Experiencing the engineering challenges of game development

Rafael Bidarra

Faculty of Electrical Engineering, Mathematics and Computer Science
Delft University of Technology
Mekelweg 4, NL-2628 CD Delft, The Netherlands
R.Bidarra@ewi.tudelft.nl

Abstract

Seven years after the introduction of project-based Computer Science (CS) education at Delft University of Technology, its highly instructive and motivating potential has been more than confirmed. Among the projects offered, the second year *Games project* has justly become the integrator course *par excellence* of the Computer Science BSc curriculum. More importantly, the Games project brings CS students for the first time to work in a realistic and interdisciplinary game development team, involving fellow students pursuing a Game Design and Development degree at the Utrecht School of the Arts.

The reputation and popularity of the Games project are being effectively exploited by the Faculty for the urgent purpose of recruiting new CS students. We describe several project features that we have developed and exploited throughout the years for these purposes, mostly in close collaboration with older CS students and alumni, and with experts from various Dutch game development companies. Among other goals, we seek that candidate students experience the contribution of the various disciplines involved in the development of a computer game, and acquire a no-nonsense view on the surrounding game industry. So far our efforts have been rewarded with a steady trend of an increasing yearly enrolment.

Keywords: project-based education, interdisciplinary education, games in education

1. INTRODUCTION

Project-based education is very well suited for applying the basic concepts behind *constructive alignment* [3], an increasingly influential stream in higher education which advocates among other things that 'students construct meaning from what they do to learn'. Consequently, an advantage of including such projects in a curriculum is that the acquisition of knowledge is strongly motivated by its immediate application in a practical environment. In addition, it encourages students to actively learn to value and promote the teamwork *process*, instead of focusing exclusively on the final *product*.

Seven years ago, Delft University of Technology introduced project-based education in the Computer Science (CS) undergraduate curriculum and, since then, its highly instructive and motivating potential has been more

than confirmed. Characteristic for CS project courses is the fact that students have to work in groups on a more or less open assignment; see [1] and [11] for some examples. One of the projects offered to our second year students is the *Games project*, in which they design and implement a computer game from scratch, using the technology provided, working in a team. This project has now achieved a substantial maturity, so much so that, by covering so many aspects of game development, it now rather closely reflects real-life game development environments. Understandably, its growing reputation and popularity are being justly and very effectively exploited by the Faculty for the urgent purpose of increasing enrolment of new CS students. In doing this, we are encouraged by the growing evidence that games projects like this are often instrumental in reverting the CS decline of the last decade [6].

In this paper, we first summarise some relevant characteristics of the Games project (Section 2) with an emphasis on its key interdisciplinary character (Section 3), in order to discuss several project features that we have developed and exploited throughout the years for recruiting purposes (Section 4).

2. PROJECT CHARACTERISTICS

The pedagogical background and organisational setup of the Games project have been extensively described in [2], including the development of its increasingly ambitious learning objectives (focusing e.g. on media, programming and soft skills), the project assessment mechanisms (featuring e.g. several peer evaluation cycles), and several crucial planning aspects (as e.g. the deployment of agile programming). Here we simply highlight a few salient features, due to either their novelty or their particular contribution to the aforementioned recruiting purposes.

It is a fact that the Games project has naturally evolved into the integrator course *par excellence* of the Computer Science BSc curriculum at Delft University of Technology. It integrates a broad range of CS topics, ranging from computer graphics (CG) and software engineering to user interaction and artificial intelligence (AI). Furthermore, it also serves as an attractive arena for successfully applying knowledge acquired in other courses, including the often problematic algebra and calculus.

Unlike all other CS projects courses, which take one trimester (typically 3-4 ECTS credits), the Games project presently spans over a full semester (7 ECTS credits). This gives us margin for a more careful and gradual structuring which has included, from the very beginning, a number of guest lectures strategically placed throughout the semester. Among the speakers, we always schedule a number of experts from renowned Dutch game developers, who tell about their experiences on games development, from a wide variety of viewpoints, ranging from design methodologies, through current CG or AI challenges, to commercial video game production and market aspects. Sometimes, the guest speakers are alumni who have formerly attended that same project, and later found a job in the game industry. In any case, involvement of real stakeholders from the games industry has not only enriched the project with significant technical expertise, but it has been an important success factor because it strongly stimulates and motivates students.

