Gaming Technologies for Anatomy Education

Jonathan Mackenzie, Gavin Baily, Michael Nitsche & Jem Rashbass

Abstract

We report on a project which is exploring how the technology and language of computer games can be used to create novel resources for anatomy education.

We present our experiences of developing a simple anatomy game, with an emphasis on software development issues, and offer this as a case study for those wishing to create 3D educational content within an academic context.

Keywords

Computer games, 3D visualisation, Anatomy education, Content development

Introduction

There are a number of software packages using 3D graphics which are dedicated to anatomy education, Voxel-Man and Primal Pictures being among the best (Voxel-Man, 2000; Primal Pictures, 2000). In these and many others, the anatomy is presented in the form of an atlas, where huge amounts of information are presented for open-ended navigation, with no guidance or means to weight its relative importance.

The mode of interactive 3D graphics used by these examples, and in fact by most medical visualisation software, is limited to rotation around, and zooming into, objects, as if they are at arms length, and the addition or subtraction of layers.

Computer games, however, present a very different use of the medium. A classic example is Tomb Raider: an adventure game where you are immersed in a virtual environment that you navigate via the heroine Lara Croft whom you control and direct in an unfolding adventure narrative [tom]. The game-play forces you to engage very differently with the three-dimensional information through your negotiation of spaces and artefacts, through manipulation of camera point-of-view, and through engagement with the narrative structure of the game.

We are interested in the way in which this new language of computer games might be applied to educational resources, especially those which, like anatomy, are inherently three-dimensional subjects.

We wish to explore and test a number of ideas, including:

Can a games-like approach improve student engagement with a difficult subject?

Can activities and narrative structure enhance learning and retention of knowledge?

Can a mental, or 'mind's-eye' model of three-dimensional structure be learnt incidentally to the activity and narrative of the game? We are thinking of the way in which we unconsciously learn architectural information and form mental maps while navigating everyday spaces.

Can useful three-dimensional information be inferred from this mental model? In the case of anatomy, in clinical and surgical settings for example.

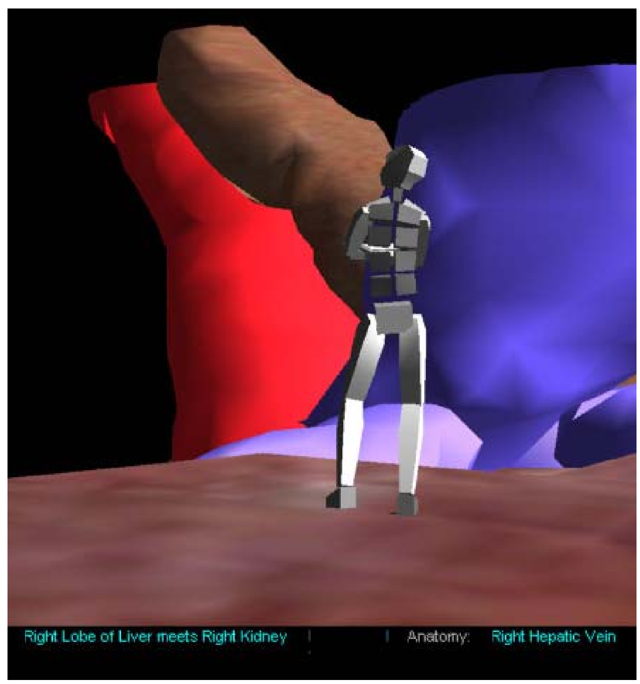


Fig. 1. Screen Capture from the Virtools authored Anatomy Game

This work is part of a wider project at the Centre for Applied Research in Educational Technologies (CARET) at the University of Cambridge, to explore the use of games technology as an educational medium. Anatomy has formed our first focus, but we are developing a generic approach which can easily be applied to other strongly three-dimensional subjects, such as physics and geography.

