Virtual Bronchoscopy: Using Game Design Techniques and Technology to Create an Interactive 3D Teaching Tool

Leslie Predy and Marc Dryer

This paper addresses a newly developing area in biomedical communications: 3D interactive, didactic game design. Using the creation of a virtual bronchoscopy simulation tool as an example, we discuss the appropriateness of certain gaming concepts in this context, the fundamentals of game design software, the applicability of the preset software functionality, and the potential for future development in this field.

Overview

Bronchoscopy is a procedure in which a flexible endoscope is inserted through a patient's nose or mouth, passed through the trachea, and into the respiratory tree. The feed from a fiber-optic camera mounted at the end of the endoscope is displayed on a television screen, and the physician uses this to guide the scope. Bronchoscopic examination is used to inspect the respiratory system for pathological abnormalities.

One of the main challenges facing residents learning to perform bronchoscopy is the three-dimensional (3D) spatial understanding required to maneuver the endoscope (Crawford and Colt 2004). The scope can be moved forward and backward, rotated around its viewing axis, and the end can be flexed in one axis. Residents must orient the scope properly and correlate the inner bronchial anatomy being viewed on screen with the overall position in the lung to identify anatomical landmarks to successfully conduct an examination (Colt *et al.* 2001). Our goal was to create a teaching tool that would address these needs, yet would be widely available as a web-based application, rather than being restricted to a surgical skills lab. We used a first -person perspective (FPP; also referred to an FPS, or *first person shooter* in a more limited application) game-based approach to create an interactive, 3D virtual bronchoscopy teaching tool. Although this tool has a very specific application, it serves as an example to illustrate many of the general considerations that must be made when using gaming technology.

The first person perspective (FPP) video game is one that renders the game world from the perspective of the player character. Typically, the player must navigate through a complex 3D environment and achieve predefined goals by moving to, and interacting with, set objects. The first person view in these games is analogous to the view of the endoscope navigating through the bronchial tree, making the constructs of interaction and spatial awareness in an FPP relatable to those needed in the process of bronchoscopy; this similarity of interaction style provided the impetus for using a game design-based approach for this teaching tool. In both instances, the user must control their movement through an environment using fine motor skills, while concentrating on the visuals presented on a screen. Research has shown that game play, and video game play in particular, can mediate self-regulated learning by intrinsically motivating the learner (Reiber 1996). In order to be profitable, a video game must be fairly easy to learn, challenging, and entertaining. The user must want to continue playing the game to its conclusion. The tenets of game play, such as layered complexity in challenges, problem solving, and understanding the rules that govern the environment, engage people in a profoundly immersive way, enticing them to continue with the gaming experience (Gee 2004). By applying these same principles to a didactic tool, a stimulating, interactive learning environment can be created.

The bronchoscopy simulation was designed with these features of immersive gaming in mind. Different modes of play or "levels" were created. The first was a beginner or exploratory mode, where the user can navigate the scope through the bronchial tree, and is prompted with text describing various anatomical landmarks. The second level presents the user with a destination (e.g. the lower left bronchus), that they must find without the aid of descriptive hints. In the future, it is hoped that we will be able to model various lung pathologies and create a simulation where the user must locate them, building diagnostic skills.

