On the Provision of a Comprehensive Computer Graphics Education in the Context of Computer Games: An Activity-Led Instruction Approach

Eike Falk Anderson[1] and Christopher E. Peters[1]

[1]Department of Computing and the Digital Environment, Coventry University, United Kingdom

Abstract

Over the past decade the development of computer games – which originated in academia with the creation of Spacewar at MIT in 1961 – has evolved into an accepted academic discipline, closely related to the field of computer graphics. Games courses can be found embedded in traditional computer science degrees or as dedicated degree programmes for students aiming to work in the games industry. In this paper we present a student-centred, activity-led approach to teaching computer graphics in the context of a computer games technology undergraduate degree. We describe our computer graphics related courses and demonstrate how they are formed by the activity-led teaching methodology.

1. Introduction

The past three decades have seen massive developments in computer graphics and equally large changes in computer graphics education. While in 1994 the focus on 3D graphics education was proposed as a novel idea [OLPL94], the ready availability of graphics capable computers and graphics APIs such as OpenGL mean that 3D computer graphics are now firmly integrated into most computer graphics curricula [CHLS04]. In a similar vein to suggested computer graphics curricula [CHLS04, Cun07], there are recommendations for computer games curricula from the IGDA (International Game Developers Association), who provide a broad curriculum framework [IGD08], as well as Skillset [Ski08], the UK's Sector Skills Council for Creative Media, who provide degree accreditation based on recommendations from the games industry and academia.

Traditionally, computer graphics education has been a minor specialism in computer science curricula, which in recent years has grown in popularity and importance. The introduction of multidisciplinary courses, such as computer games or computer graphics, into traditional computer science degrees has been seen as a possible countermeasure to the current crisis in computer science education [Car06], which is exacerbating the current skills shortage in the computer games sector [NES08, gam08]. With game programming having gained academic acceptance [AK07], this approach has led to some success in increasing student recruitment and retention in computer science faculties [PRK05]. As a result we have recently witnessed an explosion in computer games related degree programmes – in the UK there are now more than 160 games related degrees [Ski05], including over 80 [gam08] computer games degrees in higher education institutions – that usually also incorporate a prominent computer graphics component. The convergence of modern entertainment media, which all rely on modern computing hardware, has resulted in computer graphics in film effects and feature animation now using the same or at least very similar techniques to those used in computer games. A similar convergence must follow in computer graphics education, as graduates are likely to find employment working with different media. Too often, however, games degrees tend to simply add a flavour of games and graphics to existing computer science degrees [Vol08]. We believe that a better solution is to design new degrees based on interdisciplinary computer graphics education instead.

In a similar manner, the pedagogic model used must reflect the subject being taught. Since it is a necessity for graduates seeking employment in the highly competitive computer games industry to have practical problem solving skills and to be able to take initiative, any games degree programme must cater for these needs. The creative computing 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) [IJP*08] as the pedagogic model for meeting this challenge. In this context the term Activity Led Instruction refers to the instruction of students on how to embrace the ALL process. Exemplar-based activity sessions may be organised with the primary purpose of familiarising students with the process, rather than the content per se. In these sessions, the facilitator becomes an instructor, taking a more active role in demonstrating how a specific, domain-relevant problem may be solved through ALL. Students are thus provided with a concrete, real-world example of the ALL process with which they can begin to relate to new problem domains.

In this position paper we describe our games technology degree in the context of the creative computing philosophy on which it is built, focussing on its computer graphics components. We also present the instructional approach taken in the delivery of these courses, which is Activity-Led Learning, a new student-centred pedagogic model that is a generalisation of Problem-Based Learning [SBM04]. We demonstrate how our degree's computer graphics related courses are formed by this methodology and provide a discussion of our instructional approach, which aims to facilitate the expansion of the activity-led methodology to all areas of our degree programme.

