MATHEMATICS FOR COMPUTER GAMES TECHNOLOGY

Kevin WILKINS.
School of Information Technology
Charles Sturt University.
Panorama Ave. Bathurst, 2795 Australia
kwilkins@csu.edu.au

ABSTRACT

Despite their entertainment focus, Computer Games are at the forefront of computer science. These days the successful computer games (programs) make strong calls on a number of other disciplines: psychology, mathematics, statistics and physics to name but a few.

Enrolment numbers in mathematics courses across Australian Universities are in decline, however we have found that students who have enrolled in our new degree, Bachelor of Computer Science (Games Technology), are not just tolerant, but are enthusiastic about the mathematics component of the degree. In this paper we describe the mathematical demands of games technology along with the development of the mathematical sub-component of the degree.

Our program, being a full strength computer science degree demands some standard mathematical components: discrete mathematics, linear algebra, numerical analysis, statistics and ordinary differential equations.

The strong emphasis on computer graphics programming needs a firm foundation in aspects of linear algebra; the virtual world of the game scene development is underpinned by the physics of movement. Acceleration, cornering, collisions, explosions and disintegration all require ODEs.

A subtle shift in the shape of a probability distribution can help to maintain game balance and player interest by giving the underdog an unseen helping hand. Logic, computational complexity and numerical analysis speak for themselves.

Mathematics educators have always known that relevance is a strong motivator of mathematics.
This has again been ably demonstrated by our first cohort of games technology students who want more mathematics.

Key Words: Computer Games; Computer Graphics; Mathematics Education.

1. Introduction

In 2001 the School of Information Technology at Charles Sturt University in Bathurst, Australia admitted the first cohort of students into its new degree, Bachelor of Computer Science (Games Technology). For the previous two years the School had been developing curricula for this course. The degree was to span four years, and was designed for an unusually bright cohort. We were aiming the curricula for students from the top 10% of their age cohort with proven mathematics and computing skills from their secondary schooling.

The Australian Computer Games industry was an enthusiastic supporter of this degree. Their help in the development of this program should be acknowledged. It would be the only degree level training available in Australia for their workforce. The degree program has a considerable component of work experience and agreements are in place with the industry to place the final year students within appropriate projects.

On a worldwide basis the Computer Games industry is larger than the Movie industry. It employs more people. This is not so surprising when you consider its scope. It encompasses hand held devices (Game Boy), the games consoles (Play Station, Sega, Microsoft's X-box), PC games, Internet multiplayer games, arcade games, gambling casino games (poker machines) and Internet gambling. The games themselves range from sophisticatedly intelligent strategy games, such as chess, to first person shooter games, to games of pure chance (gambling).

The divide between the Movie industry and the Computer Games industry is blurring at a rapid rate. Movie special effects with computer generated backgrounds or worlds are becoming common. Animated characters are no longer just the province of cartoons. Filmed video sequences played out with live actors are now being spliced into some computer games as scene setters.

Computer graphics also have serious uses. Medical imagery and other simulation programs are examples. Computer Games are at the leading edge of real-time computer graphics technology, but much of the mathematics associated with the generation of these images is fundamental. It is to be found within the mathematics courses that we already teach.

The 'Games Technology Development Committee' allocated only four subjects to mathematics, however some other subjects, such as computer graphics, do contain other elements of mathematics (A subject is about 48hrs of instruction over one ½year session.) Efficiency within the university dictated that we did not have a free hand to develop completely new subjects. There were some suitable mathematics subjects already in existence. Three existing subjects were chosen to fill the allocated space and one new subject had to be developed. They are:

- Discrete Mathematics
- Dynamics (New)
- Ordinary Differential Equations
- Life, Chaos and Virtual Worlds.

There is always some freedom to tweak the existing subjects to fit a new cohort of students. Some examples from computer games programming were added and small changes to the list of topics to be covered were made. You will also note that the new subject is also quite traditional.

Our campus does not have a physics department so we did not have a Dynamics subject on offer. However it will be developed with Computer Games focus.

Our teaching experience at this stage is limited to the Discrete Mathematics subject. The subject was taught to a class of some 74 students, of which 28 were the Games Technology students.

