that despite our approach's expectation that the students should demonstrate initiative and solve the set challenges on their own, this does not imply reduced responsibility or workload on behalf of faculty involved in preparing the challenges and developing teaching materials for the sessions that are led by an instructor. Furthermore, as the student groups have freedom in the way in which they approach any task, their solution may very well miss a specific aspect of vital importance to the outcome of their activities. Instructors must watch for these 'wrong turns' and if the need should arise, make the students aware of potential problems with their chosen approach.

Finally, the last thing left for us to do is to come up with another 'exciting' six week challenge for next year's cohort, which given the success that we believe to have achieved, is not an easy task.

5. Acknowledgments

The authors would like to thank the other members of the Creative Computing subject group, Peter Every and James Shuttleworth who defined our "six week challenge" and developed much of the content, as well as our group's teaching assistants Michael Richards and Richard Lane. Without their hard work the student project described in this paper would not have been possible. Furthermore we need to thank our students who achieved much more than we had anticipated.

The screenshots shown in this paper show the projects of three of the student groups. Figure 2 shows the program developed by the student group *Clumsy Penguin Entertainment* (Sareena Hussain, Sennel Ionus, Charnjeet Kaur, Jaipreet Panesar, Shaun Richardson, Anthony Rickhuss and Sarah Wardle). Figure 3 shows programs developed by groups *DDG* (Ahsan Ahmed, Sean Bhadrinath, William Brady, Thomas Bridger, Constantin Cercel and Ian Evans) and *3volution* (Tom Ellerker, Majura Etuttu, Matt French, Richard Oughton, Kev Panchal and Alexa Sirbu).

References

[AM93] ALBANESE M., MITCHELL S.: Problem based learning: A review of literature on its outcomes and implementation issues. *Acad Med 68*, 1 (1993), 52–81. 2

[AP09] ANDERSON E. F., PETERS C. E.: On the provision of a comprehensive computer graphics education in the context of computer games: An activity-led instruction approach. In *Eurographics 2009 - Education Papers* (2009), Domik G., Scateni R., (Eds.), Eurographics Association, pp. 7–14. 3

[BM05] BEAUBOUEF T., MASON J.: Why the high attrition rate for computer science students: some thoughts and observations. *ACM SIGCSE Bulletin 37*, 2 (2005), 103–106. 1

[BMTM09] BATAGELJ B., MAROVT J., TROHA M., MAHNIC D.: Digital airbrush. In *Proceedings of the 51st International Symposium ELMAR-2009* (2009), pp. 305–308. 4

[Bux86] BUXTON W.: There's more to interaction than meets the eye: Some issues in manual input. In *User Centered System Design: New Perspectives on Human-Computer Interaction*, Norman D., Draper S., (Eds.). awrence Erlbaum Associates, 1986, pp. 319–337. 3

[Cam96] CAMP G.: Problem-based learning: A paradigm shift or a passing fad? *Med Educ Online 1*, 2 (1996). 2

[Car06] CARTER L.: Why students with an apparent aptitude for computer science don't choose to major in computer science. *ACM SIGCSE Bulletin 38*, 1 (2006), 27–31. 1

[CC09] CASE C., CUNNINGHAM S.: Teaching computer graphics in context - cge 09 workshop report. Available from: http://education.siggraph.org/conferences/eurographics/eurographics-2009-computer-graphics-education-09-workshop/teaching-computer-graphics-in-context/, 2009. 2

[CGSG04] CRAIG S., GRAESSER A., SULLINS J., GHOLSON B.: Affect and learning: An exploratory look into the role of affect in learning. *Journal of Educational Media 29* (2004), 241–250. 2

[Csi90] CSIKSZENTMIHALYI M.: *Flow: The Psychology of Optimal Experience*. Harper and Row, 1990. 2

[Cun99] CUNNINGHAM S.: Re-inventing the introductory computer graphics course: providing tools for a wider audience. In *GVE '99: Proceedings of the Graphics and Visualization Education Workshop* (1999), pp. 45–50. 2

[DG06] DOMIK G., GOETZ F.: A breadth-first approach for teaching computer graphics. In *EG Education Papers* (2006), pp. 1–5. 2