2. Creative Computing in Coventry University's Department for Computing and the Digital Environment

The games technology undergraduate degree in CDE evolved from the creative computing degrees in EC. It was designed following an interdisciplinary approach to computer science, developing the students' programming skills as well as their creative design skills by teaching concepts from computer science and art and design, transcending the art / science divide. This sets it apart from the traditional computer science degrees to which courses on games or graphics have been added in order to increase the degrees' attractiveness to students. While many of these have sprung up in the current climate of decline of computer science in higher education, such degrees cannot really take into account employer expectations. This is one of the advantages of running a dedicated computer games degree, such as our games technology degree. One of its main objectives is to instil in students games industry awareness, i.e. employment oriented knowledge of the workings and the requirements of the games industry, in order to enable students to enter working life prepared to meet the challenges presented by the games industry. This is important, as initially students have many misconceptions of the complexities of game development and the game production process, greatly underestimating the level of difficulty involved. The creative computing group specialises in this type of industry aware education for the creative industries.

2.1. Introduction to Computer Graphics

Following on from a common first year that includes introductory computer programming, system architecture and which the games technology undergraduate degree shares with other creative computing degrees, more specialist courses, including computer graphics courses, are taken by games technology students. From the beginning of their second year, games technology students take a course in graphics programming which covers the bases found in the traditional beginning graphics course [CHLS04]. This course is augmented with a parallel running course on Physics for Computer Graphics that introduces the concepts of rigid body dynamics and particle systems.

2.2. Developing an Interdisciplinary Culture

The current academic year (2008/2009) is the first year in which the final year of our games technology degree is running. One of the aims of the final year of the games technology degree is to develop an interdisciplinary culture in the games technology students, integrating a firm grasp of the technical foundations of computer games technology with an understanding of creative design principles and practice. The core aim is to prepare them for a career in the creative industries. To achieve this, students not only take a course in Advanced Games Programming, which very much focuses on computer graphics and graphics programming, but they also take a course in 3D Modelling and Animation, focusing on the creative and aesthetic aspects of computer graphics.

2.2.1. Facilitating Industry Awareness

The specific skills that the computer games industry expects from graduates are well documented [gam08, Ski08, McG08]. In terms of hard-skills, i.e. technical knowledge and competencies, it is necessary that early specialisation or overspecialisation by students is avoided, as games companies tend to mould new employees to particular areas, often involving in-house developed toolkits and game engines specialised to particular tasks in the company's production pipeline. Instead, students should be exposed to as many different subject areas and techniques as possible, following an educational breadth-first approach to games and graphics [DG06]. Apart from general computer science skills and the obvious subject specific knowledge, students aiming for a career in the games industry also need to demonstrate the ability to learn independently, as they will have to keep up with rapid development and changes in technology [PRK05], and the ability to teamwork, possessing strong interpersonal and communication skills [McG08]. Students develop these soft-skills through their experience of the teaching methodology employed in EC.

2.2.2. Advanced Games Programming

Figure 1: *Exploring game engine architecture.*

The advanced games programming course aims to endow students with practical low and mid-level technical fundamentals and integrative capabilities of necessity for developing and extending games engines, either from the ground up, or through the utilisation and adaptation of pre-existing off-the-shelf libraries and components. This course encapsulates many elements identified with a standard computer graphics course, although their application is not limited to the graphics domain. The course is divided into two conceptual halves. The first half is bottom-up, focusing on fundamental issues of importance for animation, interaction and synthesis of graphical representations. Students implement and investigate low-level techniques by creating interactive visualisations using a graphics API and basic input libraries. This half of the course implicitly illustrates to students through practical means the necessity of adopting a principled approach to implementation, highlighting the purpose of graphics and game engines, which are introduced in the second half of the course (Figure 1). This part of the course, concerned with mid-level techniques for extending pre-existing graphics engines, is top-down in nature so as to provide students with a broader perspective of the technical aspects of game engine development, allowing them to relate different components and their interactions and interfaces.

