Enhancing Game Realism via Particle Models, Character Skeletons, and Contextualized User Interface

Jenna Langer
jenna.langer@csus.edu

John Morales
john.morales2@upr.edu

David Oluwatimi
o.oluwatimi@nsu.edu

ABSTRACT

Meta!Blast is an educational video game about cell biology geared towards high school and college students. The game helps students visualize an accurate representation of a three-dimensional plant cell. Students learn cellular plant biology by exploring a virtual soybean cell and performing biochemical reactions to help keep the cell alive. Today's video games are constantly improving graphics and visual effects. The same is needed for educational games in order to keep the user's attention. The goal of this research is to enhance the user experience by animating characters, adding particle systems to enhance visual realism, and creating a look and feel that matches the artistic approach of the visuals.

1 INTRODUCTION

Meta!Blast is a video game in development that teaches plant biology. The target audience is high school and college students because complex cell biology can be very hard to visualize. Meta!Blast allows students to explore an accurately modeled three-dimensional cell. The purpose of the game is to revive a soybean cell by using knowledge of biology. The game intends to keep students immersed and entertained while teaching them complex biological material. Meta!Blast is a game of "edutainment," meaning it consists of a balance between education and entertainment. The goal of this game is primarily to teach biology, but entertaining students is also an important factor.

Video games are extremely popular among young adults and create "worlds that help us learn by integrating thinking, social interaction, and technology" [14]. But the problem at hand is designing a game for educational learning that is engaging and easy to learn. Current video games that are awarded accolades, prizes, or recognition are often designed solely for entertainment, and games that are unsuccessful in entertaining are typically considered a complete failure. "Too often…fun games are designed and instructional designers come in and suck all the fun out it' in the quest to meet instructional goals. Herein lies the big challenge for designers of next generational learning environments using next generation technologies" [10]. To create an educational game based on the high standards of current popular video games is a hard feat in itself, but not impossible. This paper describes how the addition of particle systems, realistic character animation, and a unique graphical user interface to Meta!Blast will enhance the user's experience of an educational game.

2 BACKGROUND

A 2003 study by Jones found that 65% of college students play video games on a regular basis and 48% of college gamers say they play to keep themselves from spending too much time studying [9]. If Meta!Blast engages and entertains students, they may not feel like they are studying while playing the game. Jones also stated that 23% of student gamers "ranked realistic graphics as the most important feature of a good game" [9]. According to Rieber, students are more likely to play an educational game multiple times if it includes dynamic graphics. The graphics are said to "enhance the motivational appeal of instructional activities" [5]. If the student is motivated, he or she is more likely to continue the activity for longer periods of time.

Entertainment is an "effective way to transmit information,"[4] making the game a fun and effective tool for teaching biology. A plant cell is a very complex concept that is hard to visualize. Kurt Squire suggests that allowing students "to interact with a model of a complex system…places learners in a unique position to understand the system's dynamics" [16]. Using visual realism and contextual aesthetics will keep the students immersed in the environment while interacting and learning from "working simulation of a cell"[4].

3 RESULTS

Several components were added to Meta!Blast to enhance the user's experience and the visual realism. Particle systems were added to simulate natural phenomena resulting in visual enhancements. This will improve Meta!Blast by adding realism and visually stunning graphics. The realistic animation of the characters in Meta!Blast will be improved by applying the techniques of skeletal animation and inverse kinematics. A unique user interface look and feel was designed to graphically blend with the game and the currently designed Heads-Up Display (HUD). It was also designed to be easy to use so users are not distracted from the game.

3.1 PARTICLE SYSTEMS

Particle systems provide an effective method of creating visual effects within an educational game while maintaining the same level of engagement as popular video games. Particle systems are a three-dimensional graphics technique that replicates complex environmental phenomena using individual particles. These systems can simulate physics-based events such as explosions, clouds, or smoke. "The resulting model is able to represent motion, changes of form, and dynamics that are not possible with classical surface-based representations" [14]. Particle systems have been commonly used to create special effects in popular movies such as The Incredible Hulk or Wall-E. For example, in Wall-E, particle systems are used to simulate spray from a fire extinguisher (Figure 1).

