Interdisciplinary game projects: opening the Graphics (back) door with the soft skills key

Rafael Bidarra[1]

[1] Computer Graphics Group, Delft University of Technology, The Netherlands

Abstract

Project-based Computer Science (CS) education was introduced at Delft University of Technology more than 10 years ago, and its instructive and motivating potential has steadily increased. Among the projects offered, the second year Games project *has justly become the integrator course* par excellence *of the Computer Science BSc curriculum. Recently, a pioneer, campus-wide course* Building Serious Games *followed up on that success, making MSc students face the new and exciting challenges of serious game development. More importantly, these projects bring most students for the first time to work together in a realistic and interdisciplinary game development team, involving fellow students pursuing a variety of other degrees, at other faculties or even at other schools. Since we set up such collaborations, all parties have witnessed a considerable leap forward in their students' soft skills, including communication, self discipline, mutual appreciation and team management.*

We describe several game project features that we have developed and exploited throughout the years, often in collaboration with experts from numerous Dutch game studios. Among other goals, we seek that our students experience the contribution of the various disciplines involved in the development of a computer game, and acquire a no-nonsense view on the real *game industry, which turns out to value more the personal soft skills of new applicants than their elaborate graphics programming abilities. Our experience is that a streamlined collaboration in interdisciplinary teams is a very powerful catalyst that significantly raises the proficiency level achieved by students of all curricula. As to our own CS students, the volume and depth of the computer graphics expertise acquired in this process is incomparably higher than we could possibly have expected.*

Categories and Subject Descriptors (according to ACM CCS): K.3.2 [Computers and Education]: Computer and Information Science Education – *Computer science education, Curriculum*, K.8.0 [Personal Computing]: General – *Games*

1. Introduction

In the year 2000, Delft University of Technology introduced project-based education in the Computer Science (CS) undergraduate curriculum. Since then, despite all curriculum upgrades and reforms, the highly instructive and motivating potential of project-based courses 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.

In the last decade, there is an increasing trend of using game assignments in a variety of flavours for such purposes. Sometimes, these projects have a very clear CS-learning character, e.g. programming basics [Hic10], software engineering [NB10] or agile programming [SWM10]. In other projects, a stronger game design component has been expressly sought, as for example those described in [EB09] and [EGG09].

Yet other projects have had an explicit computer graphics (CG) motivation from the outset, as for example the courses described by [SW04] and [STG08]. Also our own trajectory has evolved from very focussed CG-oriented projects, first described in [BDZ03], up to projects covering an increasingly wide game technology spectrum, on which we reported e.g. in [BDHB08]. Both the pedagogical background and the organisational setup of those projects have been extensively described in [BBDH08a], to which we refer the reader for a thorough discussion on, among other issues, the development of their increasingly ambitious learning objectives (focusing e.g. on graphics, media, programming 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). Therefore, in this paper, we omit most of that discussion, simply highlighting a few of the more recent or salient such features.

In the last five years, we have been paying more and more attention to the pedagogic working of interdisciplinary teams. This was increased, among other things, by amazingly effective experiences like those at CMU's Entertainment Technology Center, reported by Randy Pausch and Don Marinelli [PM07] [Mar10]. This led us to further expand the scope of our learning outcomes with a spectrum

of teamwork and interpersonal skill, often also designated *soft skills*.

Here, we will concentrate on such interdisciplinary aspects as implemented in two of our courses, due to their strong connection to computer graphics education: the Games Project (GP) and the Building Serious Games project (BSG). In the GP, offered to our second year undergraduate CS students, 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; see e.g. [BBDH08b]. The BSG project, in contrast, has been taught for just two years. It is a pioneer, campus-wide MSc course, and it very much capitalizes on our experience with the GP, although exhibiting some clearly unique features, as we will describe in the next section.

In this paper, we first summarise the most relevant characteristics of both game projects (Section 2) including the technological aspects (Section 3). We put a special emphasis on their key interdisciplinary character, in order to discuss several project features that we have developed and exploited throughout the years (Section 4). Finally, we describe also some 'decoration elements' that are instrumental for the motivation of our students (Section 5) and draw some conclusions (Section 6).

2. Salient characteristics of the game projects

In this section we summarize for both projects the main features that are relevant for our topic.

