Full paper in Proceedings of the 16th Annual NACCQ, Palmerston North, New Zealand, July, 2003 (eds) Mann, S. and Williamson, A. www.naccq.ac.nz

Teaching Technology to the Playstation Generation

Paul Kearney
Stephen Skelton
School of Computing and Information Technology, UNITEC
Auckland, NEW ZEALAND.
pkearney2@unitec.ac.nz

ABSTRACT

Today's computing students arrive in our classroom familiar with a wide range of technology. They are used to rapid change and fast paced, interactive environments that this brings. This is the Playstation generation and engaging them in the classroom requires us to be innovative and creative with our learning strategies. Integrating new technologies such as the latest console and computer games is not a problem when our students have grown up with them in their living rooms. Indeed, the teaching of games development becomes a logical addition to any computing curriculum when you consider that the games industry has outstripped Hollywood and is now worth US$25 billion dollars a year.

In this paper we will discuss an approach to teaching games development that aims to inspire a sense of creativity and a desire for innovation in our students. However, this enthusiasm and creativity must be balanced with the need to acquire often complex mathematical and programming concepts. Games development can be fun but it is never an easy option. We will explore an approach that uses media that our students are familiar with, so that their ideas can be expressed with tools that are truly engaging, interactive and exciting and discuss the pedagogical issues that underpin this.

1. INTRODUCTION

The battle for consumer's hearts and minds in the games console market is escalating as Microsoft takes on Sony for domination in this field. With the gaming industry worth over US$25 billion per year according to the NPD Market research (cited at the Electronic Entertainment Expo, 2003) , the system manufacturers are relying on a constant flow of new and increasingly sophisticated software releases to drive business. This in turn sees educational institutions targeted as a source of new recruits to the world of games development, resulting in a proliferation of new courses in this field. The challenge for those teaching in this is area is the marriage of academic quality and sound pedagogical underpinnings with the right stimuli needed for the creative gamer.

You enter the lab. It is dark and warm with heads focused on monitors, emotion alive in the faces of all around. This is the reality of online games testing rooms in some of the major universities around the world. These are the teaching institutions that have realised the potential of the games industry not only as a serious business and source of future employment for their students but in awakening the minds and imaginations of the Playstation generation.

This paper discusses the creation of a games development course within the Bachelor of Computing Systems at UNITEC. The course has itself been developed as a role-play in which students adopt the persona of a games developer. Some students might be temped to select such a course expecting anything

with “games” in the title to be an easy option. As this paper will show, it is not and games development uses a variety of software development techniques and skills drawn from a range of methodological approaches (including Macromedia and Microsoft) and it requires student’s to grasp the complex nature of game structures, interactive digital media, applied physics and mathematics. In other words, games development is a challenging paper that pushes the boundaries of what we thought our students were academically capable of as well as unblocking their true innovative and creative powers.

2. PLAYING THE GAME

Many of our students come to class with an extensive knowledge base that has been developed through gaming. If our goal as educators is to excite and engage our students then what better way to do this than by applying the same innovations that are seen in the games themselves?

With this in mind, we have developed a model for the games development course that involves a process of:

1. Setting up a “Game Development business” – this gives the students/groups an identity.

2. Evolving a team, roles and a methodology – this involves a commercial group production process.

3. Creating a schedule for envisioning, storyboarding, designing, testing and releasing a product.

4. Building an educational or commercial game in 3D Studio Max, Maya, Director or Flash for a possible commercial release.

5. Presenting the “finished product” in a “celebration”.

3. THE GAMEPLAY

In teaching games development to this audience, we considered that there were some basic principles that must ourselves follow in order to achieve a creative learning environment:

a) Develop games for a broad audience – your games and teaching must embrace both a male and females.

b) Innovation is the key – look at cutting edge applications and help your students to create these – Myst was a winner in its time with beautiful graphics, the Quake engine is regarded by many as the staple of physics games engines, Age of Empires was one of the first with a historical theme and multiple victory conditions.

c) Design by playing – demonstrate examples and critique commercial games.

d) Include critical decision making as part of your gameplay and classes.

e) Start with introducing prototypes and testing early.

f) Make the first minutes of gameplay and classes very exciting.

g) The player must be the hero or heroine (not you as the designer or programmer). Engage their minds…after some sweat they must reach victory.

h) Get the player and student to invest in your game and classes – help you to build the game as a group and build assets e.g. use Half Life in architecture design or gmax to add assets to Dungeon Siege.

i) Create a great story and excellent storyline – it’s at the basis of every culture.

j) Create a series – help your students to think about characters that can develop (like Lara and Zelda) and games that create franchises (like the Star Wars series). Similarly find the links to other courses and encourage practical end of degree thesis or project work to encompass the same professionalism that you introduce.

k) Pay attention to detail e.g. graphics, sound.

l) Create an intuitive interface that’s easy to use.

m) Provide a multiple of different experiences, from single to multiplayer (individual to group assessment), co-operative play and so on.

n) Games are an art and so, like life, are unpredictable – you have to balance what you can do practically (your budget and schedule) with what you dream/envision (the quality).

