Introducing Game Development into the Computing Curriculum – A Progressive Methodology

Colin B. Price, John Colvin **and** ***Warren Wright, University of Worcester***

Abstract

The Computer Game industry is big business, the demand for graduates is high, indeed there is a continuing shortage of skilled employees. As with most professions, the skill set required is both specific and diverse. There are currently over 30 Higher Education Institutions (HEIs) in the UK offering Computer games related courses. We expect that as the demand from the industry is sustained, more HEIs will respond with the introduction of game-related degrees. This is quite a considerable undertaking involving many issues from integration of new modules or complete courses within the existing curriculum, to staff development. In this paper we share our experiences of introducing elements of game development into our curriculum. This has occurred over the past two years, starting with the inclusion of elements of game development into existing programming modules, followed by the validation of complete modules, and culminating in a complete degree course. Our experience is that our adopting an progressive approach to development, spread over a number of years, was crucial in achieving a successful outcome.

Keywords:

Computer Games, Multi-media products, Course Development

Introduction

The University of Worcester is an expanding HEI that acquired University title in 2005. Computing at the University of Worcester is situated within the Business School. We are a relatively small section with a typical intake of up to 100 students onto the computing degree course. This course, validated in 2003 is modular and runs in a variety of pathways including single and joint honours. The computing curriculum has been developed around a number of threads, Digital Technologies, Information Systems and Databases, Networking, Multimedia and Programming. These threads reflect both staff expertise and perceived market demand. We continually monitored student choice of threads and associated modules, and feedback from their module evaluations. It became apparent that modules with a more theoretical or abstract content (such as Operating Systems, Computer Architecture, and to some extent Programming) were less popular. Our students were becoming keen on the development of graphical content, and also in the use of visual simulations in their learning activities. One of the authors, responsible for teaching this material, made the decision to suspend the Operating Systems and Computer Architecture modules and replace them with Games Development modules. But this did not happen overnight. In effect, we invoked, implicitly, a "Progressive Methodology" to move from the 2003 Computing degree course to the 2006 portfolio of 5 degree courses, one of which is the "Computer Games and Multimedia Development" degree.

The stages involved in making these changes were

(i) the introduction of elements of computer games into existing modules (specifically Object Oriented Programming and Multimedia programming)
(ii) the introduction of new game-based modules at levels 2 and 3 (2005). These were game development (using a commercial game SDK – Unreal Tournament), and Multimedia game development (using Director and Lingo)
(iii) the development of a degree course "Computer Games and Multimedia Development" comprising mutually-supporting modules (2006).

It is interesting to note that our development was not planned, there was no “grand design”. It arose from the independent activities of two academics. As these activities became more focussed, the strategic mechanisms within the institution strongly influenced the development, as was a desire to make our course portfolio more appealing. This paper is structured as follows. Each section discusses one stage in our progressive development. Within each section we present:

a) the rationale for the development
b) the issues involved in developing the module content and the approaches to learning
c) the actual materials developed
d) lessons learned from our experience.

A following section discusses our plans for the future, and we present some conclusions. An overview of the “timeline” of the development is shown in Fig.1