2.1 Games Project (GP)

The GP has naturally become 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 CG and software engineering to user interaction and artificial intelligence (AI). Moreover, it provides a very attractive arena for successfully applying knowledge acquired in various other courses, including the often problematic algebra and calculus.

The GP presently spans the full spring semester (7 ECTS credits). This allows us to schedule a number of guest lectures strategically spread throughout the semester. Among the speakers, we always invite a number of experts from renowned Dutch game studios, who vividly tell about their real-life games development experiences, from a wide variety of viewpoints, ranging from design methodologies, through commercial video game production and market aspects, to current CG challenges. This last year, for example, we were happy to provide a vintage selection of topics, including *Real-time ray tracing*, *Procedural modelling*, and *Shader programming primer*. 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 not only enriches the project with significant technical expertise, but it is an important success factor because it strongly stimulates and motivates students.

One of the most valued project features is the large freedom we always give the students in order to conceive *their own game*. Typically we provide a non-exaustive list of generic requirements (not only relating to 3D world, navigation or score mechanism, but particularly 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 subjects are required, and to excel themselves in the implementation of the techniques of their choice.

In our experience, the above is particularly true of most CG topics: when the 'direct necessity' of some solution is highly felt, there is virtually no obstacle that can hold them back, no matter how difficult the topic might be. In recent years, for example, we have seen students (i) spending lots of hours researching and developing advanced shader programming and complex collision detection algorithms; (ii) studying quaternions and other non-trivial mathematical methods that seemed very suitable to model some game situation; (iii) endlessly refining their modelling, lighting and rendering techniques in order to achieve satisfactory performances; and (iv) coming up with many creative methods for procedurally generating a large variety of content.

It is therefore our more than confirmed conclusion that learning CG techniques in context, and applying them within an attractive and motivating project, can outweigh the known difficulties and hurdles inherent to some CG concepts, and is at least as important as the careful choice and chaining of learning modules.

2.2 Building Serious Games project (BSG)

As mentioned above, the BSG project has been taught in the last two years. It is an MSc project, takes only one trimester (8 weeks), during which students are asked to design and implement a *serious game prototype*. Due to the short course duration, we keep guest lectures to a minimum, in order to avoid unnecessary dispersion. We do, however, schedule an *intensive crash-workshop* on serious game design in the first week, for obvious reasons.

The main goal of the project is to take students with varying talents, backgrounds, and perspectives and put them together to do what none of them could do alone: to implement a serious game prototype aimed at being applied in a real-world setting (educational, social, professional, etc.). In other words, the interdisciplinary character and the soft skills content in this course's learning outcomes is even larger than with the GP.

Assignments for this course are commissioned by real-world companies or end-users, which we seek from within our network. In the Netherlands, this is not as difficult as it might seem, as there are plenty of potential partners with a strong interest in investing in serious game applications (including marketing, educational, civic or awareness purposes). We do require a full commitment from commis-

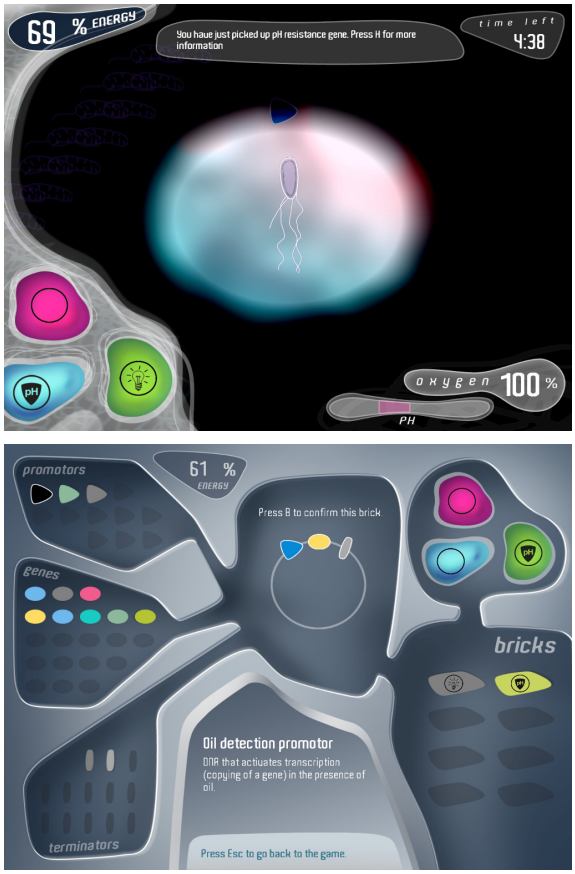

Figure 2: Pimp up your bacterium*, a serious game in which children learn synthetic biology concepts while driving a bacterium through adverse obstacles (above), in order to acquire the necessary enhancements that enable it to neutralise the Gulf oil slicks (below).*

sioning partners, in terms of dedication, feedback and assessment, even though the serious game delivered is simply a prototype and not an end-product.