4. THE DEVELOPMENT CYCLE

The final point above (n) is the most difficult. The current tertiary climate means that it is difficult to optimise educational settings to develop computer games. The cost of games development can be high and with budgetary restrictions this can cause challenges. This fact is often not helped by the stigma associated with playing computer games and the banning of such on campuses (inherent in this assumption is the failure to recognise that the development of a game is a very different matter to the

playing of it). Today's state-of-the-art games use cutting edge, and very expensive technology, and the systems to develop such programmes are even more costly. The cost of developer kits for the next generation of game consoles is upwards of US$50,000, but with the industry predicted to gross $US100 billion this decade, the initial investment appears worthwhile.

The schedule is the next stumbling block. Games development covers everything from texture artists to lighting techniques, from particle physics to spatial mathematics, and of course it includes some very sophisticated programming. Early versions of the Quake 3D engine totalled well over 100,000 lines of code and took many years to write. Most games companies today either build on what they have written during years of expensive development time (reusable code) or they licence the technology from another company and concentrate on creating the storyline and characters. When gamers are playing popular titles such as Soldier of Fortune, Half-Life, Counter-Strike, Sin, Star Trek:Elite Force, James Bond 007, they are really playing Quake with different characters, different graphics, and a new storyline. All of these games use a licensed version of the Quake 3D engine.

5. DEVELOPING A GAMES ENGINE

To develop a commercial games engine today would take an immense amount of skill, talent, and funding. Areas that would need to be learnt, and therefore taught, are as follows.

5.1 Low level programming languages

Many of the commercial engines are written in Assembler language. Although very fast and very powerful, machine-level programming is difficult and time consuming to write. Assembler programmers are very rare today, and this is a skill that is being lost as it is not currently taught in mainstream learning institutions. The C language and more recently C++, are also used, again because of the speed and ability for the programmer to have more control at the hardware level. High level languages such as Delphi and Visual Basic have been designed for rapid application development and tend to insulate the programmer from the hardware. This is undesirable and makes them a poor choice for coding a games engine. Java is rapidly gaining popularity with its cross platform abilities. Coding in Java will make it easier to port the product to other platforms at a later date, however most game consoles such as Playstation and Nintendo, use a combination of Assembler and C.

5.2 Graphics and 3D programming

Most computer games use one of two graphics methodologies - DirectX or OpenGL. Both of these options provide the programmer with an Application Program Interface (API) that allows the hardware manufacturer to build to a defined set of features for use. OpenGL is the more popular as it covers multiple platforms, DirectX was developed by Microsoft. To program either of these technologies, one needs to understand the many facets of 3D programming. These include 3D vector tables and their manipulation using trigonometry, vector calculus, and linear algebra, and the complex art of lighting and texturing techniques, which borders on the role of a graphic artist (Monet spent many hours painting London's House of Parliament just to understand the interplay of light and shadow). Modern 3D video boards are also capable of real time rendering for 3D objects. Because rendering is performed on the video board the load on the processor is reduced, however, the complexities for the programmer are increased.

Many of the expensive Game engines that are available for licence include all the features needed. However they still require a lot of skill and knowledge to configure the code following the purchase. The following areas may or may not be inclusive of the engine, but still fall in the realms of the game developer.

5.3 Game Physics

As games strive for reality, the simulation of real life becomes increasingly important. If a golf ball in Microsoft Golf does not behave correctly when sliced (magnus effect), or a rally car in WRC did not drift through the corners correctly (angular velocity), the game would not sell. The latest game engine developed for Unreal Tournament by Epic Games, uses 'ragdoll' physics. The program computes the point of impact when a character is shot and the character falls accordingly. Particle physics, collision detection, collision response, rigid body kinematics, Runge-Katta integration and even cloth simulation, all need to be modelled correctly if virtual reality is to be achieved. Physics algorithms need to be understood and then transposed into programming code.

5.4 Artificial Intelligence

Terms such as machine learning, fuzzy logic and neural networks are well known, but will a

computer program ever think like a human? Probably not for a long time, but what we are attempting in computer game code is the emulation of intelligence. In most games this is done by creating algorithms such as emergent behaviour and group dynamics, such as flocking. The basic technique is to emulate human behaviour without appearing to have the game "cheat". This skill is more art than science.

5.6 Animation

Animation is pivotal to any modern computer game. However, the techniques used by game developers are many and varied. Some games use actual film clips for the cut scenes between levels. There are games that use stop-frame animation and even some even use claymation for the entire game (such as The Neverhood from Dreamworks Interactive). There are many tools available today that enable even the novice to create a computer animation, however, the real skill lies in choosing the correct technique for the subject and knowing how to create the desired effect for maximum impact.

