POWER AND PERIL OF TEACHING GAME PROGRAMMING

Maic Masuch and Lennart Nacke
Games Research Group
Department of Simulation and Graphics
University of Magdeburg
D-39106 Magdeburg, Germany
E-Mail: masuch@isg.cs.uni-magdeburg.de

ABSTRACT

This paper reviews the teaching of computer games as an academic subject for students at university level. The benefits, challenges and problems of teaching game design and game programming are investigated and discussed. We present an overview of experiences with two different approaches to game programming, one in Magdeburg, Germany, and one in Dunedin, New Zealand. A comparison and a list of practical advices for lecturers of similar courses concludes the paper.

KEYWORDS

Computer Games, Games Education, Game Design, Game Programming, Computer Science Curriculum, Video Games

1 INTRODUCTION

Computer games are on its way becoming established in academic research and teaching (Squire 2003). In past years games research was strongly concerned with possible negative effects of computer games on social behaviour of young gamers – as it has been discussed then for many other new media like theatre, movies, television, video and likewise. Recently, the academic focus shifted towards a broader and more balanced argument on games and education, taking also the positive effects of gaming into account.

Simultaneous to this discourse the number of art schools and multimedia colleges offering game design courses increased. These schools, however, concentrate more on practical issues, e.g. teaching primarily the use of tools for game development (Yu 2002). Recently universities have drawn their attention on games and quite a number of researchers have focused on game related technologies. Teaching game development is becoming an increasingly popular subject at universities (Masuch and Freudenberg 2003, McCallum et al. 2004). Students and teaching staff are highly motivated to study a subject which many of them cultivate as a hobby anyway. Also casual gamers and even non-gamers find computer games fascinating as games receive increasing attention and reputation in public.

We present our experiences with two fundamentally different approaches to computer game education and discuss what worked out well and what did not. This text elucidates the benefits and challenges this subject offers for academia.

2 TEACHING GAMES

Teaching the content necessary for the development of games is interesting and difficult at the same time. We found out that, somehow, games are difficult to approach from an academic point of view.

On the one hand this is due to the youth of this research area. Even with the large number of publications that address games very recently, there are no long research and teaching experiences to call on for reference. On the other hand, games research has not yet established itself as an accepted academic discipline among other sciences. Often officials and representatives of more traditional Universities and computer science departments look at this field with reservation and worry about their reputation since games arise directly from the entertainment industry. Furthermore, the deep immersion in games makes it hard to reflect on them at an academic level as most students will be gamers themselves (either casual or hardcore). Nevertheless, universities that teach and research on electronic games, are starting to get a benefit out of it.

Thus, the interest in methods and techniques that investigate game development as a teaching subject has grown, especially at computer science faculties. However, sometimes the term “computer games” is generalised when only talking about specific aspects of programming (e.g. a number of books on DirectX give the impression that they just added a game topic as sales pitch). Somehow similar, for many computer science courses games – as field of application – can provide a very good motivational source. However, in our approach to teaching computer games we aim at teaching the whole process, not just one specialised part or topic

separate from the others. We claim that courses on game development should cover more than just real-time 3D-Graphics. This misuse of terms harms the reputation of computer game development as an academic subject and makes it difficult to establish an academic critique, as there are no agreed terms yet.

1.1.1 Motivation

Game programming and game development are interdisciplinary fields that require educators to go beyond standard topics of computer science. Although we focus on teaching game development at computer science departments, we think it is rewarding to reflect on a humanistic level what your software is capable of doing. Here, games give an excellent bridge for engineers towards reflections on the impacts of their software. Many game developers state that prior to the point of programming a game, relevant factors like game play, player reward and popular culture have to be considered to develop a software product that has some chance to compete on the mass market.

The integration of literally speaking "multimedia" makes games also a very interesting subject because you have to consider compression and media programming issues as you try to deliver the best looking results in the smallest package possible.

Before steeping further into the discussion it seems necessary to clarify the difference between game design and game programming. As these terms are often misused and mixed-up, even by people who implicitly know the very difference, there is some confusion about these two concepts.

1.1.2 Game Design vs. Game Programming

We understand *Game Design* as a creative process of developing a game concept, its core elements and structure. This includes art work and story as well as considerations of playability and game balance. The actual realization of the game, however, can be restricted to a paper document. Strictly speaking computer game design does not require any game programming.

The term *Game Programming* refers to the process of actually coding a game. It consists of making a project plan for the realization of a game idea and actually programming the game elements. This might include the programming of a game engine or "only" implementing the game assets and interaction in an existing game engine.

We use the term *Game Development* to cover Game Design, Game Programming and all other production related topics.