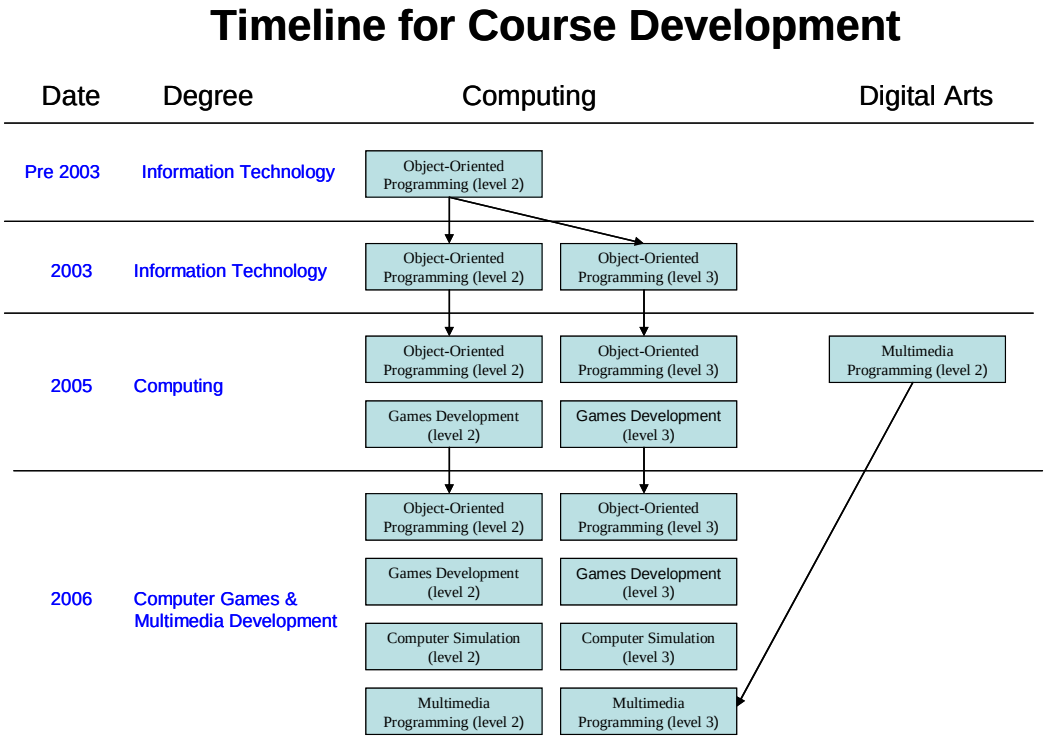

Figure 1. Timeline of our progressive development towards a Computer Games degree course

Stage 1: First steps - Introducing Game Elements into Existing Modules

Great ideas or cultural developments may appear simultaneously in several places or minds (Kuhn, 1977). So was it at Worcester where two lecturers independently decided to use game-type activities within existing modules and used a game-based “product” to satisfy the original intended learning outcomes (ILOs) of these modules. Perhaps interestingly, these modules both involved “programming”; the first a “Multi-media programming” module (Macromedia Dreamweaver with Lingo) delivered to students on a “Digital Arts” degree, and the second a module of “Object Oriented Programming”, delivered within Computing. Macromedia Dreamweaver is a widely used authoring tool (Savatierra et al, 2003) and also highly-regarded and robust (Neubauer, 2002).

Inclusion of games within the Multi-media module was not an intentional learning and teaching strategy, rather the intention was that learning activities offered to students should involve a mixture of multi-media products. It is recognised that students showed more enthusiasm for activities involving elements of games. We also suggest that these activities incite processes of deep learning (Marton & Saljo, 1976), that they fire up a desire to understand in order to achieve the final product, that they encourage self-exploration of related topics (such as trigonometry). There was clear evidence of activity outside the classroom as shown by email contact with the lecturer between sessions. The module is now taught with a much greater emphasis on the construction of games. Learning activities include discussion of programming principles, reflection on technical issues, often done "in groups", and individual or group creation of multimedia products.

Turning to the second module, "Object Oriented Programming", this recently validated module had never previously run, we had free reign on designing the learning and teaching styles and the module content. The supporting level-2 module had been theoretical, emphasising the concepts of OOP. But the medium of exploration in this module (which used Java) had relied on a numerical or textual interface. In moving to the level-3 module we decided to use the graphical capabilities of Java, in particular, "Swing". Our experience in teaching programming strongly suggests that a visual interface was both engaging and effective in teaching concepts of programming. We had previously taught Java using simulated robots and the generation of computer art. When presented with alternatives to the vehicle of instruction (graphics – games, databases, web-applications) the students unanimously elected game development. The response of students was very positive, in line with the experiences of Sweedyk & Keller (2005). Details of the course and its pedagogy are found in (Price 2006).

Naturally there were constraints on this development. There was a need to adopt the section's constructivist approach to learning (Colvin & Phelan, 2006). The material was to be used within existing modules, so it had to satisfy the existing ILOs. There were of course resource constraints, time needed for development of the material and software resources. In summary the motivating factors for including of game elements are

(i) a desire to inspire learning
(ii) a wish to engage the obvious enthusiasm of our students
(iii) a response to the personal interests of lecturers involved.