There are other groups looking at developing educational software using games technology. The Games-to-Teach project in the Department of Comparative Media Studies at MIT have developed a number of conceptual frameworks for educational games that explore cultural and pedagogic issues surrounding game play (Games-To-Teach, 2002). NESTA Futurelab aim to pursue projects in this area, but with content aimed more at school children (NESTA Futurelab, 2002).

We have found, however, that one of the main issues in developing experimental content is the inaccessibility of games technology to educators in academia. Commercial games development is based on technically sophisticated programming expertise, large budgets and production teams. The disparity between the interests and expertise of game developers and educators has been identified as a major issue to be addressed if quality content is to be produced (Games in Education, 2002).

Our approach to this issue has been to develop the means to create experimental content within a University setting, and to investigate both the development tools and working practices required.

In this paper we present our findings in the form of a review of the current state of games development technology, and our experiences of developing experimental content with our chosen solutions.

Anatomy education

Anatomy is a notoriously difficult subject to learn, both because of the sheer quantity of information and, some would argue, because it is dull when taught as an isolated subject. It is often difficult to engage and motivate students to digest the vast numbers of structures and names and, once this is achieved, the crammed knowledge is often forgotten or difficult to apply in clinical practice.

From early on in this project we have developed a dialogue with the staff and students of the University's Anatomy Department, through use of our Learning Laboratory and seminars. The specific needs of the Cambridge syllabus have therefore shaped our approach. This syllabus increasingly emphasises clinical relevance of anatomy, as well as traditional relational and functional aspects.

Games production

Computer gaming is now an industry which is beginning to rival that of film, both in cultural and financial terms. Large teams of programmers and artists with multi-million pound budgets produce highly polished virtual environments experienced by a large proportion of the population. But because of this economic model of production and the technical sophistication of realising a real-time virtual world, this is a medium that is fairly inaccessible to people outside the industry.

It is, however, a powerful and compelling medium, open to a far wider and richer range of uses than are chosen by the games industry which typically caters for adolescent male interests. One of the research aims of the team at CARET is to look at how this medium can be made more accessible to the academic community, and to explore how it can be used as an educational medium within realistic limits of budget and expertise.

To this end we have investigated a wide range of development solutions and put in place a suite of software and hardware tools, with the accompanying expertise, which enables rapid prototyping of virtual environments and games-like interaction so that we can freely experiment with the medium and evaluate our efforts. While we are not intending to match the production values of commercial games, and in fact worry about the expectations they raise in users of our software, we can begin to creatively explore the potential of games-like software and examine the huge range of issues surrounding its use in an academic context.

Software Options

The production of computer games revolves around two main areas: programming the games engine - the software which generates the real-time 3D graphics on screen and processes all the user interaction - and designing and creating assets, such as 3D models, the texture maps which colour them, animations and characters.

Assets are typically created using one of several industry standard 3D modelling packages, such as 3DS Max, or Maya. Each of these are large products containing a wide range of tools for making, editing, animating and rendering virtual versions of anything from characters to landscapes. There is little difference between the basic functionality of these packages, and the choice is between price, features, and user interface design. There is much more variation, however, in the implementation of the games engine.

The games engine is typically programmed using the C++ programming language and the OpenGL or Microsoft DirectX 3D graphics libraries (openGL, 2002; DirectX, 2002). This gives most flexibility and control over image quality, speed, and the creation of novel eye-catching techniques and effects, but requires many man-years of specialised expertise, and is now only really done by the big production houses which lead the industry.

There is a distinction between a 3D games engine, and 3D scientific modelling and rendering software, such as the Visualization Toolkit or Open Inventor (VTK, 2002; Open

Inventor, 2002). The latter does not include the functionality for easily implementing certain important games elements, such as character animations, or object collision detection.

Middleware, which are games engines and additional components being sold and supported by specialist companies, is increasingly being used by smaller development companies, but still requires considerable programming expertise and time to implement. Widely used examples of middleware include Renderware and Intrinsic Alchemy (Renderware, 2002; Intrinsic Alchemy, 2002). There are also a large number of open source games engines, either written by enthusiasts or derived from older games no longer for sale, but these tend to be either of poorer quality than middleware, or not well documented or supported. Examples of these include Crystal Space, Doom and Unreal (Crystal Space, 2002; Doom 2002; Unreal, 2002).