1.1.3 General Approach to Teaching Game Development

When teaching game development at an university level there are different approaches at different institutions around the world, depending on their curriculum. Some are going for a very narrow approach in topics, while others are trying to cover all aspects relevant to game programming in one module instead of a degree program. Of course, many educational facilities that offer a degree in "Game Design" just respond to the demand of their customers – the students. However, from the point of view of the games industry a too general "Game Design" degree programs remain questionable, as those breed all-round-talents instead of specialised team players with deep expertise in their specific domain.

Often it is necessary to make a trade-off between a more technical or more artistic (if not humanistic) side of teaching. This depends on the type of curriculum of the university. Obviously, the main focus of a game course depends on the main teaching direction i.e. programming or design. A good way to address this disjunction is to built interdisciplinary teams. In general, game design does not necessarily require programming skills and can be taught at a beginner level (1st to 2nd year) and game programming as specialization later at an advanced level.

The IGDA suggested a helpful curriculum for game development education (IGDA 2003). It is broad enough for every educational facility to shape their individual "Games"-curriculum – may it be a liberal arts collage or a computer science faculty of a technical university. In the next section we introduce our experiences with teaching games as course modules at two universities.

3 EXPERIENCES IN TEACHING GAMES

We will introduce two different approaches to game development for computer science students at two universities (University of Magdeburg, Germany, and University of Otago, New Zealand) in different countries and social contexts.

1.1.4 Otto-von-Guericke University of Magdeburg

At the Otto-von-Guericke University of Magdeburg the first game programming course was established in 1999 (Masuch 2004). Almost six years of teaching experience later it is interesting to take a look at the development from then to now.

The first try at teaching computer games was to give a very broad and "all inclusive" overview of topics that are used in the world of electronic gaming. So the start was one broad scope course on computer games that featured a programming assignment in parallel with the

lectures. Of course most of the students were very eager to create complex games. Most of them took on a task that was far too large for them to finish within one semester given that they also had other courses running parallel to this one. Some of the projects were never finished – lecturers and students came out of this with lots of experience but also some frustration.

The second run had a new concept. We split the computer games course into two courses Computer Games I and Computer Games II. The first course pursued a broader approach with the techniques on the one side and social/humanistic topics on the other. The second course had its focus on algorithms and tools. Parallel to this second course the students would do group-based project work on a game. We had a large team working on a single project. The most capable and motivated students finally drove it to a successful end. The projects were much more successful and the outcome motivated students as well as lecturers.

The third year included a third course besides CGI and CGII that concentrated solely on game design. Prior to the course, we conducted an informal evaluation of several game design tools and finally used an open source 3D scripting environment. Unfortunately, the team collaboration that worked fine in the evaluation turned out as a catastrophe in real project work. While in the evaluation we managed to integrate our test-scenes into one game-world, the system became unpredictable and unstable dealing with more complex scenes and interactions. It was hardly possible to integrate the work of team members in a single project file and students were more occupied (and frustrated) fighting the tool then to get on with their game. No team managed to conclude their work in a playable prototype. Now, in our sixth year we can look back on developing successful and working curriculum on computer games for computer science and computational visualistics students.

The process of breaking "computer games" down into its subtopics went very well, so it was used again to subdivide topics a little more and focus on specific aspects of computer games in different courses. The several parts of computer game development covered in separate lectures were: "Techniques and Reflections", "Algorithms and Tools", "Modelling and Animation", "Game Design" and other game related lectures, respectively seminars like "educational gaming", "games and simulation", etc.

So instead of trying to put all game-relevant topics into one course, we now provide a range of diverse courses that cover many of the topics that are part of game development.

1.1.5 **University of Otago, Dunedin**

A different approach for a computer game course was co-developed at the University of Otago, New Zealand (McCallum 2004). The primary approach of this course was to introduce students to the concepts of game design in the format of an "all-day" summer school. This is an intense course block format running for six weeks during the summer holidays (four days a week with lectures, labs and tutorials and one day just with guest lectures), the course – a completely different approach from the one in Magdeburg – is not compulsory, which assured beforehand that only interested and motivated students would participate.

Since this was the first and only course of this kind at the University of Otago, the intention of the lecturers was to give a broad overview of the topics covered in the game development process. Thus the course used the IGDA curriculum framework (IGDA 2003) as an outline of topics that were suitable for students at an university level of education. Although the framework is meant as a guideline for an entire degree program on game design, the ambitious plan was to cover all of the topics (on a very shallow basis). The course was co-created by the design and the computer science department.

As the staff at the university had never taught computer game design before it was difficult to evaluate the IGDA framework and tailoring it to the intended audience and timeframe (six weeks). Like the first course at the Otto-von-Guericke University this was an "all inclusive" approach, which tried to cover everything: broad and shallow. Nevertheless the course also included project-based group work within a small timeframe towards a predefined goal – a finished game. One of the unfortunate decisions in the planning was to let the students choose which tools to use for achieving this goal. Thus, the teaching staff had to consider different problems with different tools and even different languages since the students employed techniques from DirectX programming to 3D Game Studio with its embedded C-Script language.