[Fel96] FELTON J.: Problem-based learning as a training modality in the occupational medicine curriculum. *Occup-Med-Oxf 46*, 1 (1996), 5–11. 2

[FW10] FURMAN B., WERTZ E.: A first course in computer programming for mechanical engineers. In *IEEE/ASME International Conference on Mechatronics and Embedded Systems and Applications (MESA)* (2010), pp. 70–75. 3

[Ger04] GERODIMOS R.: How to present at conferences: a guide for graduate students. *PSA Graduate Network Newsletter (October 2004)* (2004), 13–16. 6

[Gon00] GONZALEZ R.: Disciplining multimedia. *Multimedia, IEEE 7*, 3 (2000), 72–78. 2

[Gri09] GRISWOLD W.: How to read an engineering research paper. Available from: `http://cseweb.ucsd.edu/users/wgg/CSE210/howtoread.html`, 2009. 5

[IJP*08] IQBAL R., JAMES A., PAYNE L., ODETAYO M., AROCHENA H.: Moving to activity-led-learning in computer science. In *Proceedings of iPED 2008* (2008). 2

[KSC06] KIRSCHNER P. A., SWELLER J., CLARK R. E.: Why minimal guidance during instruction does not work: An analysis of the failure of constructivist, discovery, problem-based, experiential, and inquiry-based teaching. *Educational Psychologist 41*, 2 (2006), 75–86. 2

[Lam86] LAMPORT L.: *Latex: a document preparation system*. Addison-Wesley Longman Publishing Co., Inc., 1986. 6

[Lar09] LARAMEE R. S.: How to write a visualization research paper: The art and mechanics. In *EG 2009 - Education Papers* (2009), pp. 59–66. 5

[LR88] LEVIN R., REDELL D. D.: An evaluation of the ninth sosp submissions or how (and how not) to write a good systems paper. *SIGGRAPH Comput. Graph. 22* (1988), 264–266. 5

[MGJ06] MARTÍ E., GIL D., JULIÀ C.: A PBL experience in the teaching of computer graphics. *Computer Graphics Forum 25*, 1 (2006), 95–103. 2

[PBTF09] PELLICER J. L., BLANES J. S., TORMOS P. M., FRAU D. C.: Using processing.org in an Introductory Computer Graphics Course. In *Eurographics 2009 - Education Papers* (2009), pp. 23–28. 3

[Per92] PERELMAN L.: *School's out: hyperlearning, the new technology, and the end of education*. William Morro, 1992. 2

[PJB*10] POOLE N., JINKS R., BATE S., OLIVER M., BLAND C.: An activity led learning experience for first year electronic engineers. In *Proceedings of the 2010 Engineering Education (EE2010) Conference* (2010). 2

[RF06] REAS C., FRY B.: Processing: programming for the media arts. *AI & Society 20* (2006), 526–538. 3

[SD95] SAVERY J., DUFFY T.: Problem based learning: An instructional model and its constructivist framework. *Educ Technology 35*, 5 (1995), 31–38. 2

[SEA*10] SHUTTLEWORTH J., EVERY P., ANDERSON E., HALLORAN J., PETERS C., LIAROKAPIS F.: Press play: an experiment in creative computing using a novel pedagogic approach. *Anglo Higher 2*, 1 (2010), 23–24. 2

[Sto09] STORNI C.: The ambivalence of engaging technology: artifacts as products and processes. In *NORDIC Design Research Conference* (2009). 3

[Tuc96] TUCKER A. B.: Strategic directions in computer science education. *ACM Comput. Surv. 28* (1996), 836–845. 2

[Tud92] TUDGE J.: Processes and consequences of peer collaboration: A vygotskian analysis. *Child development 63* (1992), 1364–1379. 2

[VW00] VANDERBERG S., WOLLOWSKI W.: Introducing computer science using a breadth-first approach and functional programming. In *Proceedings of SIGCSE 2000* (2000). 2

[WM08] WILSON-MEDHURST S.: Towards sustainable activity led learning innovations in teaching, learning and assessment. In *Proceedings of the 2008 Engineering Education (EE2008) Conference* (2008). 2

[YG96] YATES W., GERDES T.: Problem-based learning in consultation psychiatry. *General Hospital Psychiatry 18*, 3 (1996), 139–144. 2