2.2.3. 3D Modelling and Animation

Knowledge of the mathematical foundations of computer graphics and graphics programming techniques are insufficient for an inclusive interdisciplinary computer graphics education. Games technology students need to gain an understanding of the creative process, its stages and the theories behind it to allow them to effectively communicate with

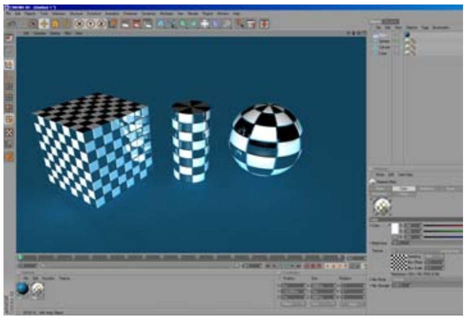

Figure 2: *Experimenting with textures and materials.*

artists and designers. All games technology students therefore take a course on 3D modelling and animation. The main aim of this practice based course is to introduce students to the practical skills that are relevant to the creation and subsequent animation of 3D models. Students learn how to create 3D models and environments using a variety of techniques (Figure 2), providing them with an insight of the processes involved in producing content for interactive multimedia applications and computer games. This course is closely linked to the Advanced Games programming course which reiterates many of the concepts covered in 3D Modelling and Animation from a computer games perspective and also helps students to develop an awareness of the requirements and limitations of current 3D rendering engines and related technologies, which in turn informs their creative practice.

3. Activity-Led Learning

Activity-Led Learning (ALL) is a pedagogic model pioneered at Coventry University. It is related to scenario based teaching, learning and assessment [IE05] and can be viewed as a generalisation of Problem Based Learning (PBL) [IJP*08], taking a student-centred approach to the learning experience and encouraging students to "learn by doing" and therefore to assume responsibility for their student experience.

3.1. ALL at Coventry University

ALL differs from traditional teaching approaches in that students are viewed as active participants in the learning process rather than passive consumers of information. Unlike traditional techniques, the accomplishment of a specified activity leads the learning process, rather than the activity merely acting as a support for tuition. Furthermore, in ALL educators assume the role of *facilitators* who address students' needs in a primarily reactive mode. During the process, students are encouraged to reflect upon their own development, both cognitive, for example in terms of the problem solving approach adopted, and emotional, for example, in terms of coping with the demands of independent thought and teamwork.

From the institutional perspective, ALL is desirable as it is envisaged to allow improvement of engagement with students, enhance achievement and hone skills that are applicable to both industry and postgraduate research.

ALL encourages students to show initiative by demonstrating a proactive role in problem-solving. Furthermore, students are encouraged to discover optimal approaches to problem-solving and adopt problem-solving strategies, such as dividing more complex problems into subparts that can be solved individually. In addition, ALL is concerned with *capacity building* and *sustainability* [WM08]. In this respect, a most important goal of ALL is to endow students with the ability to learn how to learn and thus become independent in charting their intellectual pursuits.

ALL can be regarded as sharing many similarities with PBL [MGJ06]. Problem Based Learning is concerned with problem solving "without the traditional lecture and tutorial provision" [IJP*08], which has also been a source of criticisms of this educational approach [KSC06]. ALL can be differentiated from PBL as it is more adaptable, ranging from small tasks to longer projects [IJP*08]. Thus, it is more suitable to being interleaved, to a variable degree, with traditional teaching techniques: a course may consist of primarily traditional teaching techniques intermixed with one or two activity-led tasks, or may be fully oriented towards a single broad activity, in the form of a project, resembling PBL. Intermediate mixes are also possible.

ALL is believed to map well onto computer science, as this "has traditionally embraced practical activity" [IJP*08].

3.2. ALL in the Context of Creative Computing

We believe ALL also to have great potential benefit to creative computing, given its focus on a practical learning-by-doing approach. The adoption of ALL techniques need not imply a radical shift in course design. ALL is flexible and may be adopted piecewise, interleaved with traditional teaching methods. For this very specific purpose, we have enumerated a number of different categories that can help to apply ALL in current course designs. These categorisations have helped us integrate ALL into the current games technology degree and start a transition towards an activity-led degree. Activities can fall into four general categories, each category operational within a single course or across several courses.