2. Discrete Mathematics

Discrete Mathematics covers a standard set of topics. These are:

- Logic, Sets and proof.
- Numbers, Combinatorics, Complex numbers.
- Probability.
- Vectors, Matrices and Solutions of Systems of Equations.
- Graphs.
- Recurrence and complexity.
- Boolean Algebra and Logic Circuits.

The subject was taught as a mathematics subject, with the usual emphasis on understanding. The development of the theory was not compromised. To cater for the new cohort of students, Probability was a new topic introduced to an already full curriculum, and examples drawn from Games programming were used to enhance the relevance of the study to these students.

The majority of the group, not just the Games students were motivated by these examples. These examples are from the students own field of interest and experience, real (perhaps I should say virtual) world examples of mathematics in use.

We shall have a look in a little detail at some of the ideas used.

Probability. Lecky-Thompson [2001] discusses two quick and dirty pseudo random number generators (my description). Both are probably not good enough for Poker Machines, and both will quickly generate a sequence of numbers, random enough for most game applications. Both sequences are repeatable given the same seed value. These random numbers are sampled from the uniform distribution.

How do we deal out a set of shuffled cards at the start of 'Free-cell' or 'Pairs'?

How do we generate random samples from other distributions?

How do we make a monster so that its behaviour is a little unpredictable and becomes a challenge to shoot?

Two-person games quickly loose their appeal if one player consistently outperforms the other. This is known as Game Balance. How can we change a probability distribution to bring the game back into balance and keep both competitors enthusiastic? A shift to the underlying probability distributions can also be used to increase the level of difficulty of a game as we enter new phase or level.

Matrices. (3D Graphics). The action of the First Person Shooter or the Car Racing Simulation takes place in a 3D virtual world. Objects and characters are built from a set of points in 3D. These points are stored in a data structure as columns of coordinates relative to a convenient local coordinate system. The objects are manipulated, (moved, rotated, scaled) to their desired shape

and orientation then positioned in the world by a ‘change of coordinates’ to the world coordinate system.

Interestingly, a 4D homogeneous coordinate system is used. This has the distinct advantage over 3D of allowing translations to be achieved by a matrix multiplication. Associativity of matrix multiplication allows this entire placement procedure to be accomplished in one pass of matrix multiplication across the data structure.

Example. Homogeneous coordinates.

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} \Rightarrow \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \qquad \begin{bmatrix} w_x \\ w_y \\ w_z \\ w \end{bmatrix} \Rightarrow \begin{bmatrix} \frac{w_x}{w} \\ \frac{w_y}{w} \\ \frac{w_z}{w} \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \Rightarrow \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

3D Þ 4D **4D Þ 3D**

Transformation matrices.

Rotation about x-axis.

$$R = \begin{bmatrix} Cosq & -Sinq & 0 & 0 \\ Sinq & Cosq & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Scaling to desired shape and size.

$$S = \begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Translation.

$$T = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

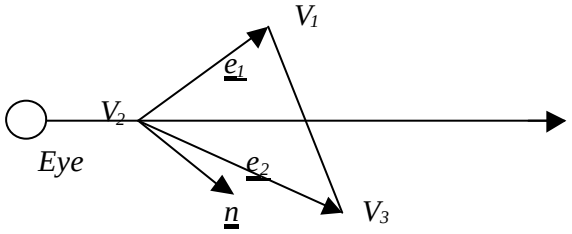
Note:

$$\begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{bmatrix} x & + & t_x \\ y & + & t_y \\ z & + & t_z \\ & 1 & \end{bmatrix}$$

Transformation Matrices for rotations about the Y and Z axes are similarly built.

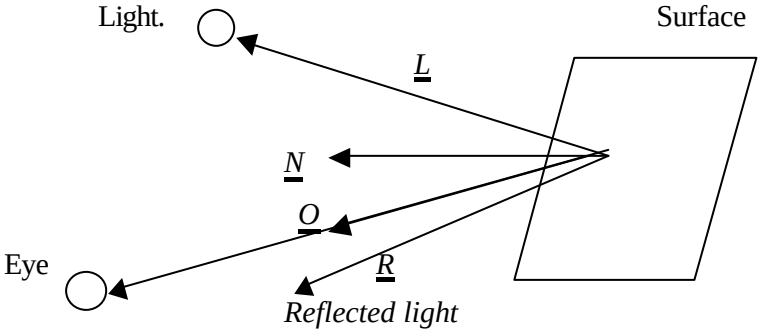
The usual composition of transformations rules apply, non-commutivity can be demonstrated easily and matrix inverses are needed to undo transformations and for change of coordinate systems. Put the movement in a loop and you have a stealth bomber flying in doing a barrel roll.