At the other end of the spectrum, are games editors - software which is released with a particular game that allows the player to modify the games or to create their own levels, characters, and graphics. These are fairly easy to use, but do not allow for adaptation of the fundamental modes of interaction and game play.

Somewhere in the middle of the spectrum are a small number of dedicated games authoring packages, being part middleware, part games editors. These include Blender, WildTangent and Virtools (Blender, 2002; Wild Tangenet, 2002; Virtools, 2002).

We spent some time fully investigating these options, both from a technical point of view, and considering issues of academic working practices. Part of our research aim is to build a community of experimenters and researchers across a wide range of academic subjects who are able to share and communicate using a common software language.

Questions considered include:

What are the learning curves for using these tools, both for an experienced programmer, and for a non-programmer?

Which would most easily facilitate a community of users so they could share work and expertise?

Our conclusion was that Virtools by far offered the most attractive solution for our needs. Virtools stands out from other software in its class in the way it supports several modes of use. It can be used as a graphical development environment where assets can be imported and assembled into scenes. Assets then have 'behaviours' attached to them, chosen from a large library, and the relationships between assets and behaviours are graphically manipulated with a block-diagram and wiring editor. This effectively presents a very well conceived interface onto the process of constructing the programmatic elements of a 3D game, without the need to write any program code. The downside of this approach is the paradox that although complex elements such as character animation and simulated physical behaviour are very simple to implement, sometimes simple tasks become very involved.

All of the functionality available in the development environment, which includes 3D transformations, camera and lighting control, animation, characters, input devices, rendering effects, and data management, is also available as a well structured C++ class library. Any additional functionality required can be added through user-written behaviours which use and extend this code base.

Resulting projects can be played using a free downloadable browser plug-in, which makes for easy web deployment of content, or written as standalone executables for Windows.

Experiments

Zanzarah

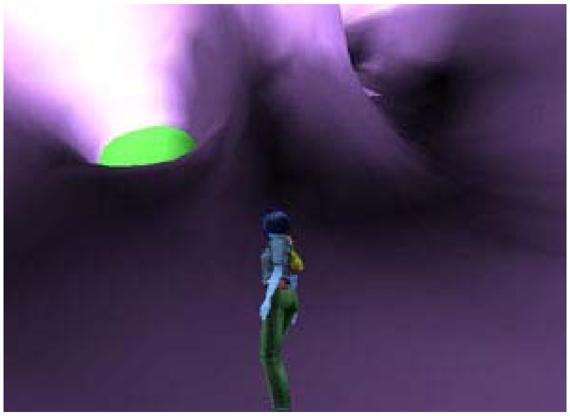
The aim of our first experiment was to get an interactive 3D environment working as quickly as possible so that we would have something to show that would seed discussion. Working with our colleagues at the Cambridge University Moving Image Studio (CUMIS) we used the engine and editor from Zanzarah, a game developed by German developers Funatics. (Zanzarah, 2002). Zanzarah is a fantasy adventure game in which, as in Tomb Raider, you play the part of a character who negotiates the virtual environments in the course of the narrative.

Into this engine we imported a simple 3D surface model of a digestive system, adapted from a model taken from an online 3D modelling resource (3D Cafe, 2002). The user can follow the route of the digestive system and has control over both the movement of the character and the camera point-of view.