- Independent activities
- Incremental activities
- Iterative activities
- Integrative activities

In the most basic case, activity outcomes may be short-term and independent from each other. We have found this method fits well into models involving a lecture and lab per week and requires minimal changes to introduce activities into the course. It is a good starting point for testing activity-led methods. Incremental activities build up on previous activities on a week-by-week basis, so the starting base of work for one week's activity is based directly on the activity that the students have completed on a previous week. Iterative activities refer to those that reiterate and elaborate on concepts previously covered. New material may not be involved: rather previous material may be considered at greater detail, difficult concepts can be revisited or new problem-solving approaches can be encouraged towards solving previously attempted tasks in better ways. Finally, integrative activities may amalgamate the results of several independent, iterative, incremental or integrative activities. This integration may take place within a single course, or can work to bring together concepts across several courses. These are basic building blocks of ALL course design.

Thus far, we have adopted a combination of activity types in combination with traditional learning practices. The principles and techniques explored in workshops and exercises are explained in lectures, relating them to real-world problems, as well as the activities that the students are engaged in. These activities are incremental, i.e. they build on one another, progressing from simple problem solutions to more complex digital artefacts, each taking the form of a mini-project and thus allowing the students to build a portfolio of work and construct an integrated demonstration of their capabilities for future presentation to prospective employers. Assessment is conducted by evaluating this portfolio, which provides a relatively accurate reflection of the students' progress over time.

4. Graphics for Game Development – Sample Syllabus

Activities that students engage in through independent study are central to our games technology degree. Consequently, the computer graphics related courses do not rely on textbooks, as following a specific text does not map well onto ALL. However, a text recommended to students is "Mathematical and Computer Programming Techniques for Computer Graphics" [Com05]. They are encouraged to consult this if the need arises, as this book provides a well structured and easy to follow introduction to the mathematical principles and techniques that the students are required to master in order to succeed with their activities. These activities are used to bridge the divide between creative design and programming as students are solving problems that must be approached through the combination of different disciplines. These can be problems related to computer game artificial intelligence (AI), a subject covered by another course taken by games technology students, which can be solved using transferable skills obtained through the study of computer graphics. This is possible because there are substantial overlaps between these disciplines in some areas. Examples for this would be crowd simulation and related techniques, such as flocking, which were originally developed as graphical special effects, based on particle systems [Rey87].

4.1. Advanced Games Programming

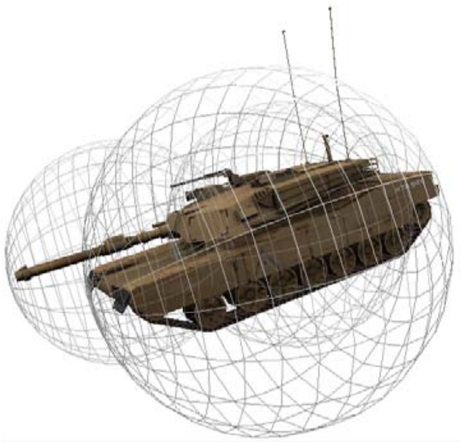

Figure 3: *Visualising bounding volumes.*

Subject areas that are covered in the Advanced Games Programming course are:

- Maths for Computer Graphics, which reinforces existing knowledge, broadens it and illustrates its application for practical use in computer programming. It elaborates vector, matrix and quaternion algebra and introduces parametric curves and surfaces, geometric subdivision and level-of-detail techniques Activities running in parallel are incremental, so students build up their own library of basic 3D operations and visualisation aids to provide them with a code base for use in later parts of the course.
- Object Representation and Interaction, which presents the construction of hierarchical objects, bounding volumes and hierarchies in the context of intersection-tests and collision detection techniques. Building on the maths subject area, which can sometimes appear abstract, this area introduces geometry and the use of mathematical techniques to allow interaction to take place between objects. Activities that may be attempted by students could be the loading of a mesh using an object loader library, in order to create hierarchical bounding volumes to experiment with collision detection (Figure 3).
- Game (Engine) Architecture, which focuses not only on the overall architecture of game engines but also on topics such as the game loop, engine subsystems and middleware, as well as related subjects like scripting (languages) and mobile games programming. Activities that students might engage in could be the integration of an existing game engine with a scripting language, allowing the script-driven creation of a simple computer game.
- Scene Management and Animation, which covers scene management techniques such as Binary Space Partitioning (BSP) trees [FKN80], rigid body animation and real-time character animation [And01] methods and visual effects, using particle systems, decals, billboards and impostors for creating rich virtual environments. Activities that students may undertake include the integration of an animated character into a computer game by adding an animation subsystem to a game engine.
- Advanced Rendering, examining advanced topics in rendering and shading, including more recent developments in real-time global illumination, as well as shader writing techniques for the programmable rendering pipeline. Activities that students may be asked to undertake may include the replacement of fixed function graphics pipeline functionality in a given graphics program with programmable shaders.