The overall focus on a final product maintained high levels of motivation throughout the course for the students. Unfortunately the time that they needed for programming their game prototypes left little time to reflect enough on social issues and non-programming topics involved in the process of computer game design.

One of the outstanding things in this course was that the students had to give feedback at least every week but were encouraged to keep staff informed every day. Since most of the staff worked in the labs together with the students this concept worked out well at the price of the workload of the staff exceeding even that of the students and spare time shrunk to zero.

A bit surprising (but also rewarding) was the fact that all students finally managed to present some kind of game prototype at the end of the course. Not all of them were of equal quality but most of them were at least somewhat playable.

The range of topics were another problem that required expertise from different departments. Thus, even though the focus was on graphics and programming for games, a large number of lecturers from other departments were engaged to add value from their research areas, which among others included game business, media studies, character animation.

In the end, this overview format was a good way of showing to the computer science faculty how broad the scope of computer games is and how many areas of research can be involved. After the course was finished, the University of Otago also agreed to host New Zealand's first Game Developers Conference. This event has attracted more attention to game development in this small country and lead to the establishment of many interdisciplinary projects.

4 POWER AND PERIL

Based on our experiences we conclude that the teaching of game development can offer a wide variety of benefits. However, it also contains some inherent problems that have to be addressed. In the following sections 4.1 and 4.2 we summarise the common experiences from both approaches. Arguably, taking discussions with lecturers of similar game courses into respect, we think that – to a certain extend – these can be generalised. As the benefits of teaching game design and programming are more obvious than the drawbacks, we will discuss the perils in more detail.

1.1.6 Potential and Benefits

The use of games as an academic topic can have tremendous benefits. Among others, we discovered the following favourable aspects:

Motivation – Students motivation to learn is the key to successful teaching. Games turned out to be a motivational source second to none.

Versality and interdisciplinarity – Games are complex multimedia projects and thus ideally suited to drawing together various aspects of computer science. Further, game development inherently requires input from many disciplines: programming, design, sound, business, and many more. The ability of students to talk to and work with people from other disciplines is an important one in the game production process, both inside and outside of University.

Self-contained projects – The outcome of a course can be a finished product (if the student project is completed). Knowing that they are working towards a product is intensely motivating for students as they can possibly include it in their portfolio and use it for future job applications.

Practical experience – practical experience is given to the students by using industry-like tools and atmosphere, which makes the educational training highly relevant for industry training. After finishing the project work at university this experience can also be used by entrepreneurial students to start their own spin-off company (which has actually happened).

Teamwork – students learn how to work in teams by a group-project based teaching curriculum. In general teamwork is one of the best things to go along with classic teaching, not only because it is a necessary soft skill required by the market but also because it brings social aspects in an otherwise theoretic environment. Having students work together in groups and assigning them to different areas of computer game development allows a high level of social interaction and personal development. On the other hand, this demands, that the educator faces the challenge of grading students individually rather than as a team. This does not need to become a problem as meetings with students can be arranged on a weekly basis, or a feedback system can be established so that it is completely transparent who does which kind of work in the group.

1.1.7 Challenges And Problems

Teaching game courses also comprises some drawbacks that need to be considered:

Claim vs. capability – Students are influenced by existing media. For most students a game is simply a larger software project; while trying to compete with popular game titles they underestimate the amount of work and the capabilities needed for completion by an order of several magnitudes.

Interdisciplinary teamwork – Since games cover a broad range of topics, the audience attracted to a computer games course will eventually attract people from different subjects. This leads to students with very diverse interests and capabilities, which makes it difficult to form homogeneous teams. Therefore it is important that every member identifies himself with the task assigned and the team. Teams should choose a project lead and organise team events.

Missing focus and complexity of projects – In a single course the lecturer has to focus on either game design, game programming or reflections on games not both – or worse – all three. Trying to cover more than one of

these topics in a single course, can lead to a watering down of the content of any one topic. Regarding practical exercises, it is futile trying to implement all aspects of a game (or even a game engine). Again, focus should be set on specific aspects of a game.

No experience and use of wrong tools – In comparison to other Computer Science subjects there is little experience in teaching and programming games to build upon. Many course topics come from other areas of expertise (e.g. psychology, pedagogy, drama theory, law, design, fine arts etc.) so that lecturers have to get assistance from other departments (good for interdisciplinary cooperation, bad for the time table) or have to research all relevant information on their own. The workload for this is much higher than in classical sciences. Further, it is futile to build a game and a game engine from scratch. Finding adequate tools that are suitable for teaching the specific course aspects (even focusing on game design or game programming) can be difficult. Our recommendation is to begin software evaluation at least one semester early, preferably with a well documented prototype.