Vector products. (Hidden Surfaces). Once an object is positioned in the game space, we need to render its seen surface. Deciding which part of an object is seen, and which part is hidden, is achieved by a process that uses both vector products. We cover the set of points that define the object with a convex polygon mesh. Within this data structure the polygon vertices are always ordered in a clockwise pattern when viewed from the outside of the object. By taking cross products of two suitably defined edge vectors we generate a normal vector to the plane of the polygon. This normal will point out from the object's surface. This allows us to quite simply determine if this section of the surface is visible or hidden from view. If the dot product of the vector from the observer to $\boldsymbol{V_2}$ with the normal a $\underline{\boldsymbol{n}}$ is positive then this is a back surface and hidden from the observer. Hidden sections of the surface are deleted from the data set.



$$\underline{n} = \underline{e_1} \times \underline{e_2}$$
$$= (V_1 - V_2) \times (V_3 - V_2)$$

More vectors. (Light). Various lighting scenarios, ambient light, parallel light and point source light are used to give the Game scene a touch of realism. If either of the last two scenarios is used then the brightness of that part of an object is dependent on the component of reflected light reaching the observer.



If $\underline{\boldsymbol{N}}$ is a unit normal then we can show that $\underline{\boldsymbol{R}} = 2(\underline{\boldsymbol{N}} \bullet \underline{\boldsymbol{L}})\underline{\boldsymbol{N}} - \underline{\boldsymbol{L}}$. The component of reflected light is $(\underline{\boldsymbol{O}}.\underline{\boldsymbol{R}})\underline{\boldsymbol{O}}$ if $\underline{\boldsymbol{O}}$ is a unit vector in the direction of the observer. Scaling factors on the length of this vector can then be used to generate the desired effect. Spot lights from the observer that can be turned on and off (a torch) are used to good effect in some game play.

Graphs. (Path Finding). Perez and Royer [2000] make the observation that finding the shortest path between two nodes on a graph is quite often needed when programming. It can be used for planning airline trips to generating the 'door to door' directions using GIS software in modern cars. In the game environment often the 'badies' are hunting you. Dijkstra's algorithm finds the shortest path from any source node to all connected nodes in a directed graph with positive weights. Getting the ghosts to attack PacMan or the 'beast' to attack in Doom seems a little more exciting than the well trodden travelling salesman.

3. Dynamics

Some of the earlier computer games captured motion by holding pre-scripted animation sequences in storage. These sequences were fairly easy to detect as they all started and finished with the same neutral position. In the older fight games this position was the boxing stance from which the player had a choice to have his character punch, kick or jump. Storage requirements were later reduced by holding only the significant positions of the motion sequence and achieving the animation by dynamically interpolating between these positions. This is still quite limiting as only a small number of prerecorded options are available to the player.

Dynamic simulation (or physics based animation) can be used effectively to script the motion of objects. These can be used to great effect in Simulation Games like F1 Racer and Flight Sim.

For the sake of realism some modern car racing simulation games have gone as far as modelling weight changes on sprung suspensions components in cornering. [See Brian Beckman's articles on *The Physics of Racing*, Tutorials section of the Gamedev.net site.]

The Dynamics subject covers topics such as:

- Newton's Laws of motion.
- 1D particle dynamics within various types of force fields.
- 1D oscillating systems.
- Motion in 2D and 3D.
- Systems of particles, conservation laws and collisions.
- Rigid Body motion, Moment of Inertia and rotational motion.
- The Lagrangian and Hamiltonian formulations of dynamics.
- Rigid body motion, inertia tensor and Eurler's equations of motion.

There is still "work in progress" on this subject for now.