In contrast to the GP, the emphasis of this project is on sensibly deploying *whatever* game technology is adequate to constructively fulfilling the game requirements. This may well include unconventional interaction, apparently primitive technologies, 'original' user interfaces, etc. Therefore, a predominant role is played by solving the various, possibly conflicting, technical challenges while watching over the fundamental game requirements.

To make this point, we give two recent examples of serious games made during this project. Both games have children as their target public, but with rather different ages and goals. The first game, *Pimp up your bacterium* (see Figure 1), aims at familiarising high-school pupils with the basic concepts of synthetic biology. It was developed in collaboration with the iGEM TU Delft team, which eventually became an award-winning finalist in the iGEM world championship held at MIT, Boston, USA. In the game, you play a bacterium that starts out small and without any interesting abilities, but quickly has to build up several useful skills. In one of the game levels, for example, you have to overcome hardships in the form of antibi-

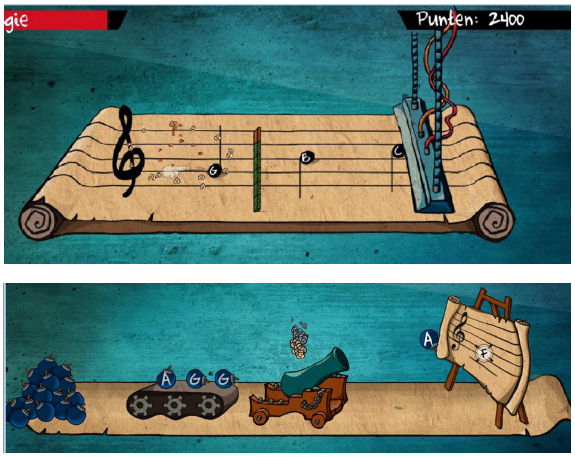

Figure 1: Melosion*, a 'serious shooter game' in which children learn to identify (above) and place (below) music notes on a score.*

otics, dangerous concentrations of alcohol, and even viruses, and put your newly acquired abilities to good use by neutralizing oil slicks.

The second game, *Melosion* (see Figure 2), aims at teaching young children the place of musical notes on a score. For this, they have to chase the notes side-scrolling on the score, by jumping from line to line, while they listen to a familiar melody. Another game level features a cannon which you have to carefully aim, in order to shot each cannon ball (a musical note) back to its correct location on the score. An ingenious Wii-controller interface was devised, that also allows for a very competitive multiplayer version.

3. Game technology

Throughout the years, the technology deployed in these projects has evolved a lot, and the development framework used has been continuously improved.

We started the GP, in 2003, using C++ and the Ogre engine [Ogr10]. Later on, we moved to C# and deployed Cannibal Experience [BDHB08], an innovative game development environment specially tailored for higher education purposes, conceived by Cannibal Game Studios, a spin-off company set up by former students of this project. In the last three years, we have been using C# and XNA Game Studio [XNA10], which has the advantage that the games developed run both on a PC and on an Xbox 360 game console.

Lately, we decided to open up the choice of game engine (including Ogre, XNA, jMonkey and Panda3D) and even the programming language to use (including C, C++, C#, Java and Python). Still, the vast majority of undergraduate students (GP) chooses for 'the safety' of XNA, while MSc students (BSG) mostly make their choices based on pragmatic criteria mentioned in Subsection 2.2. The bottom line for us is: as long as we have expertise in-house to assist them in the chosen technology, their choice is acceptable. In any case, all students very much appreciated this additional freedom.

Currently we are using the open source collaborative framework DrProject for integrating the various support tasks of project teams [RW07]. 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). So far, this light but very convenient online framework has proved to be quite useful to our teams.