5.7 Networks and the Internet

One of the main attractions of many computer games today is the interaction with other people. A large portion of the game industry is devoted to multiplayer games, both on a small local network and on a larger, even massive scale, over the Internet. Massive Multiplayer Online (MMO) games have created virtual worlds where upwards on 100,000 people play simultaneously. One attraction for the player is the built in chat screen of the game. An entire new language, similar to texting on a cellphone, has evolved to enable players to play and converse at the same time. The MMO is popular with developers because it offers the opportunity to charge a monthly subscription fee and therefore offers a more stable business model and cash flow than for one-off games sales. Skills needed by the developer include networking and security issues. Synchronising the game with a large player base requires some real in-depth hardcore programming.

6. DEVELOPING THE STORY

Once the games engine is in place, a storyline needs to be added complete with characters, sound, game levels, and a well thought out user interface.

6.1 Characters and Plot

The game must have a goal (an ending) that players feel compelled to achieve. A good storyline and great gameplay often compensates for low-budget graphics and visual effects (just look at Tetris!). The gameplay is such that the player becomes addicted and must complete that next level. First person "shooter" games such as Quake and Half-life have that addictive quality. Role playing and "God" mode games such as Diablo and Warcraft entice the player to achieve that next level by progressively becoming more challenging but allowing the player the satisfaction of winning early in the game. Good character design often creates a cult figure, such as Lara Croft of Tomb Raider. This alone can ensure future markets for the sequel. Developing a storyline for a game is similar to writing a script for a play or a movie and many of the same skills and creativity are required.

6.2 Audio and Music

If you have ever played or watched a game with the speakers turned off, you can see how much added effect sound and music has in creating the game players environment. Take the sound away and often the game becomes just a cartoon on the screen. Add realistic sound effects and appropriate background music and the player becomes immersed in another time and place, a realm of fantasy. But the art of audio involves talent and skill. Admittedly not every game developer is an accomplished musician, but adding the music and timing the sound effects requires a knowledge of Microsoft's Direct Sound API, and the skill of being able to code background sound effects. A sound is just a data stream and has to be read and processed by the game program. The player does not want the game to pause while "BANG, CRASH, BOOM" is being played.

6.3 Level design

A good game designer often overlooks the fact that the player will require levels of difficulty within the game. The player will want to pause at varies stages throughout the game after achieving a certain level of success, rather like the chapters in a book or the episodes in a television program. However this is not as easy as it appears. The developer does not wish to create new game code for each level, nor does the player wish to just have the blocks fall down at a faster rate. For a platform game or puzzle, the level of difficulty needs to increase but at the same time be achievable. A Role Playing Game (RPG) game follows a storyline and should build on this progressively. The skills of the programmer will

include Object Oriented Programming (OOP) and modular design with reusable code. Rather like the next step up from a game engine, the game itself needs to encompass the ability to change levels without re-coding the game.

6.4 GUI programming and User Interface

Many games live and die on their user interface. Does the game use the mouse and/or the keyboard? Does it have a chat window? Are there shortcuts? Can the player configure the controls or customise the graphics? All of these questions become a critical part of what makes a game successful in the market or what makes it "bomb". Good gameplay can overcome a bad interface but players are becoming more and more demanding and in many cases expect to be able to completely change the look and feel of the game through the set-up option.

7. ONCE THE GAME IS WRITTEN

Quality assurance in the form of game testing is critical to ensure that the product is reliable and does what is expected. Developers must also consider issues such as packaging, marketing and publishing. And of course, after-sales support is paramount in today's environment.

7.1 Game testing

Many people enter the Game Development Industry as a tester, moving up through the ranks. To write good games you must play games. Just as an accomplished book author spends time reading books before developing their own technique and style. Team work and communication skills are essential skills here as well since you must be part of the development team, feeding back constructive advice and comments to the developers. Imagine that tact and diplomacy required to tell a hardcore developer that their code doesn't work – especially when he or she has been coding for 24 hours straight on 15 cups of double shot espresso!

7.2 Packaging and presentation

Game design does not stop at the actual product. The packaging, presentation and marketing of the game is an essential factor in whether customers will purchase it, even in whether the developer gets a publishing contract. Many software products are now distributed over the Internet. However, this is often impractical for games due to the size of the files involved. Downloadable demonstrations and trial versions are often developed as a 'teaser' for the market, but this must be thought out carefully as many players exhaust their curiosity playing the demo. Novel and creative packaging also sells product, but there are now industry standard guidelines on how software is packaged for distribution. The trick is how to be creative within these guidelines.