5 CONCLUSION

Summing all up, we think that computer games have an enormous potential in teaching and research. Our experiences in the last years with teaching game programming to computer science students showed to us that games are exceptionally well in

- *motivating* students,
- teach them *interdisciplinary* teamwork and
- give them hands-on experience in *multimedia integration*.

After having analysed these two different approaches to the teaching of computer games and the difficulties that they were facing, some final conclusions can be drawn. The approach of the University of Otago to teach a complex subject like this in summer school requires full-time availability of students and lecturers. It is a suitable course format to emulate the "crunch period" similar to the game industry. The amount of work necessary for preparing the lectures and the course was underestimated. More time would have been helpful. In addition, the University of Otago did only have limited resources to provide such a course.

The prerequisites in Magdeburg now are much better, but it all started very similar to the approach made at the University of Otago. The games research group is now an established part of the faculty and therefore can allocate more resources on teaching and research. A number of lectures have been developed, each with a focus on a different aspects of computer games. Both directions are unwearyingly supported by students while undergoing a practical software lab or a lab internship. Our start-up phase is over now, but there is still much room for improvement.

Finally, we would like to share some practical hands-on advice for people who are planning similar courses:

- Focus on either game programming or game design, not both.
- Ensure equal levels of programming experience and capabilities.
- Students should no start to learn tools whilst in production – use crash courses in advance of the lecture for preparation.
- Do not allow different platforms, engines, languages or tools – students should help each other.

And finally – if you only take one advice from this paper – then the most important one: Have fun!

ACKNOWLEDGEMENTS

We would like to thank all our students who made teaching such a rewarding effort and our student advisors: Bert Vehmeier, Ralf Armin Böttcher and Jan Fietz for their help in Magdeburg. We would also like to thank Simon McCallum for his personal devotion and optimism of starting a game programming course from scratch in New Zealand.

REFERENCES

IGDA Education Committee, 2003. IGDA Curriculum Framework - The Study of Games and Game Development, Version 2.3 beta. February 25, 2003.

McCallum, S., 2004. COSC360 - Computer Game Design at the University of Otago, Dunedin, New Zealand: http://cosc360.otago.ac.nz

McCallum, S.; Makie, J. and Nacke, L., 2004. Creating a Computer Game Design Course. In *Proceedings of the New Zealand Game Developers Conference, (NZGDC).*

Masuch, M., 2004. Computer Games Research Group at the University of Magdeburg: http://wwwisg.cs.uni-magdeburg.de /games

Masuch, M. and Freudenberg, B., 2002: Teaching 3D Computer Game Programming. In Ralf Dörner, Christian Geiger, Paul Grimm, and Michael Haller, editors, *Workshop Proceedings Production Process of 3D Computer Graphics Applications -- Structures, Roles, and Tools*, Aachen, Shaker.

Squire, K., 2003: Video games in education. *International Journal of Intelligent Simulations and Gaming (2) 1.*

Yu, C., 2002: Developing a Game Programming Course For Computer Science Majors In a Liberal Arts College. In *First*

International Conference on Information Technology & Applications, (ICITA).

POWERS AND PERILS OF TEACHING GAME PROGRAMMING

Maic Masuch and Lennart Nacke
Games Research Group
Department of Computer Science
University of Magdeburg
D-39016 Magdeburg, Germany
E-mail: masuch@isg.cs.uni-magdeburg.de

ABSTRACT

This paper reviews the teaching of computer games as an academic subject for students at university level. The benefits, challenges and problems of teaching game design and game programming are investigated and discussed. We present an overview of experiences with two different approaches to game programming, one in Magdeburg, Germany, and one in Dunedin, New Zealand. A comparison and a list of practical advices for lecturers of similar courses concludes the paper.

KEYWORDS

Computer Games, Games Education, Game Design, Game Programming, Computer Science Curriculum, Video Games

BIOGRAPHY

Maic Masuch, PhD in computer animation, graduated at University of Magdeburg, Germany where he is now Germany's first professor for computer games. He has been teaching and researching on computer game programming for six years.
Research focuses on authoring of virtual worlds, user interfaces, audio-only-interfaces and graphics for games, especially real-time non-photorealistic rendering techniques for game engines. He supervised several game-related student projects and works as a game consultant for game development companies and the German ministry of research. In addition, Prof. Masuch is co-founder of Impara, a technology think tank that is developing media systems for playful learning.

Lennart Nacke, Student of computational visualistics at the Otto-von-Guericke University Magdeburg, Germany (Vordiplom 12/02), internship at the University of Otago, Dunedin, New Zealand (09/03-02/04), student research assistant at the Department of Simulation and Graphics, member of the student council of the Department of Computer Science, IGDA student member since 01/04. Experience as a tutor for game design courses in New Zealand and Magdeburg. Research interest lies in game design and game programming.