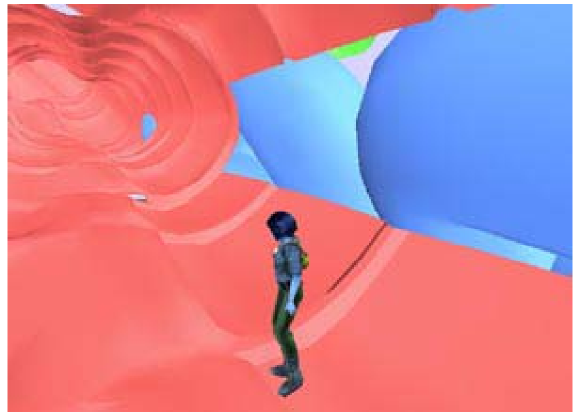
You begin by being dropped down the oesophagus (shown in blue) into the stomach (purple) where you may run towards and then leap into the duodenum (green). In another version the model lies flat, instead of upright, and you run along the length of the sigmoid and descending colon (red), where a window has been cut allowing you to view the colon's relationship to the small intestine. You can then jump thorough this window into the abdominal cavity.

This simple experiment succeeded in its goal of capturing interest and sparking discussion with the lecturers, medics, colleagues and students we have shown it to. The main conclusions we drew from these discussions were that the following would be necessary:

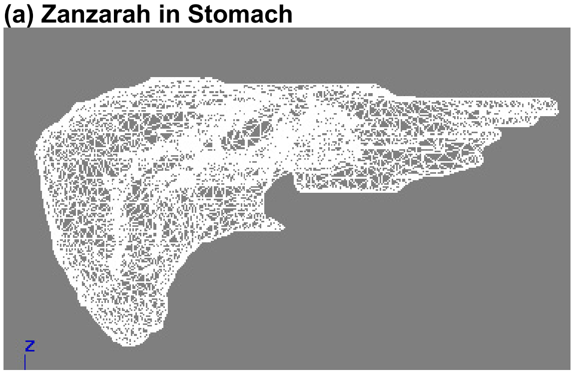
To create proper anatomical models by using digital data sets, such as the Visible Human (Visible Human, 2002), CT or MRI.



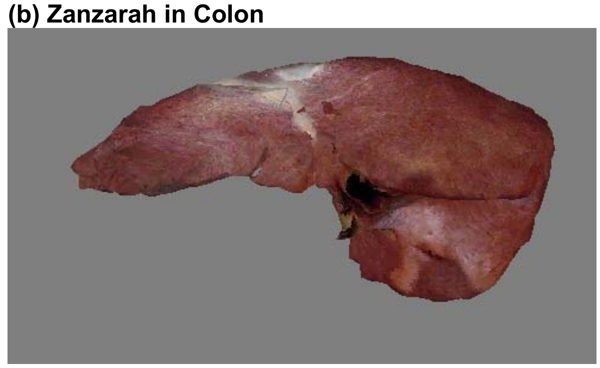
(a) Zanzarah in Stomach



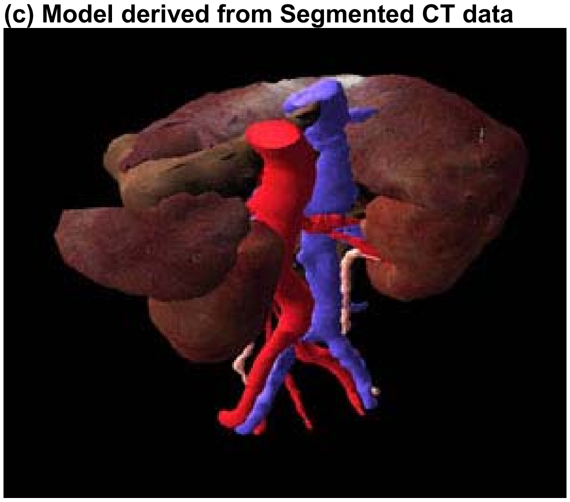
(b) Zanzarah in Colon



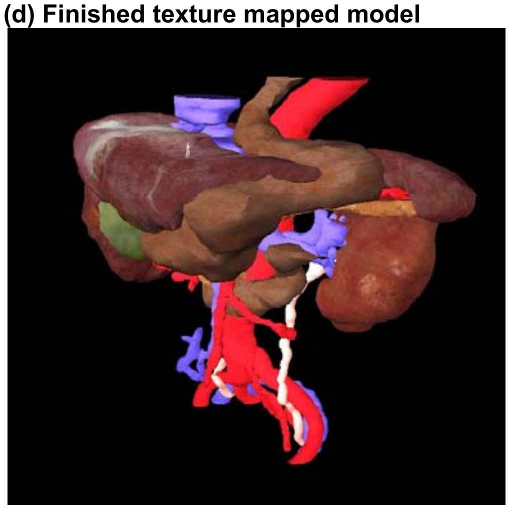
(c) Model derived from Segmented CT data