Another much valued project feature is the huge freedom we always give the students in order to conceive *their own game*. Typically we provide an *inspiring list* of generic requirements (not only relating to 3D world, navigation or score mechanism, but also regarding CG and AI techniques), and ask students to select a *fixed* amount of requirements, out of that list, which their game will have to fulfil. In practice, by offering a wide choice among many game-related techniques, we guarantee that for every student there are always enough challenges to explore. This, in turn, encourages students to remain motivated, to delve deeper into whatever study subjects required, and to exceed themselves in the implementation of the techniques of their choice.

Throughout the years, the technology deployed in this project has evolved a lot, and the development framework used has been continuously improved. We started using C++ and the Ogre engine [8], and are presently using C# and XNA [12], which has the advantage that the games developed run both on a PC and on an Xbox 360 game console. Along the way, we have also deployed Cannibal Experience [4], a rather innovative game development environment specially tailored for higher education purposes, conceived by Cannibal Game Studios [5], a spin-off company set up by former students of this project. Currently we are using the open source collaborative framework DrProject for integrating the various support tasks of project teams [10]. Among other features, it deploys, for each team, a wiki (essential for dynamically keeping track of ever-changing project documentation), a versioning system (indispensable for code, art and documents), a Trac-based planning tool (a very accessible and instructive way of learning to work effectively) and a mailing facility (instrumental for proper group communication and management).

3. INTERDISCIPLINARY CHARACTER

Initially, the Games project was run in groups of about 5-7 CS sophomores, who had to handle alone *all tasks*, from game design, through game art to game implementation [1]. This was a double nuisance, because (i) the skills required for game design, concept art and modelling work, mostly absent in their educational curriculum, were more often than not disappointing (to say the least), and (ii) the time wasted in that struggle distracted students from their core business as programmers, for which they were indeed to be assessed.

Around that time, we got in contact with the Utrecht School of the Arts, which offers a BA degree on Game Design and Development (GDD). Their second year students also had a one-semester project, focusing on the game design process as a whole, which was motivated by a similar goal of integrating all their other courses (game design, ludology, modelling, animation, audio, etc.). Interestingly enough, the reverse conflict of our CS sophomores was being experienced by their GDD mates: all too often, their very creative and original art work, game designs and concepts could hardly make it through a rapid prototyping phase, in the absence of enough programming skills in the team; and this was again a pity and a shame.

Realising this blatant complementarity, we started a pioneer collaboration with the Utrecht School of the Arts, integrating their game design project with our game development project, leading to one large multidisciplinary game project. Typically, this is the first time in their lives that students from two such different schools have to work together in a realistic and interdisciplinary game development team.

In this integrated project, teams consist of 3-4 CS students and 5 GDD students. CS students are mainly responsible for the implementation of the game, while GDD students are in charge of game design and artwork/content creation; in doing this, they work as two departments of 'one single company', with lead programmer and lead designer roles, respectively, assigned among them. The role of producer is typically assumed by a teaching assistant, who oversees the whole process and watches over the communication and planning.

Integrating these two projects brought much more realism and power to the project: realism, because it more closely matches the actual team composition in a real-world game development company; power, because this interdisciplinary collaboration promotes that each team member contributes with his/her best skills to the common goal. In other words, we fully confirmed the value of the motto left by the late Randy Pausch to his Entertainment Technology Center: "Not to turn artists into engineers or vice versa, but to teach students how to work in teams that utilize the disparate talents of their members" [9].

These mixed teams, though having clear advantages over traditional uniform groups, also have some disadvantages; for example, more time is spent on communication, travelling and a variety of appointments. In particular, everyone in these teams vividly experiences the additional challenges brought about by communicating with people from outside your own discipline, which requires a rather different way of thinking and explaining.