4. Interdisciplinary features

In this section, we describe the key interdisciplinary features of these projects, highlighting how they enhance students' soft skills, and why they so strongly improve the learning process.

4.1 The history

The GP was initially 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 [BDZ03]. 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 tasks as programmers, for which they were 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.

4.2 Games Project

In this integrated form, GP 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 the 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" [PM07].

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 3).

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.

4.3 Building Serious Games project

The BSG project is explicitly open to all interested students from *any* MSc program of *any* other faculty, which as far as we know is unique in project courses at our University. So, as a specifically desired methodology, all students work in interdisciplinary teams, under the supervision of the respective company/end-user to whom the team reports throughout the term of the project. Together with the commissioner, the team will identify the serious game requirements, and quickly converge to a first game concept.

Figure 3: *Screenshots and concept art from the games produced in the Games Projects of the last three years*

At the same time, the team has to decide and watch over the project work plan in order to produce the scheduled deliverables and achieve the proposed goals in due time. Of course, this includes assigning and personally assuming the roles that better suit each of the team members. Typically, CS students will carry (most of) the implementation burden, whereas students of other faculties will mostly fulfil other crucial roles, including designer, producer, artist, modeller, tester, etc.

Being composed of graduate students, these interdisciplinary teams are in a better position to efficiently collaborate than their undergrad mates in the GP. Also, many of the CS students attending the BSG project have the prior experience of having followed the GP, which really helps set the tone.

5. The 'essential' decoration

Along the years, we have often wondered what else can we do around these courses to stimulate the motivation of our students, and help them overcome the more obstinate barriers, technical or otherwise. Our best answers to this mostly orbit around three personal concepts: *appreciation, exposure, reward*. We just mention a few of the initiatives taken in this area:

- Our University organises twice per year an Open Day programme, aimed at interested high-school candidate freshmen 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 our junior students actively involved, for example giving demos of the impressive games *they* developed in the year before. 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, instead of 'yet another first-person-shooter'. This, in turn makes the website of the Games Project [MKT11] 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.
- The final presentation of the Games Project, at the end of the Spring semester, is now becoming a one-of-a-kind event on campus: students organize together an exhibition, including posters, game trailers, and of course, the chance for the public to get their hands on the newly developed games. See Figure 3 for some samples of the last few years.
- Usually, that is also the day of announcing the winners of the *Game of the Year* award, an exciting contest 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 an award 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

Figure 4: In *the Nuna Evolution game, as currently set up at the TU Delft Science Centre, a team adjusts the design of their race car while they are racing on the track. To further motivate players, we used the model of the Nuna Solar car, the award-winning solar vehicle designed at TU Delft.*

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.

- Finally, we were fortunate enough to have a very enthusiast and supportive Science Centre at our University [Sci11], which made it a point to 'acquire' the serious games developed in the BSG project, and include them in the permanent exhibition of their Game Lab. As a result, those games, perfectly identified with full names of all team members, have been played and admired by thousands of visitors. A prime example is the serious game Nuna Evolution (see Figure 4), a two-player serious game in which your co-pilot can modify, on the fly, as s/he pleases, the design of your car while you are driving (including the type of propulsion, wheels and material used). The game very successfully creates awareness on the various effects (e.g. environmental, economical, maintenance, performance) of the players' choices.

6. Conclusions

We discussed the most salient highlights of two CS game projects at Delft University of Technology: the Games Project, which is now the integrator course *par excellence* of the Computer Science undergraduate curriculum, and

the Building Serious Games project, a new interdisciplinary course open to students of all faculties. Both projects have one key feature in common, a.k.a. *headfake* [PM07]: that students acquire a very considerable amount of expertise, very particularly in computer graphics, while they are intensively 'occupied with something else'...

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

With this approach, we seek that 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.

Several recruitment initiatives at our faculty capitalize both on the popularity of this area and on the success of these projects. So far our efforts have been yearly rewarded with a steady trend of both an increasing CS enrolment and a growing amount of MSc students graduating in game technology projects. In this regard, the future certainly seems both bright and fun for our faculty; but this is especially true for those fortunate students who realize how much more they have learned in this process.