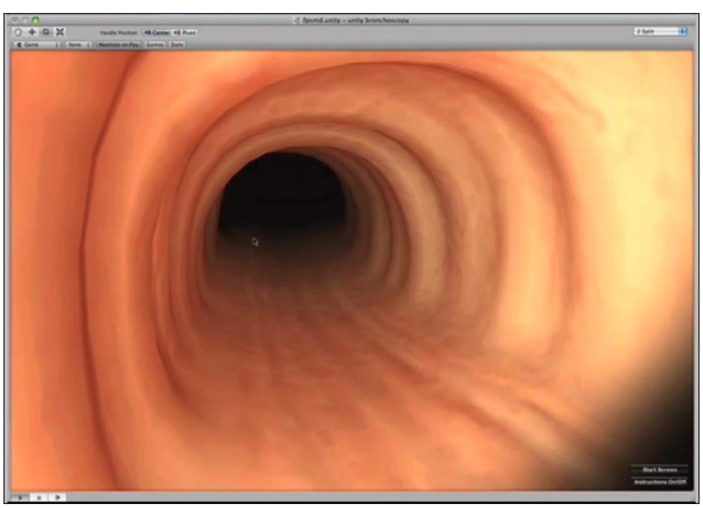
Game Engine Technology

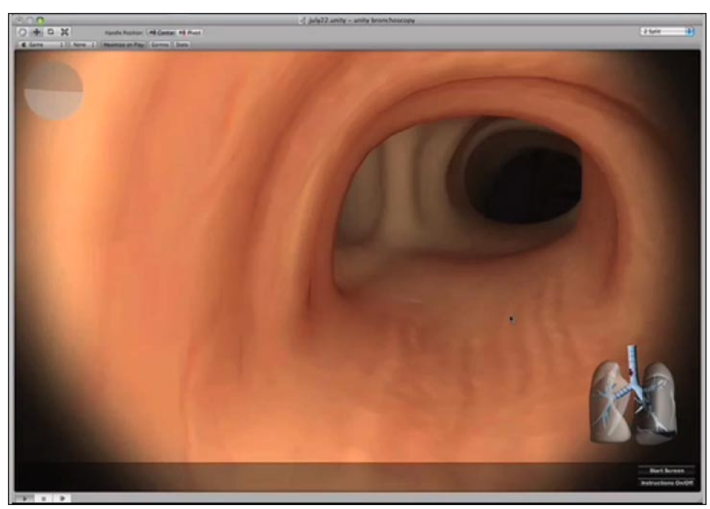
As the simulation was based on video gaming principles, we decided to use a video game design application, or "game engine" for its development. A game engine is a software program used to design interactive 3D environments that are rendered in real-time. Game engines provide visual development tools (visual representations of programmatic code) in addition to reusable software, scripting and other components (Eberly 2001). 3D character and environment models, textures and animations are not created in the game engine software; rather a 3D application (in this case, Autodesk's Maya 8.5) is used to develop these core assets. The game engine functions to programmatically define interactions between models, perform physics and collision detection, create the game GUI (graphical user interface), and render the product. One of the major benefits of game engines is that they provide a suite of built-in and reusable components, allowing for quick application development. For instance, the developer can design their own models and rules for game-play, while making use of software presets like collision detection. Although manufactured with a game design workflow in mind, game engines are often used for other kinds of interactive applications with real-time graphical requirements such as interactive product demonstrations, architectural visualizations, and training simulations. Game engines also have a tremendous, yet untapped potential to be a powerful biomedical visualization tool. Blood flow simulations, interactive three-dimensional anatomical teaching modules, and educational games are but a few examples of applications that could be easily developed with this software.

The Use of Game Engine Preset Functionality and its Contextual Limitations

As the FPP is such a popular video game genre, most game engines come with a built-in character controller, which can be used to create the first person perspective camera through which the user interacts with and views the virtual environment. However, the FPP preset is typically designed to emulate a character walking on terrain, and as such, causes the character to be subject to dynamic gravity simulation, and to bob up and down, emulating the gait of a walking individual (see: http://brodel.med.utoronto.ca/~leslie/FPSpreset_jbc.mov). While these presets realistically simulate an exploratory game situation, they were not applicable to bronchoscopy. In our application, the FPP is that of the scope's view, and unlike the preset character view the scope needs to enter branches of the bronchial tree both above and below, and should not be affected by gravity. The forward movement of the scope must also take into account the friction along the walls and the momentum of scope movement, and the fact that the FPP camera in this case is tethered to the long scope behind it, further limiting movement options. Simulating the bronchoscope view and control, while sharing the basic qualities of preset FPP, required very different physical properties from a character walking across terrain. These properties are key to creating the realistic feel of a simulation. Consequently, the presets that came with the game engine were not suitable for a bronchoscopy simulation. The challenges faced by biomedical communicators developing these types of tools are often quite unique and software presets do not take such situations into account. In order to overcome these challenges, we simplified the basic scripting that controlled movement; removing the walking/jumping functionality and the variables controlling momentum/friction/gravity were adjusted to more readily emulate scope movement, (see: http://brodel.med.utoronto.ca/~leslie/Scripted_jbc.mov).

Certain aspects of the game engine's preset functionality were useful, however, such as the abilities of the built-in physics engine. The game engine calculates all movement and collisions using a physics engine, and the game designer can programmatically adjust aspects of this movement with ease. A physics engine is a computer program that simulates Newtonian physical laws, and can dynamically emulate different real world conditions and apply them to the scene. Collision detection, friction, and bounce can all be easily modified without delving into in-depth physics equations. Creating elements like streaming cells for blood flow, interlocking or interacting receptors and much more can be done simply and easily: a great asset for the biomedical communicator as little programming is required.