4.2. 3D Modelling and Animation

Subject areas that are covered in the 3D modelling and animation course are:

- Computer Animation History, which briefly covers the history of computer graphics and computer animation, also providing an overview of related subject areas. Activities that run in parallel to this theme are mainly experimentation with the available 3D modelling software to familiarise students with its user interface.
- Virtual Cinematography, which concentrates on the virtual camera as well as virtual lighting, handled in the context of the underlying maths and physics concepts as well as the theories of traditional film and cinematography. Activities that students may be asked to undertake include lighting and framing of existing (pre-produced) 3D scenes in order to evoke different moods.
- 3D Modelling, focussing on basic approaches to 3D modelling, but also including texturing of objects. Activities that the students engage in may include the creation of 3D models using different techniques.
- Character Design and Modelling, covering both the aesthetic considerations that influence the design of a biped (humanoid) virtual character, as well as the modelling techniques for the realisation of such a character. The latter relates directly to the preparation of the virtual character for subsequent animation and activities that fit into this subject area may range from character design on paper to character modelling and rigging in a 3D modelling program.
- Computer Animation, which introduces traditional animation theory and practice and relates these to modern computer animation methods and techniques, such as real-time character animation for computer games [And01]. Other elements that are covered are past and current practices in animation production. Activities that students may undertake include the preparation of an animated virtual character for integration into a computer game.

- Advanced Concepts in 3D modelling and animation that are covered are recent approaches to 3D rendering, procedural content creation, post processing and modern games, animation and effects industry practices. Activities that students may engage in include the compositing together of live action footage with computer animated artefacts.

5. Discussion

The courses described above form an exemplar, illustrating one way in which a standard course can start to be transitioned towards the ALL approach. Course contents are flexible and subject to change according to the composition of the student body, feedback from students and results of evaluations.

Since ALL is generally applicable, there are many other options: above all, an advantage of ALL is that an all-or-nothing approach is not necessary. One need not make a decision to either adopt ALL or continue with traditional teaching methods, but can choose to mix methods as necessary. However, ALL is a different approach to learning that students may not be familiar with. Therefore, we suggest Activity-Led Instruction to instruct students about the method they will be using, inform them of what is expected of them, provide an example activity and illustrate the process by which it might be solved. Such instruction is envisaged to be important in easing the transfer of students into the mindset required for succeeding in ALL and to provide an exemplar to enable initial understanding of the approach. Students must be aware of what they are doing and why they are doing it, and must also receive appropriate feedback to encourage them in the understanding that they are taking the correct approach by moving beyond course instruction and material.

While we are still in a phase of transitioning towards ALL that is not yet complete, there seem to be a number of apparent issues that require mention. These generally concern implementation issues and methods for evaluating the approach.

5.1. Key Implementation Issues

Here, we highlight a number of outstanding issues that prospective ALL practitioners should be aware of when considering the adoption of the method.