Stage 2: A Preliminary Games Module

"The time was right!" is perhaps the best way to summarise our decision to develop dedicated game modules as the flow of lecturer and student interest awoke and converged. Discussions among the computing staff at Worcester developed a larger vision of how our curriculum should develop. It was clear that we needed to gain experience of teaching game development. A "Preliminary Module" in game development was constructed with the following aims:

(i) to investigate whether our students were able to work at an appropriate level of technical and creative ability,
(ii) to investigate how to deploy materials which would engage students of different abilities
(iii) to discover which aspects of game development our students found easy, difficult and rewarding
(iv) to evaluate various approaches to learning. Our intention was then to revalidate this module based on this preliminary experience.

Issues influencing the design of the module were various. Perhaps the most important factor was holding discussions with a local games company during the design phase. We of course investigated activities of other HE institutions. But, most importantly, we researched the use of game technology in the production of virtual reality "Immersive Environments", e.g. for use in education and training (Carpin et al.) Various technical issues were addressed before the writing the module specification; what game engine to use, what supporting software was required? Investigations of various game SDKs available directed us to one clear leader,

"Unreal Tournament". This engine comes in two flavours, "UT2004" which supports various game-types such as "First Person Shooter", "Capture the Flag", etc., and "Unreal2Runtime" which has no weapons or assets-of-combat, ideal to develop IEs! Also, the engine is built according to principles of OOP. This is apparent in the scripting interface, where it is very easy to program in a Java-like syntax. The design of this module was informed by our desire to provide a valid learning experience for our students, but also to enable us to perform the investigations mentioned above. Professional game production involves teams of specialists each contributing a core development activity (cite IGTA). Typical activities include level design, 3D content creation, AI scripting, game-play production. There are also aspects of the theory of games ("ludology"), and the "business" of game production. We wished to expose our students to all of these elements, and to evaluate their performance in each of these elements. We also wished to create an "open" learning and teaching forum, where various approaches could be tried.

Before setting down to an implementation, we reflected upon our pedagogy, especially the importance of learning activities supporting the students in constructing their own understanding (Biggs, 2003). This suggested the development of a series of work-book based activities where students were led progressively through a series of graded tasks to explore the mechanics of game development.. Our experience is that students find these interesting, challenging and at an appropriate level (Colvin & Phelan 2006). These activities, prepared by the Tutor, also led to the development of Tutor expertise. As students progressed through this material, the Tutor would provide short, formal inputs to the class when appropriate, both to explore problems and to provide extension material. There were also weekly formal inputs from the students on aspects of game theory. These were, essentially, chapters from Rolling's book (Rollings & Adams, 2003). The workbook activities commenced using the "Unreal2Runtime" SDK which provided a limited set of resources (object, textures), and as mentioned above, no fighting characters. Students learned how to develop IEs, involving rooms, "friendly" characters, movers. They were exposed to scripting, and developing intelligent "bot" activity. Only later was the full game introduced.

Evaluation was done both by Tutor observation on a weekly basis, and by the end-of-module student evaluation. All students engaged with the material and produced unique and viable IEs, working at a level appropriate for their abilities. Yet they were soon were outperforming our expectations. The able students worked beyond the workbooks, researching and using additional tutorials and materials available. There were no issues with gender, and our anticipated problem of "thinking in 3D" just did not materialise. All students were able to produce an IE according to an individually chosen "theme", this was researched and in all cases an aesthetically pleasing and highly individual product was realised. One aspect was not well-embraced by any student, this was the creation of 3D content (using Maya or 3DSMax). This is understanding since this work is time-consuming, and requires even more artistic skills.

Constraints on this development were several: The modules had to be developed to maintain the integrity of the existing computing degree course, to align with the course aims and ILOs and to respect the subject benchmarks. Also teaching the material in two modules is a severe constraint, where creative, technical and theoretical aspects all had to be addressed. We have had a similar experience when teaching programming to non software-engineering students (Colvin 2005). Again there were resource issues. In summary, the motivation for validating games modules is

(i) response to student demand,
(ii) a first step in developing a full games degree course
(iii) a strong pedagogical base. It should be noted that there was little institutional drive or involvement at this stage.