In order to help overcoming these vicissitudes, one of the mechanisms we came up with is the so-called *kick-off workshop*, a sort of 'Game Jam' that spans the whole first week of the semester. Basically, we bring all (CS and GDD) students together on Monday, assign them to groups, and give each team the 'impossible assignment': make an Arcade game until Friday, using the technology and the collaborative environment provided (see Section 2).

The advantages of this 'therapy' for each student are countless. First, under this time pressure, they quickly move on to form a solid team, get to know their *other* team mates, and experience how different their language and mind-set are. Sure enough, this quickly leads them to two crucial experiences, often competing in intensity: the *mutual wonder* for the skills of their *other* mates, and the occurrence of the first *arguments*. Eventually both effects contribute to strengthen the team and, typically, as they reach the Friday deadline and present their game in a plenary session, they touch the bewildering magic of a synergy that has just dawned. Of course, another amazing side-effect is that, in one week, everyone painlessly learns to use the collaborative tools and the technology needed for the rest of the semester.

4. CLOSING THE LOOP

Along the years, we have often wondered about how to capitalise on the success and popularity of the Games project, and in general, of the game technology research work being done at our faculty. When it comes to the critical goal of recruiting new CS students, we call that process 'closing the loop', because it indeed allows us to feedback all those positive and alluring results to the younger generations, in order to maintain, and hopefully increase, the input stream that will keep it alive.

Firstly, our University organises twice per year an Open Day programme, aimed at interested high-school candidate students and, naturally, computer games are always present in one way or another in these events, as one of the faculty's 'choice subjects'. So far we have done that in a variety of flavours, but clearly the most successful consists of getting junior students actively involved, for example giving demos of the impressive games *they* developed in the year before during the Games project. Our constant experience is that, in this role, whatever they show, do or say, students are the best ambassadors you might possibly desire for CS recruiting.

Another very convenient advantage of the freedom of choice given in the Games project, pointed out in Section 2, is that throughout the years you hardly have two similar games: students make it their point that they have chosen to build their own, *unique* game, and not for example 'yet another first-person-shooter'. This in turn makes the website of the Games project [7] a very pleasant and surprising repository: there is a story and a group of people behind each game in there, which is *obviously* totally different from any other. And because all games are there available for download, it is no wonder that invariably, after Open Days, the amount of visitors noticeably increases.

Occasionally, talks have been scheduled in such Open Day programmes around some game related issues as, for example, identifying and matching the variety of courses in the CS curriculum with the list of subjects involved in the making of a game. Another rather successful talk involves building a Pacman game in real-time (30 minutes), culminating with a demo, again by a former Games project student, of his own magnificent 3D (sort of) Pacman.

The final presentation of the Games project, at the end of the semester, is now becoming a one-of-a-kind event on campus: CS and GDD students together organize an exhibition, including posters, game trailers, and of course, the chance for the public to get their hands on the newly developed games. See the figures on the cover page, for some samples of the 2008 edition.

Usually, that is also the day of announcing the winners of the *Game of the Year* competition, an exciting contest 'unofficially organized' every year in our faculty among the participating teams. The basic idea is that we invite game development companies to sponsor the competition, by providing both a jury member and a prize for the winning team. This scheme not only offers those companies a chance to promote their games, but above all it helps them get acquainted with the best skills of our best students.

No wonder that as a result of this close collaboration with the game industry, over the years many former students of the Games project have found either their BSc internship or their final MSc research project at selected Dutch game development companies, in a variety of original and advanced research topics (e.g. multi-core path-finding, GPU-based terrain generation, procedural generation of serious game levels, or enhancing game characters with emotions). It goes without saying that eventually numerous CS alumni either have found their career in one of the various game development companies they met during this project, or established their own start-up companies in the field, supported by our university's start-up incubator; in both cases, they gladly maintain a close collaboration with us.