Two screen grabs from bronchoscopy simulations.

Technological Limitations and End-User Considerations

Although using a game engine provides one with the advantage of easily creating interactivity, there are some end-user standardization costs. To visualize an interactive virtual environment, real-time rendering must be used. The scene is rendered using the computer's graphics processing unit (GPU), and therefore, the visual quality of the product and its responsiveness to user input will vary depending on the graphics card the end-user has. In contrast, linear, non-interactive animation is characterized by pre-rendered image sequences where the entire duration of the scene or film is distributed as a movie to viewers. This allows for the high-quality, life-like renderings seen in 3D biomedical animations or the computer graphics in popular media, and allows for a consistent product, but not for interactivity. The essence of the video game experience is the multiplicity of choices afforded to the user by the ability to render in real time. Many avid video game players have specialized machines with high-end graphics cards to ensure an optimal game-playing experience. However, when designing for a medically-oriented audience, the biomedical communicator must take into consideration that many medical professionals have computers geared toward office use and are not optimal for real-time rendering. One must make certain compromises in quality to ensure that the end application can run on the majority of machines. Most of these compromises involve aesthetics, as the rendering quality differs with the graphics card.

The process of how the computer renders the 3D scene is complex and requires explanation that is beyond the scope of this paper. However, one pertinent aspect of game design that must be addressed is the role of shaders on render quality. In the field of computer graphics, a shader is a set of software instructions used by the graphics card primarily to perform rendering effects, responsible for calculating the color and lighting of an object. Graphics software libraries such as OpenGL and DirectX program the graphics card by defining special shading functions (Eberly 2001).

In a game design application, a shader describes the quality of a model's surface using painted or procedural textures and colors, and its appearance under varying lighting conditions. Generally speaking, there are two types of shaders that calculate lighting information differently. Vertex shaders calculate lighting at vertices only, so interpolated "overall lighting" is drawn over the actual pixels of the object's geometry. These shaders are simpler than the alternative, pixel shaders, which calculate the color and lighting of individual pixels. Vertex shaders do not display any pixel-based rendering effects, such as specular highlights, bump mapping, or shadows. Creating the look of a corrugated, wet, flesh-like surface such as the trachea depends greatly on creating realistic specular falloff and bump maps and seemingly necessitates the use of pixel shaders. However, using pixel shaders has a performance cost, and graphics cards have varying levels of support for pixel shaders. So one must compromise and attempt to find alternatives for those running machines less capable of pixel shader support. In creating a virtual bronchoscopy simulation, the longitudinal muscular bands and cartilaginous rings lining the trachea were actually modeled into the polygon geometry in Maya, as opposed to being created with a bump map. A bump map containing fine details was created, but if the user runs the application on a machine incapable of rendering the bump map, the important anatomical landmarks will still be rudimentarily visible. The model was constructed under the supervision of, and approved by a staff scientist and pulmonologist at Toronto General Hospital, to ensure the anatomy would remain accurate when the application was used on different machines.

Another consideration when using game engine technology is the limit imposed on the polygon count of models. Game engines must consider each model in the scene, including those parts not visible to the camera. Because of this, game engines institute a polygon limit to improve performance. Modeling more intricate geometry into the bronchial model required a greater level of subdivision, increasing the polygon count. At a higher level of subdivision, the model exceeded the polygon limit and couldn't be imported. This was easily overcome by breaking up the model into three parts, but this problem illustrates another consideration for biomedical communicators who frequently require large, detailed models. The creation of a realistically modeled bronchial tree presents its own unique requirements, but the limitations of shaders and model polygon count can be widely generalized to many game design applications.

Distribution Considerations

Game engines have a robust ability to publish for a wide variety of platforms. In addition to making standalone applications for various computer operating systems, many engines can publish streaming web applications, or be used for mobile device application development. The bronchoscopy teaching tool was published as both a downloadable standalone and streaming web application; web players do require installation of a special plug-in to interpret the unique game engine file format, and because of restrictions in IT policy in many institutions, including hospitals, the use of the web-based application may be hindered. In this case, the standalone application could be an acceptable alternative.