Stage 3: The Games and Multimedia development Degree

There were a number of issues with the existing computing course (validated 2003) which led us to review our provision. We had in the course of previous developments amassed a large

number of mandatory and optional modules. The national trend in recruitment into computing is downwards, resulting in loss of viability of some modules. Often modules were being cancelled at the last minute which led to frustration of both staff and students. In a series of brainstorming and development sessions, we agreed to rationalize our offering, to reduce the number of modules so that all would guaranteed to run, and worked out the details. We reduced the number of mandatory modules and introduced a wider student choice.

Out of our discussions, there emerged five “pathways” of study within a single computing degree: Games and Multimedia Development, Information Systems, Web-Applications, Networking and “vanilla” Computing. During the process of course review, the University strongly advised us to market these “pathways” as full and individual degree courses, so to make their flavours instantly visible in our prospectus and through UCAS. It was decided to adopt this advice. Small groups of tutors worked on each course with three objectives, first to identify if any new modules needed to be written, second to identify which modules were irrelevant and should be scrapped, and third to identify the “core” modules suitable for each course from the pool of remaining modules. There was some iteration in this process, which has led to the structure shown in Figure 1. While this “bottom-up” process was running, we kept an eye on the existing program specification, its aims and objectives, and worked to establish alignment between these, and the module intended learning outcomes, approaches to learning, and assessment. The computing benchmarks and Frameworks for Higher Education documents were never out of sight! Again there was some iteration here.

Students enrolling on a degree course with “Games” in the title may well have an expectation of employability in the games industry. Realizing this is a prime factor in the course design. But so are the constrains of the existing course and of the other proposed courses. These constraints emerge as the consequence of staff interests and the small size of the section. Again our design was motivated by discussions with local games companies, and a reading of the International Game Developers Association (IGDA) Curriculum Framework (IGDA 2002) and report ion the IGDA 2006 Curriculum workshop (IGDA 2006).

The course structure (Figure 2) shows the game thread and the multimedia thread embedded in supporting modules drawn from the core threads of the other degree courses. The game thread contains two mutually supporting modules. The “Game and Immersive Environments” ILOs involve elements of analysis and synthesis. Existing commercial games are analysed and so are the principles and theories of game design. Attention is given to the working of AI scripting and the physics engine. While working towards these outcomes, students are simultaneously using a game SDK to produce their own levels.

Generation of 3D content is located within the “3D Modelling and Simulation” module where industry standard packages (3DSMax or Maya) can be used. But there is an emphasis on visual design including 2D drawing, composition and geometry. One ILO specifies an evaluation of architectural principles (building, rural and urban design). The rationale is through a study of the physical world and interaction with humans in this world, to be able to create effective virtual worlds, IEs and computer games.

In previous offerings of our degree course Multimedia modules always proved popular with students. These modules were offered at level 1 and level 2, with recent modules using Macromedia Director as their implementation software. Predominately, the content of these modules was of a creative, rather than technical, nature and the feedback from employers and Alumni suggested that the employability of students might be improved through an increased focus of technical aspects of Multimedia. Andersen et al (2003) argues that not including programming in such a curriculum would deny students exposure to a defining characteristic.

Elements of creativity are already present in the existing level 2 and 3 multimedia modules. These include Interface Design, 3D Modelling and Advanced 3D Modelling as mentioned above. However, these modules build on fundamental design principles delivered at level 1 in Introduction to Web Development. We decided, in order to maximize the sharing of modules with other degrees, to use the level 1 programming skills delivered in Introduction to Web Page Scripting as a platform to offer technical multimedia programming modules at levels 2 and 3. Feedback from current students and Alumni suggested that the multimedia

programming module currently being delivered to students on a “Digital Arts” degree (see Stage 1) could be used as a development platform. Level 2 and level 3 modules were developed that both support students in their solution of problems of increasing complexity, and introduce a range of technologies e.g. 3D programming and external databases.

The progressive approach to development has entailed different academics collaborating and bringing their own teaching styles to each level of development. There has never been an attempt to prescribe pedagogy into any iteration of development, in fact the authors believe that the subtle differences in approach add to the richness of the offering and delivery enjoyed by students.