As a final balancing remark, in all student recruitment activities described in this section, we put a particular emphasis in making clear to interested candidates that there is no such thing as an easygoing CS (or, for that matter, engineering) study programme, and certainly not at Delft University of Technology. The main message passed is unmistakable: our CS curriculum is definitely not an easy one, there is a huge lot of hard work hiding in its project-based courses, and that is particularly true of the Games project. For us, it is a matter of integrity to be very unambiguous about the degree of difficulty in our study programme; we believe that consistently doing this is, paradoxically, one of the reasons why more and more young students consciously enrol in our CS programme.

5. CONCLUSIONS

We briefly presented the main and most recent highlights of the Games project, the integrator course *par excellence* of the Computer Science undergraduate curriculum at Delft University of Technology.

In our experience, the key to the huge attractiveness and success of the Games project lies in the consistent combination of its careful interdisciplinary organization with the deployment of a fine-tuned working environment, including proven technology. In this setting, we came to conclude that a streamlined collaboration among students of related disciplines always works as a very powerful catalyst.

Ultimately, with all recruitment initiatives described here we strive to achieve that candidate CS students (i) vividly experience the contribution of the various disciplines involved in the development of a computer game; (ii) learn to appreciate work in such interdisciplinary teams; (iii) acquire a no-nonsense view on the game industry, and (iv) increase their awareness of the importance of games in modern society. So far our efforts have

been yearly rewarded with an steady trend of both an increasing CS enrolment and a growing amount of MSc students graduating in game technology projects.

Games are, and have always been, all about fun; still, it is amazing that getting students in the position of making games can be even more fun. But the most fortunate of them are those who realize how much they have learned in that process.

6. AKNOWLEDGMENTS

We are very grateful to each and every student of the Games project, for all their patient and invaluable feedback throughout the years, and to all colleagues who contributed to the success of this project with their constructive ideas and criticism; this, of course, includes the fantastic horde of teaching assistants whom we have been fortunate enough to work with. Finally, special thanks go to all our colleagues of the Dutch game industry, for their enthusiast and supportive involvement, both in the Games project and in our ongoing research work.

REFERENCES

[1] Bidarra R, van Dalen R, van Zwieten J (2003) A Computer Graphics pioneer project on computer games. Proceedings of CGME 2003 – Workshop on Computer Graphics, Multimedia and Education, 8 October, Porto, Portugal, pp. 61–65

[2] Bidarra R, Boers J, Dobbe J, Huijser R (2008) The making of an interdisciplinary games project. Journal of Game Development, 3(2): 7–25

[3] Biggs J (1999) Teaching for quality learning at university. SRHE and Open University Press, Buckingham, UK

[4] Boers J, Dobbe J, Huijser R, Bidarra R (2008) From a light CG framework to a strong Cannibal Experience. In: Proceedings of Eurographics 2008 - Education papers, 14-18 April, Crete, Greece, pp. 149-153

[5] Cannibal Game Studios (2008) http://www.cannibalgamestudios.com/

[6] Estey A, Gooch A, Gooch B (2009) Addressing industry issues in a multi-disciplinary course on game design. Proceedings of Fourth International Conference on the Foundations of Digital Games, 26-30 April, Port Canaveral, FL

[7] MKT4 (2009) MKT4 – The *Games Project* website. Delft University of Technology. http://graphics.tudelft.nl/~mkt4/

[8] OGRE (2008) http://www.ogre3d.org/

[9] Pausch R, Marinelli D, (2007) Carnegie Mellon's Entertainment Technology Center: combining the left and right brain. Communications of the ACM, 50 (7): 50–57

[10] Reid KL, Wilson GV (2007) DrProject: a software project management portal to meet educational needs. Proceedings of SIGCSE'07, 7–10 March, Covington, Kentucky, ACM Press, NY

[11] Schaefer S, Warren J (2004) Teaching computer game design and construction. Computer-Aided Design 36 (2004): 1501–1510

[12] XNA (2009) http://msdn.microsoft.com/xna/