4. ODEs

In any simulation, the equations for the physical system must be set up in advance of any play. If the ODEs have an analytical solution then these are the stored equations and the motion can be

calculated accurately and quickly. However if the ODEs do not lead to such a solution then they will need to be solved in real time by a numerical integration technique. The speed of solution, along with the level of desired accuracy will determine the type of numerical procedure employed.

The ODE subject currently covers:

- First and second order linear ODEs.
- Power series and Laplace transforms
- Special functions, Gamma, Bessel and Legendre polynomials.

It is the writer's view that the special functions section of the course should be replaced with a section that covers numerical integration.

5. Life, Chaos and Virtual Worlds

This subject was developed initially as an elective study for interested students (mainly Information Technology students). It was thought to be important enough in ideas and concepts that it covered to become a core subject within the Games degree.

Fractals are now a mainstream topic in computer graphics. They are used in creating exciting visual patterns of plants, mountains and surface textures. Students will visit Cantor sets, measurement, mapping, iterated function systems and a set of tools for describing chaotic attractors. It is shown that Chaos can arise from very simple differential equations of just three variables in a dscrete system. Students will learn how to cast an equation into a form, which is amenable for numeric solution. They will develop an appreciation of the sensitivity of such systems to initial conditions. In artificial life we get to a more difficult and less understood topic. Evolution and fitness landscapes are useful mathematical techniques. Emergent behaviour is a powerful idea but it still lacks a rigorous formulation. Finally, aside from the beautiful images that we can create with fractals, chaos and emergent behaviour, students will gain an appreciation of the unifying nature of mathematics and its powerful inner beauty.

$$\text{chaos} \Rightarrow \text{fractals} \Rightarrow \text{emergence} \Rightarrow \text{chaos}.$$

6. Conclusion

For a person of my age a certain level of proficiency with a pool queue is said to be a sure sign of a misspent youth. These days a misspent youth would be signalled by a similar level of proficiency with a computer game pad. We can take advantage of the student's intense interest in computer games because of the wealth of mathematics that is used in the development of these games. The development of the mathematics curricula for the Bachelor of Computer Science (Games Technology) degree shows that although much of the needed mathematics is quite fundamental in nature.

There are of course many areas of intersection between Computer Games and Mathematics that I have not touched upon, artificial intelligence paradigms such as neural nets and genetic algorithms are to be found in Game intelligence. There are also a number of mathematical skills that are developed in the secondary schools that are found in Games. Intersections of lines and

planes and spheres are used in collision detection techniques. The breadth of mathematics used is in this area is immense.

It was not only the Games Technology students who were motivated by these examples. It was apparent that all students were interested by the examples. Vectors, Matrices, probability and graphs were seen to be useful mathematical constructs by the students.

Mathematics educators have always known that relevance is a strong motivator of mathematics, but relevance has to be **'- to the student'**. Games development was found to be an ideal vechile for the motivation of much of the mathematics that we teach.

Acknowledgment: I would like to express my thanks to Professor Terry Bossomaier for his encouragement and help in writing this paper.

REFERENCES.

-Lecky-Thompson, G.W., 2001, *Infinite Game Universe: Mathematical Techniques,* Massachusetts: Charles River Media.

-Perez, A., Royer, D., 2000, *Advanced 3-D Game Programming using DirectX 7.0,* Texas: Wordware.

-Watt, A., Policarpo, F., 2001, *3D Games: Real-time Rendering and Software Technology,* New York: ACM Press.

ON LINE REFERENCES.

-Goodman, D., 2000, *The use of Mathematics in Computer Games*, Mathematics Enrichment http://nrich.maths.org/mathsf/journalf/may00/art3/index.html [accessed: 9/1/02]

-Unknown, *Computer Graphics Topics,* Georgia Tech College of Computing. http://www.cc.gatech.edu/gvu/multimedia/nsfmmedia/cware/graphics/notes/vue3d/3dmap/3dmap00.html [accessed: 23/1/02]

Game Developers Net, *Math and Physics,* A list of resources, books and articles: http://www.gamedev.net/reference/list.asp?categoryid=28 [accessed 23/1/02]

Gamasutra, a large site of information on games and programming http://www.gamasutra.com/ [accessed: 23/1/02]