- Base Competency
  To be effective at teaching content rather than process, ALL most likely requires students to possess some degree of base competency in the subject area before engaging in activities [Mer07].
- Preparation
  The expectation that students should demonstrate initiative should not be misinterpreted to imply reduced responsibility or workload on behalf of the facilitator for preparing and planning materials and sessions for activities. See below for a discussion on the topic of time involved.
- Unexpected Solution Paths
  Since students have freedom in how they approach a task, it is possible that their solution path may miss a specific aspect of vital importance to the course [IJP*08]. The facilitator should therefore be tasked with identifying these situations and insuring that students are made aware of material that has been missed.
- Teamwork
  Although teamwork need not necessarily form part of an ALL programme, it is likely that many activities will involve it to some degree. In this case, stronger students may have a tendency to feel they are expected to support some of their less capable teammates without receiving due recognition. This implies the need to define appropriate assessment / marking schemes that recognise and award individual achievement in the team-context.
- Collaboration and Referencing
  Since personal initiative, collaboration and the consultation of online material are actively encouraged, students need to be made aware of the necessity of accurate referencing of sources used and collaborations engaged in during the project. They must also be provided with precise definitions of what constitutes plagiarism. As above, this implies the need for assessment methods suited to ALL and collaborative work.
- Culture
  The cultural background of the student demographic is an important factor of consideration when adopting ALL. The approach may initially be more suitable to some backgrounds for socio-cultural reasons, while others may need additional time for adjustment.
- Managing Expectation
  Interactive entertainment technologies are inherently engaging, stimulating and motivating for students, in some cases attracting those to the sciences who might not otherwise have shown interest. One must be aware, however, of potential pitfalls relating to high expectations of fun from newcomers who may have incorrect preconceptions about the nature of study involved. It is important to instil the fact that a large proportion of the subject-area relies exclusively on programming and mathematics. High-level integrative activities exposing some low-level technical aspects can help to provide students with an overview of the techniques that they will later investigate in more detail.

The purpose of this list is to focus attention on some of what we regard as key issues of importance when considering adopting the ALL approach. This list of issues is certainly not conclusive or complete and must remain speculative until informed evaluations can be conducted, as described next.

5.2. Evaluation

Evaluating the successfulness of ALL techniques in the context of computer games is a challenging endeavour. Since we are still in a phase of transitioning towards ALL that is not yet complete, our discussion here is limited to an enumeration and discussion of some of the most important elements for consideration when attempting to conduct evaluations. These are currently based on anecdotal evidence and observation through liaison with teaching staff within our department and will form a basis for future empirical inquiry.

When considering an evaluation of ALL, it is important to account for the specific domain to which it is to be applied. Due to the practical nature of graphics programming and design for games, our evaluation hypothesis is that the adoption of ALL elements should prove to be more effective than traditional learning approaches alone, accounting for a number of factors. This proposition is based on the similarity that ALL has to PBL, which has been pioneered in medicine studies and introduced in domains requiring strong professional training, such as nursing and law [MGJ06]. Indeed, evidence suggests that attitude and problem-solving skills are sought-after soft-skills in the games industry [McG08].

Effectiveness can be regarded as a composition of components. First of all, student retention may be considered as a metric for effectiveness. It is possible that group-based activities can provide a social support structure to retain students who might otherwise leave. Secondly, student experience may be considered. Associated time cost is an important issue here, as it has been noted that in the case of group-activities a warm-up period may be required in order for students to become familiar with ALL and comfortable with the associated methods. Finally, employer and academic requirements should be considered. The ALL process may improve retention of students, and provide student satisfaction, but may not convey content of high relevance to employment or fulfil academic requirements.

In addition to effectiveness and other possible benefits of the approach, an evaluation should also account for the possible costs in order to insure it is practical for the situation in which it is being employed. In this case, time is an important cost factor to be considered. In relation to the time costs associated with PBL and ALL, opinion seems mixed: The literature on PBL suggests that some time will have to be spent on preparation, but not as much as one might assume [MGJ06], whereas Iqbal et al [IJP*08] suggest that this is not the case, but that instead it takes less time than traditional teaching methods.

A final item of discussion concerns the difficulty of integrating a focus on vocational skills e.g. those demanded by game companies, with the maintenance of degree worthiness and academic-orientation. A delicate balance must be achieved between these aspects to insure that all necessary academic requirements are fulfilled and postgraduate opportunities in both industry and academia are not limited. The overriding aims of ALL, focusing on student initiative and ability to problem solve, seem highly compatible with these requirements: we are optimistic that a proper future evaluation will shed light on the arguments presented here.

6. Acknowledgements

We would like to thank our colleague Peter Every for comments and suggestions that helped to improve this paper. We also wish to extend our thanks Susan Moron-Garcia for providing us with useful feedback, as well as general encouragement.