7.3 Game Server Management

Increasingly games now offer Internet content in some or all of the playing of the game. Large farms of Internet servers are installed to cope with the massive load incurred through online gamers. Managing these systems is a specialised task. However, the developer must be aware of what is involved to be able to create an online game that is manageable and will perform adequately on today's hardware.

Conveying an appreciation of all of this in the classroom is not a simple task. Many students would willingly sign-up for three to four years tuition solely on game development (La Trobe University in Melbourne has recently announced a new four year programme, majoring in Games Technology and has been inundated with applicants). For us, we are offering an introduction to the topic in a single semester course.

8. COURSE DEVELOPMENT

UNITEC's introductory course on game development is a limited "first step" in the field. We have started our journey by developing an introductory course with a focus on the programming prerequisites that briefly covers the topics described above. These are covered one per week, over a single semester. As acceptance for game development increases, this course can be extended to develop an introductory course with no prior experience and culminating with an advanced course on game development. The long term objective would be to run numerous different classes, each covering a separate areas of this highly complex yet exciting subject, even culminating in a Degree of Computer Game Development!

Initially we have used Macromedia's Shockwave as a game engine. This decision is based on enabling the student to produce games, from simple through to complex multiplayer, without having to be concerned with the specialised and highly complex 3D game engines (there is simply not enough time in a single

semester course to introduce these products). To further reduce development time and to create a realistic environment, students work in groups, each taking a part of the production cycle. They are asked to create a company name and logo and include this in the credit screen of their game to provide a feeling of ownership and pride.

For more advanced students, Shockwave supports a plug-in physics engine (by Havok), and 3D rendering through 3D Studio Max. Students do not have to code the concept of gravity, friction, or elasticity, as they can apply these properties to an object and add textures and lighting as required.

By using Macromedia's Director, sounds, music, and animation can all be coded with relative ease. The game is then produced either using a full screen projector or as an object within a web page.

The final objective is for the group to create a game that demonstrates all of the elements that are taught. From the initial concept and planning, through to creating a storyboard and finally compiling and testing the game. Presentations and critiques are done by the whole class and marks are attributed for original and unique ideas in gaming.

9. THE END GAME

Many students attend to programming class disillusioned by all that coding, but when they apply it to games and can see the result of their work in a short space of time in 2D or 3D spaces, the light of awareness and accomplishment dawns. They get excited and are filled with energy. Suddenly it matters, it has an affect on "mass" in the real world, it does something they can relate to and it launches them at high speed into the very lucrative area of games technology. Engaging students in exciting and energising tasks is crucial to deep learning. These console and computer game fans have been profoundly influenced by this sort of media, so why not indulge them by immersing them in it to create breakthrough creativity. This may be exactly the type of breakthrough media that computer science students want the opportunity to produce.

10. THE GAME OBJECTIVE

It is our hope that students are able to at least conceptualise how they could create a computer application or game, which meshes together converging technology and transforms life, potentially a game so incredible that it causes millions to rush out and buy it. We also want students to gain an appreciation that game development is serious business and that this is not a "back room" industry. The original computer game was created back in 1961 on a mainframe computer at the Massachusetts Institute of Technology (MIT). It simulated an intergalactic battle and was tilted Spacewar. This was the game that started the whole industry and it was developed by a student (DeMaria, R. and Wilson, J. 2002). Gaming today is a serious business; Myst generated US$320 million for the publishers, the developers receiving 10% of this. The Electronics Arts Game "The Sims" has recently overtaken Myst in terms of revenue and Nintendo's character "Super Mario" has reportedly generated more than US$500 million since conception.

11. CONCLUSION

The universe is itself a game of massive forces, which over time has seen sorcerers, shamans and prophets seek to understand it through a combination of astrology or mercurial alchemy. Our society has spawned cultures based on these patterns and tapestries and bred geniuses who come from time to time to tell us what it all means. The electronic universe is little different: The current breed of virtual worlds have given rise to Zelda (legendary creator Shigeru Miyamoto), Lara Croft (Toby Gard), Age of Empires (co-creator Brian Sullivan) and Starcraft (Blizzard Entertainment). Behind every gaming hero is a talented and innovative game designer.

Gaming is about finding a way to tap into our dreams so that we are able push the boundaries of possibility, engaging in new, exciting and energising ways. Although today we might see computer games as a new and developing industry, it is one that lives on the outer edge of our computing knowledge and which is already big business. It is our hope that in this paper we have been able to give you an insight into the complexities of computer games development and demonstrate how this can be brought into the classroom in an engaging, challenging and pedagogically valid way.

REFERENCES

DeMaria, R. and Wilson, J. (2002) .*High Score! The illustrated history of electronic games.* McGraw Hill/Osborne, California.

Electronic Entertainment Expo (2003). Retrieved on 28 May 2003, http://www.e3expo.com/media_center /articles/051203_game Makers.shtml