(d) Finished texture mapped model

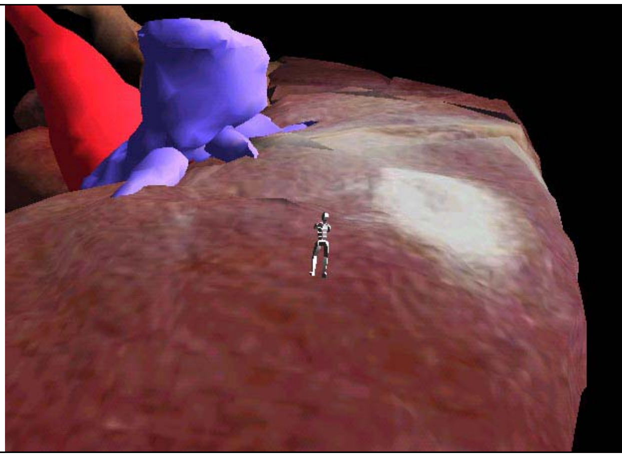


(e) Virtools: Abdomen 1



(f) Virtools: Abdomen 2

Fig. 2



(a) Virtools: Character walks across liver

(b) Virtools: Character navigates on fixed path through stomach.

Fig. 3

To explore different ways of representing solid anatomical objects using 3D graphics. We are limited by the processing requirements of the real-time 3D games engine to depicting surfaces, but these can be rendered in a number of styles, for example transparent, or photo-realistically textured.

To experiment with forms of navigation. Is a character needed? What role does scale play? What alternative control could the user be given over the space? For example, is normal gravity a restriction, or could the model be turned so that you can walk around objects without falling off? How does the first-person perspective affect understanding of the virtual space and its applicability to real-world situations?

The games editor approach was too limited to allow us sufficient control to investigate these issues. After much investigation as outlined in 2.3, we chose to use Virtools as the means to create the second prototype.

Virtools

Our second prototype developed into being a simple activity set in the abdomen. Again, the user navigates a character, but this time on the outside surface of several abdominal organs: liver, stomach, spleen, gallbladder, kidney, and some major arteries. The game is aimed at teaching first year undergraduates a number of clinically relevant abdominal features.

The user has to find a sequence of markers positioned around the abdomen which draw attention to the location and relative arrangement of these features. The location of the markers are described in text on the screen (bottom left), and the user can interrogate objects in the scene by pointing the mouse cursor at them causing a text description to appear bottom right.

The evolution of the game involved research in the following main areas:

generating informative 3D models from anatomical data sets;
devising usable and comprehendible navigation;
creating pedagogically sound game play.

Anatomical Models

To create the anatomical models, we processed a segmented CT data set created by the Brigham and Women's Hospital Radiology Department.. This consists of a volume of CT data values, arranged in slices, each accompanied by another slice containing segmentation indices for each data value. This allows each anatomically distinct object to be separated out from the volume of CT data.

Using the Visualisation Toolkit (VTK) and its marching cubes algorithm, polygonal surface models are generated for each segmented object. Various stages of smoothing and decimation are also involved to get well formed models of a desired resolution. For our use, object sizes of between 1,000 and 10,000 faces were required to run in a real-time rendering engine

After generating the polygonal models, these are then texture-mapped to give surface colour and detail. While the polygonal models capture the gross anatomical forms, the surfaces are used more as a means to differentiate between objects, and to provide an aesthetic to the virtual space. Texture mapping was accomplished using photographs taken from an anatomy atlas (Yokochi et al 1989) which were then matched to the surface models using 3ds Max and Deep UV texturing tools (3ds Max, 2002; Deep UV, 2002).