Acknowledgments

We are very grateful to each and every student of our game projects, for all their patient and invaluable feedback throughout the years, and to all colleagues (from both Delft and Utrecht) who contributed to the success of these projects with their constructive ideas and criticism; this, of course, includes a horde of creative teaching assistants whom we have been fortunate enough to work with. We also thank all our colleagues of the Dutch game industry, for their enthusiast and supportive involvement, both in our game projects and in our ongoing research work. Finally, special thanks to everyone at the TU Delft Science Centre, for their encouragement and supportive commitment to disseminate the results of the Building Serious Games project at the Centre's Game Lab.

References

[BBDH08a] BIDARRA R., BOERS J., DOBBE J., HUIJSER R. The making of an interdisciplinary games project. *Journal of Game Development*, **3**(2), 2008

[BBDH08b] BIDARRA R., BOERS J., DOBBE J., HUIJSER R. Bringing a pioneer games project to the next level. Proceedings of GDSCE 2008 - Third International Conference on Game Development in Computer Science Education, February 28-March 3, pp. 11-15 doi: 10.1145/1463673.1463676

[BDHB08] BOERS J., DOBBE J., HUIJSER R., BIDARRA R. From a light CG framework to a strong Cannibal Experience. Proceedings of Eurographics 2008 - Education papers, 14-18 April, Crete, Greece, pp. 149-153

[BDZ03] BIDARRA R., VAN DALEN R., VAN ZWIETEN J. 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

[EB09] EAGLE M., BARNES T. Evaluation of a game-based lab assignment. Proceedings of FDG 2009 - Fourth International Conference on the Foundations of Digital Games, 26-30 April, Port Canaveral, FL, pp. 64-70, doi: 10.1145/1536513.1536533

[EGG09] ESTEY A., GOOCH A., GOOCH B. Addressing industry issues in a multi-disciplinary course on game design. Proceedings of FDG 2009 - Fourth International Conference on the Foundations of Digital Games, 26-30 April, Port Canaveral, FL, pp. 71-78, doi: 10.1145/1536513.1536534

[Hic10] HICKS A. Towards social gaming methods for improving game-based computer science education. Proceedings of FDG 2010 - Fifth International Conference on the Foundations of Digital Games, 19 - 21 June, Pacific Grove, CA, pp. 259-261, doi: 10.1145/1822348.1822386

[Mar10] MARINELLI D. How to brings artists and technologists together to produce installations that entertain, inform and inspire. Keynote talk, SBGames 2010 - Brazilian Symposium on Computer Games and Digital Entertainment, 8 November, Florianópolis, Brazil

[MKT11] The Games Project website. Delft University of Technology, 2011. http://graphics.tudelft.nl/~mkt4

[NB10] NICKEL A., BARNES T. Games for CS education: computer-supported collaborative learning and multiplayer games. Proceedings of FDG 2010 - Fifth International Conference on the Foundations of Digital Games, 19 - 21 June, Pacific Grove, CA, pp. 274-276, doi: 10.1145/1822348.1822391

[Ogr10] OGRE website. http://www.ogre3d.org/

[PM07] PAUSCH R., MARINELLI D. Carnegie Mellon's Entertainment Technology Center: combining the left and right brain. Communications of the ACM, 50 (7): 50–57

[RW07] REID K.L., WILSON G.V. DrProject: a software project management portal to meet educational needs. Proceedings of SIGCSE'07, 7–10 March, Covington, Kentucky, ACM Press, NY

[Sci11] Science Centre website. Delft University of Technology, 2011. http://sciencecentre.tudelft. nl/

[STG08] SUMNER R.W., THUEREY N., GROSS M. The ETH game programming laboratory: a capstone for computer science and visual computing. Proceedings of GDSCE 2008 - Third International Conference on Game Development in Computer Science Education, February 28-March 3, pp. 46-50, doi: 10.1145/1463673.1463683

[SW04] SCHAEFER S., WARREN J. Teaching computer game design and construction. *Computer-Aided Design* **36** (2004): 1501–1510

[SWM10] SCHILD J, WALTER R, MASUCH M. ABC-Sprints: Adapting Scrum to Academic Game Development Courses. Proceedings of FDG 2010 - Fifth International Conference on the Foundations of Digital Games, 19 - 21 June, Pacific Grove, CA, pp. 187-194, doi: 10.1145/1822348.1822373

[XNA10] Microsoft Corporation. XNA Website. http://msdn.microsoft.com/xna/