The academics do share many core pedagogic beliefs and, for example, they all advocate and practice student-centred learning (Colvin & Phelan, 2006) and exploit the benefits of students learning from each other (Rowntree, 1995). This commonality is currently expressed in a shared Learning and Teaching Strategy that acknowledges the disparate educational backgrounds of students enrolled on our courses. Students are encouraged to engage in deep learning (Marton et al, 1976) by an emphasis on student-centred learning activities instead of relying solely on a traditional lecture approach. In order to fully exploit the benefits of student-centred learning, care is taken to ensure alignment of the learning activities, Intended Learning Outcomes and assessments (Biggs, 2003). The Learning and Teaching Strategy also requires that generic skills are embedded into the curriculum, an approach the team has previously found effective (Colvin & Keene, 2004). Although e-learning is not employed exclusively in any of the modules, staff make extensive use of many of the facilities and benefits associated with e-learning (Salmon, 2000) in order to enrich the learning process and to enhance the support that is offered to students.

The subtle differences are evident in emphasis of approach and context that each academic believes is most effective in enabling student-learning in a particular modules. One academic employs a student-centred learning model that occurs in a rigorous and realistic development environment, and that emphasizes good software engineering practice (Colvin, 2005). Another academic rejects any a priori cognitive or affective models of learning and has adopted a subject-content-based informed pedagogy that uses coordinated visual and verbal learning experiences to enable a rich learning environment (Price, 2006)

Constraints on this stage in the development were numerous: There was a need (due to our need for rationalization) to integrate games and multi-media. Our modular system implies overlap with other courses, our modules had to fit into the whole portfolio. In other words our modules had to be “sharable” with other courses and *vice versa*. Again there was the institutional drive to produce viable courses. The effects of these constraints were dissipated by an iterative development process within this stage, where there was active collaboration between the teams developing the different degree courses. Course creativity was thereby not stifled by any project management infrastructure which may have been deemed necessary to monitor and control the development of tightly coupled courses. It is interesting to speculate whether our collaboration was enhanced the community and social atmosphere that exists in our small team. The teams successfully “shared” modules e.g. we included, *inter alia*, Web Page Scripting and Introduction to Web Development, (from the Web Development Degree), so responding to the institutional drive for viable courses, but without jeopardising the integrity of our course.

In summary, motivation for developing this course originated as outline above, but a most significant factor was the institutional drive to produce courses rather than pathways, to increate their visibility and so improve recruitment into computing.

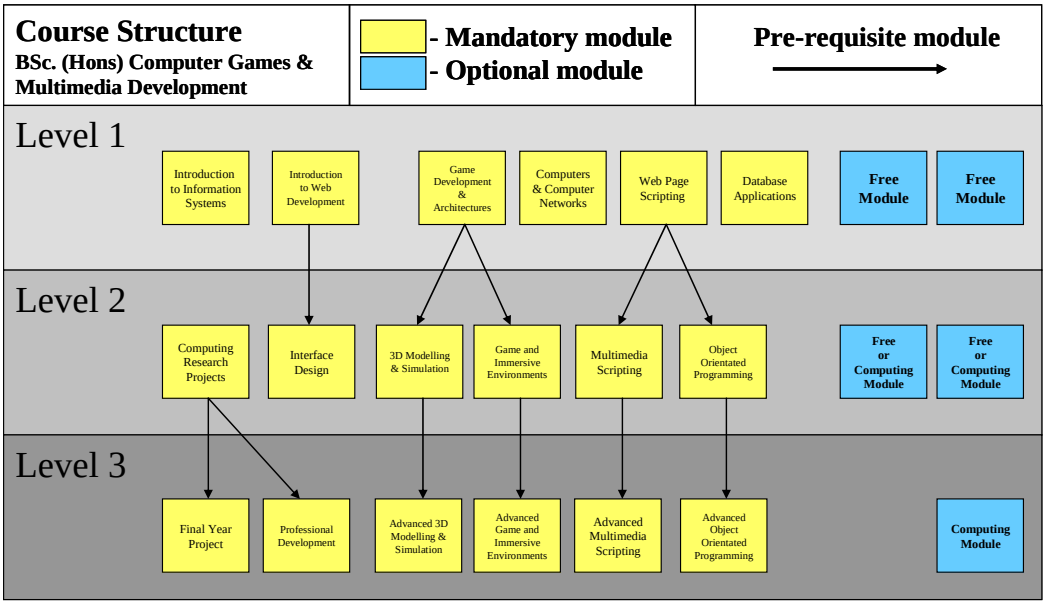

Figure 2. Modules comprising the Game and Multimedia Development degree course.