Having created a set of anatomical models that would work in the Virtools real-time rendering engine, we experimented with a number of visualisation, navigation and interaction options. Fig 3. shows two prototypes demonstrating some of these, including rendering of solid and transparent surfaces, navigation on inside and outside of surfaces, and movement along fixed paths or unrestricted. One of the great benefits of Virtools is that altering and testing these different options can be done without complex programming.

Evaluation

The Anatomy Game prototype is being evaluated using the Learning Laboratory facility at CARET. So far this has involved groups of first year undergraduate students playing the game, followed by an interview asking about usability, opinions and anatomy. This process is beginning to answer the questions posed in the introduction, and given valuable feedback to prototype development. Our initial findings are that students do form mental models through playing the game which they are then able to interrogate to give accurate anatomical knowledge. We also found that students enjoyed using the game and that the game play encouraged enthusiastic group working. More detailed and rigorous results of this and further evaluation will be presented elsewhere.

Conclusions & Further Work

The technical sophistication of creating 3D game-like content, and the huge budgets available to commercial games developers are major obstacles to the use of this medium in academic contexts. This, however, is starting to change and, as we have shown here, recent advances in authoring tools and middleware enable small research projects to explore the creation and evaluation of educational content.

For this process to develop effectively there needs to be an increase in the overlap of knowledge between those able to create and program games and those with ideas for using this medium. We are finding that tools such as Virtools allow researchers from different backgrounds to communicate using a common software language and readily share their work.

References

Games-To-Teach (2002). http://cms.mit.edu/games/education
NESTA Futurelab (2002). http://www.nestafuturelab.org
Voxel-Man (2002), developed by Institute of Mathematics and Computer Science in Medicine, University Hospital Eppendorf, Hamburg, published by Springer Electronic Media, http://www.uke.uni-hamburg.de/institute/imdm/idv/forschung/vm/index.en.html
Primal Pictures (2002). http://www.primalpictures.com
Tomb Raider (2002), developed by Core Design, Published by Eidos, see www.tombraider.com
Games in Education (2002). British Educational Communications and Technology Agency (BECTA) Online Conference. http://forum.ngfl.gov.uk/WebX?14@@.ee738de
OpenGL (2002). http:// www.opengl.org
DirectX (2002). http://msdn.microsoft.com/library/default.asp?url=/nhp/Default.asp?contentid=28000410
Visualization Toolkit (2002). http://public.kitware.com/VTK
Open Inventor (2002). http://oss.sgi.com/projects/inventor
Renderware (2002). http://www.renderware.com
Intrinsic Alchemy (2002). http://www.intrinsic.com
Crystal Space (2002). http://crystal.sourceforge.net
Doom source code and editors (2002). http://www.idsoftware.com/business/home/techdownloads
Unreal Tournament editor (2002). http://www.unrealtournament.com/editing
Blender (2002). http://www.blender3d.org
Wild Tangent (2002). http://www.wildtangent.com
Virtools (2002). http://www.virtools.com
Zanazarah (2002). Developed by Funatics, http://www.zanzarah.com
3D Cafe (2002). http://www.3dcafe.com
Visible Human project (2002). http://www.nlm.nih.gov/research/visible/visible_human.html
Yokochi, C., Rohen, JW. & Weinreb, EL. (1989). *Photographic Anatomy of the Human Body*, Igaku-Shoin, Tokyo.
3ds Max (2002). http://www.discreet.com/products/3dsmax
Deep UV (2002). http://www.righthemisphere.com/products/duv

Acknowledgements

We are grateful to the following for their generous help and support: Funatics Development for allowing us to use their Zanzarah game engine and editing tools. Criterion Software for the Renderware middleware used in the Zanzarah engine. Brigham and Women's Hospital, Harvard, for CT data sets.Ian Parkin and his students from the University's Anatomy Department