Potential for Future Developments

The highly adaptable technology of the gaming industry provides exciting possibilities for new ways of human/computer interaction. Game engines allow one to readily tailor an application to work with many controller devices. In addition to the traditional mouse, keyboard and joystick inputs, there is the possibility for creating applications using touch screens, haptic (pressure sensitive) devices, or wireless controllers. Trends in the gaming industry are moving toward more creative and tailored controllers, such as the Nintendo Wii remote™, Novint Falcon™, or the many specialized control devices of EA Games' *Rock Band*™.

These trends hold great potential for the field of biomedical communication, opening up a new frontier for the design of medical simulation, educational games, and interfaces in general. The falling costs and widespread availability of new interaction technology means these devices are no longer restricted to use in large institutions such as hospitals or other training facilities. Medical and surgical simulations in particular can provide the user with much more realistic methods of navigation and feedback by improving upon the standard mouse/keyboard computer inputs. For example, we created a version of the bronchoscopy simulation that is controlled using the Nintendo Wii remote™, with an infrared sensor connected to the computer via Bluetooth. The user can manipulate the Wii remote™ to control the pitch and yaw movement of the scope head, while the roll controls its rotation. A secondary controller (the Nintendo™ Nun-chuck) is used to advance the scope forward and backward. This allows the user to control the simulation using hand movements similar to those they would use when actually performing the procedure, adding an entirely new layer of realism to the simulation.

Conclusion

Game engine technology is a greatly advantageous development tool: it is very easy to create interactive environments with realistic physical laws, and the possibility for web-based distribution can obviate the need for location-based equipment. However, there are certain considerations that must be made. Some of the preset functionality of game engine applications is inappropriate for biomedical projects. The lack of standard end-user equipment leads to limitations in terms of a consistent rendering aesthetic, and the target audience may be restricted from using the tool due to institutionally based internet security restrictions. Relying on video game technology presents a series of benefits and compromises that need to be addressed for each particular project.

Although there are technological limitations at present, the gaming industry is growing at an incredible pace, both financially and technologically. A market research study done in 2006 predicted game industry revenue to double by 2011 (ABI 2006). Video games developers lead the field in developing new interfaces, technologies and platforms. This dynamic field offers the biomedical communicator exciting opportunities for not only using new technologies, but also for implementing new methods for developing educational tools. Educational researchers are validating the principles of game design as a framework for promoting learning, and these principles can be applied by the biomedical communicator to create educational tools. All of these factors make the emergent field of game design one that is sure to be influential on the field of biomedical communications.

References

ABI Research. 2006. Video Game Hardware, Software and Services: Online and Mobile Gaming Drives Console, Portable and PC Gaming Devices. http://www.abiresearch.com/products/RR/GAME/06/file/944

Colt, H.G., S. W. Crawford and O. Galbraith. 2001. A revolution in procedural training: Virtual reality bronchoscopy simulation. *Chest* 120: 1333-1339.

Crawford, S.W. and H.G. Colt. 2004. Virtual reality and written assessments are of potential value to determine knowledge and skill in flexible bronchoscopy. *Respiration* 71:269-275.

Eberly, D. 2001. 3D *Game Engine Design: A Practical Approach to Real-time Computer Graphics*. 2nd ed. New York: Morgan Kaufmann.

Rieber, L. 1996. Seriously considering play: Designing interactive learning environments based on the blending of microworlds, simulations, and games. *Educational Technology Research & Development* 44(2):43-58.

Gee, J.P. 2004. *What Video Games Have to Teach Us About Learning and Literacy*. New York: Palgrave Macmillan.

Authors

Leslie Predy, BSc, MSc.BMC, graduated from the Biomedical Communications (BMC) Program, University of Toronto, in 2008. She is currently employed at InViVo Communications in Toronto, where she is using interactive media for creating learning tools and applications. This Masters Research Project was supported by the Canadian Institutes of Health Research (CIHR) Health Communications Award and a Vesalius Trust Research Grant. leslie.predy@utoronto.ca

Marc Dryer, Hons BA, MSc, MScBMC, is a lecturer in Biomedical Communications (BMC), Institute of Communication and Culture (ICC), University of Toronto Mississauga (UTM). He is cross appointed to the U of T Knowledge Media Design Institute (KMDI). His academic background, taught curriculum, and present research interests involve the visualization of data in Anthropological sciences, particularly in Archaeology, Paleoanthropology, and Forensics, in addition to subjects in the life sciences. In these areas his focus is on the use of 3D modeling and animation to facilitate the communication of research findings and to aid in scientific discovery itself. marc.dryer@utoronto.ca