References

[AK07] AMRESH A., KARNICK P.: Creating Interest in Computer Graphics by Teaching Game Development . In *EG Education Papers* (2007), pp. 9–16. 1

[And01] ANDERSON E. F.: Real-time character animation for computer games. Project Report, NCCA, Bournemouth University, 2001. 5

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

[CHLS04] CUNNINGHAM S., HANSMANN W., LAXER C., SHI J.: The beginning computer graphics course in computer science. *Computer Graphics 28*, 4 (2004), 24–25. 1, 2

[Com05] COMNINOS P.: *Mathematical and Computer Programming Techniques for Computer Graphics*. Springer, 2005. 4

[Cun07] CUNNINGHAM S.: A computer graphics curriculum to meet the european bologna requirements. *SIGCSE Bulletin 39*, 3 (2007), 310–310. 1

[DG06] DOMIK G., GOETZ F.: A Breadth-First Approach for Teaching Computer Graphics . In *EG Education Papers* (2006), pp. 1–5. 2

[FKN80] FUCHS H., KEDEM Z. M., NAYLOR B. F.: On visible surface generation by a priori tree structures. In *SIGGRAPH '80: Proceedings of the 7th annual conference on Computer graphics and interactive techniques* (1980), pp. 124–133. 5

[gam08] Games up? Available from: http://www.gamesup.co.uk/, 2008. [Accessed 08/12/2008]. 1, 2

[IE05] IQBAL R., EVERY P.: Scenario based method for teaching, learning and assessment. In *SIGITE '05: Proceedings of the 6th conference on Information technology education* (2005), pp. 261–266. 3

[IGD08] IGDA EDUCATION SIG: Curriculum framework - version 3.2 beta. Available from: http://www.igda.org, 2008. [Accessed 08/12/2008]. 1

[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, 3, 4, 6, 7

[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. 4

[McG08] MCGILL M.: Critical skills for game developers: An analysis of skills sought by industry. In *Proceedings of ACM FuturePlay 2008* (2008), pp. 89–96. 2, 3, 7

[Mer07] MERRILL M. D.: A task-centered instructional strategy. *Journal of Research on Technology in Education 40*, 1 (2007), 33–50. 6

[MGJ06] MARTÍ E., GIL D., JULIÀ C.: A pbl experience in the teaching of computer graphics. *Computer Graphics Forum 25*, 1 (2006), 95–103. 4, 7

[NES08] NESTA POLICY & RESEARCH UNIT: Level up - building a stronger games sector. Policy Briefing LU/31, available from: http://www.nesta.org.uk/level-up-building-a-stronger-games-sector/, 2008. [Accessed 08/12/2008]. 1

[OLPL94] OWEN G. S., LARRONDO-PETRIE M. M., LAXER C.: Computer graphics curriculum: Time for a change? *Computer Graphics 28*, 3 (1994), 183–185. 1

[PRK05] PARBERRY I., RODEN T., KAZEMZADEH M. B.: Experience with an industry-driven capstone course on game programming: extended abstract. *SIGCSE Bulletin 37*, 1 (2005), 91–95. 1, 3

[Rey87] REYNOLDS C. W.: Flocks, herds and schools: A distributed behavioral model. In *SIGGRAPH '87: Proceedings of the 14th annual conference on Computer graphics and interactive techniques* (1987), pp. 25–34. 5

[SBM04] SAVIN-BADEN M., MAJOR C. H.: *Foundations of Problem Based Learning*. Open University Press, 2004. 2

[Ski05] SKILLSET: Computer games accreditation: Current education report. Available from: http://www.skillset.org/games/accreditation/article_3700_1.asp, 2005. [Accessed 08/12/2008]. 1

[Ski08] SKILLSET: Degree course accreditation guidelines for computer games. Available from: http://www.skillset.org, 2008. [Accessed 08/12/2008]. 1, 2

[Vol08] VOLK D.: How to embed a game engineering course into a computer science degree. In *Proceedings of ACM FuturePlay 2008* (2008), pp. 192–195. 1

[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). 4