Notes: The arrows indicate pre-requisites
The Final Year project is a double module

Stage 4: The Future, A Computer Games Degree Course

We hope the Computer Games and Multimedia degree will recruit well and show good levels of student achievement and retention. If this is the case, then we expect pressure from the University (at executive and departmental levels) to develop a specialist Computer Games degree. We anticipate a positive response from the computing team. Our future work will be informed by all stages of our progressive approach to development. One vital area is the development of necessary staff expertise. Since we do not expect any new appointments to the section, staff development is vital. Again, our progressive approach will support staff development in a realistic way.

The development of a specialist Computer Games degree should be principally informed by the needs of the industry, indeed we are already laying the foundations for such a course by engaging with local games development companies. In the meantime we have evaluated our Computer Games and Multimedia course against criteria extracted from Skillset documentation. We have chosen to use the report *Computer Games Accreditation, Industry Opinions on Higher Education,* (Skillset 2006b) which is the industry's response to the Skillset *Games Accreditation Project*, since this report contains a most current reflection of the industry's needs. This evaluation is presented in Table 1, which shows a mapping from the Skillset criteria to the modules comprising the degree course. We have used the module intended learning outcomes, learning and teaching approaches, and assessment methods to effect this mapping. Filled diamonds represent major matching, open squares minor. There are several clear observations we can make: (i) The coverage of the criteria is quite good with most criteria being matched when all modules are considered, (ii) There are clear "holes" in the matching; mathematics is poorly matched and optimization not at all, (iii) It is a significant feature of our course that 2D Drawing and Painting, Image Creation and 3D Art are strongly matched, and that the Level 3 programming module contributes to this, (iv) Involvement with industry occurs principally in the final year, Level 3. We envisage that further analysis of this mapping when taken together with our experience in delivering the Computer Games and Multimedia course will help inform the assembly of the future specialist course

Skillset Criteria	Level 1			Level 2						Level 3					
	Game Development	Web Development	Web Page Scripting	Game and IE Devel.	Modelling & Simulation	Interface Design	Multimedia Scripting	Research Projects	OO Programming	Game and IE Devel.	Modelling & Simulation	OO Programming	Professional Devel.	Multimedia Scripting	Dissertation
Portfolios				□	◆					◆		◆			
Mathematics					□							□			
3D Graphics	□				◆							□		◆	
Simulation and Physics	□			◆	◆					◆	◆				
Artificial Intelligence	◆			◆						◆					
Low-level Programming	□				□					□	□				
Optimization															
Project Management							□			□			◆		
2D Drawing and Painting				□	◆							◆			
Image Creation					◆						◆	◆			
3D Art	□			□	◆						◆	◆		□	
Storyboarding				◆			◆			◆					
Technological Limitations	□				□										
Advice from Industry	□			□						□					
Speakers from Industry										◆					□
Industry projects										□					◆
Industry Case Studies										◆			◆		□
Student Placements															◆
Game Design	□			◆			□			◆					□
User Interf. Design			□			◆						□		□	□
Game creation	◆			◆					□	◆					□
Design Doc. Creation		□	□		□		◆		□					□	□

Table 1. Mapping between criteria for games degree courses suggested by industry (extracted from Skillset documentation) to module content and approaches. (Filled diamonds represent major matching, open squares minor.)

Conclusion

Our progressive approach to development has, in our opinion, so far proven successful. We believe that adopting an iterative approach was crucial to this success. Had we been adopted

a linear development approach then we may have successfully identified the needs of prospective students and employers. However, we suggest that other influences would have jeopardised development. In particular, we possessed a relative lack of subject expertise, we needed to integrate with our other computing courses, conform to the frameworks prescribed by our University, and the size and timescale of the development would not have attracted the support and full participation of staff. As it was we were able to nurture staff expertise, build on staff interest to engender enthusiasm for development, and integrate and conform to frameworks on an incremental basis. The timescale involved has also allowed the team open up a more expansive dialogue with employers and with current and prospective students than would otherwise be possible.

The team has adopted a student-centred approach to delivering the curriculum. In line with our previous experiences this has proven successful in capturing the interest of our students and also in encouraging them to develop their high-order cognitive skills.

Biographies

Colin B. Price is a Teaching Fellow and currently Principal Lecturer at the University of Worcester. His education and teaching experience encompass Experimental and Theoretical Physics, Electronic Engineering and Computing. His interests include non-linear dynamics, self-organizing systems, (particularly in biophysics), computer science education, and the generation of interactive computer art. He has recently led the development of the MSc Computing degree course.

John Colvin is a Teaching Fellow and Senior Lecturer in Computing at University of Worcester, who teaches on computing programming modules. His research interest focuses on Learning & Teaching pedagogy. He has been involved in numerous action-research projects in areas such as e-learning, information literacy, constructive alignment

Warren Wright is Director of Human Resources and Planning at the University of Worcester Business School. His research interests are information systems and knowledge management. He has led a number of course developments including the BSc Computer Games and Multimedia Development.

References

Andersen, P., Bennedsen, J., Brandorff, S., Caspersen, M & Mosegaard, J. (2003) Teaching programming to liberal arts students: a narrative media approach, *ACM SIGCSE Bulletin*, 35(3), pp. 109-113.

Biggs, J. (2003) *Teaching for Quality Learning at University*. Maidenhead: Open University Press.

Carpin,S., Wang J., Lewis M., Birk A., Jacoff A, (2005) High fidelity tools for rescue robotics: results and perspectives.: Robot Soccer World Cup IX, *Springer, Lecture Notes in Artificial Intelligence*.

Colvin, J. & Keene, J. (2004) Supporting undergraduate learning through the collaborative promotion of e-journals by library and academic departments. *Information Research*, **9**(2) paper 173 [Available at http://InformationR.net/ir/9-2/paper173.html]

Colvin, J. (2005) *Teaching Software Development to Non-Software Engineering Students*, Proceedings of the 2005 International Conference on Frontiers in Education: Computer Science and Computer Engineering, Las Vegas, June 2005, pp. 3-9.

Colvin, J. & Phelan, A. (2006) *Evaluating Student opinion of Constructivist learning Activities on Computing Undergraduate Degrees,* Proceedings of the 1st Annual Workshop on Constructive Alignment, Nottingham, February, 2006, pp. 3-14.

Holiday, M. (1995) Incremental Game Development in an Introductory Programming Course, *Proceedings of the 33rd annual on Southeast regional conference***,** South Carolina, March 1995, pp. 170-175.

Kuhn, T. (1977) *The Essential tension: Selected Studies in Scientific Tradition and Change*, The University of Chicago Press

Marton, F. & Saljo, R. (1976), On Quantitative Differences in Learning I. Outcome and Process. *British Journal of Educational Psychology*, 46, pp.4-11.

Neubauer, B. (2002) Lingo and Macromedia Director as Knowledge Management Tools, *Journal of Computing Sciences in Colleges*, 17 (4), pp. 87-93

Price, C.B. (2006) Learning Java Learning Game Programming *Proc. Java in the Curriculum Conference (10) London Metropolitan University.*

Rollings, A. & Adams, E. (2003) *On Game Design*, New Riders Publishing.

Rowntree, D. (1995) Teaching and learning online, *British Journal of Educational Tech*, 26(3),1995, pp.205-215.

Salmon, G. (2000) *E-moderating: The Key to Teaching and Learning Online*. London: Kogan Page.

Salvatierra, R., Weaver, P. & Wilson, S. (2003) *Director MX*. Thomson Delmar Learning: Florence.

Sweedyk, E. & Keeler, R. (2005) Fun and games: a new software engineering course**,** *Proceedings of the 10th annual SIGCSE conference on Innovation and technology in computer science education***,** Caparica, June, 2005, pp. 13 -142.