Figure 1. The use of particle systems in *Wall-E.*

Particle systems have been developed in Meta!Blast to increase a player's engagement based on the visual effects. Using various effects, to this date, we have created smoke, bubbles, fireworks, gravity, mixed geometry patterns, and explosive behaviors. For example, if a player in Meta!Blast decides to use adenosine triphosphate (ATP), a biological energy source for all living cells from plants to humans, to attack a dangerous virus cell that may harm the plant cells (preventing the player from completing his or her mission), then the particle system will give the player a dazzling simulation of an explosion. Meta!Blast's particle systems can simulate an explosion from the initial burst to the outward force it releases, to the residue the virus will leave from the cell body (This is shown in Figure 2).

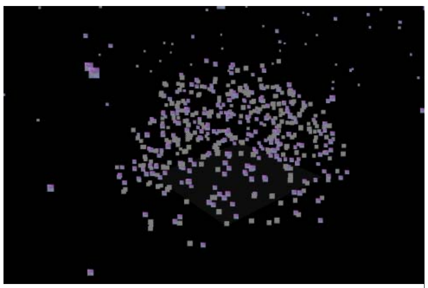

Figure 2. Explosion simulation of outward force.

Particle systems in Meta!Blast, as is typical for other particle systems, entail defining each particle's lifespan, initial velocity, and generation rate; these have all been added. Since particle systems have a range of applications, each of the systems must have a range of behaviors that are associated with it. "Behavior," in the context of particle systems, describes how particles change in various ways using different effects, e.g., the spherical behavior shown in Figure 3. The particle systems in Meta!Blast were modeled in order to exhibit behaviors similar to that of particular natural phenomena. The system in Figure 3 has been calibrated so the particles' velocities are uniformly sampled from a spherical distribution.

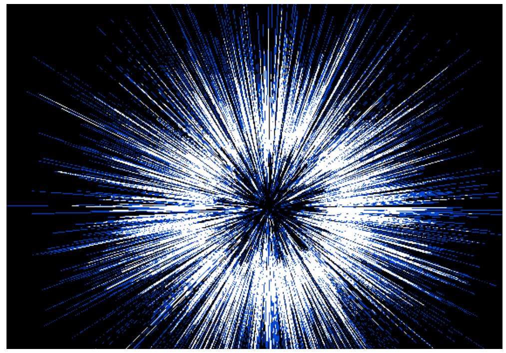

Figure 3. Particles in a spherical pattern.

Particles can be directed by anything within physics model. For example, particle systems can be affected by velocity (the change in direction), acceleration, and gravity. Visually, particles may change properties such as color and opacity based on the model parameters. However, these factors may require the particles to be drawn in specific ways (i.e. polygons, points, lines, or mixed geometries). Each geometry has its pros and cons. If each particle is drawn as points and/or lines, the scene will be rendered very quickly as a result of its simpler geometry. While points and lines are simple, they are limited because they cannot create intricate visuals. Polygons (quads) and other geometries can offer more intricate visuals by applying textures to them, but more complex rendering can reduce the realism of the scene. Because intricate visuals were desirable in Meta!Blast, implementation of polygons particles were used for some effects like bubble generation. In bubble generation, a bubble-like texture is attached to a particle system using a 3-D sphere model created in MAYA, resulting in a realistic scene of an explosion of bubbles (Figure 4). Unfortunately, large numbers of polygonal particles have an adverse effect on system performance. The system began to slow down after rendering 1000 bubbles, and the system significantly lagged after 1500 was rendered. This lag can be reduced by reducing the number of polygons on the sphere, allowing the system to render more bubbles. (Note: CPU: Dual Xeon 3.2GHz, RAM: 2 Gb, OS: WinXP Pro x64, Quadro FX 3450/4000 SDI, Memory: 256 MB)

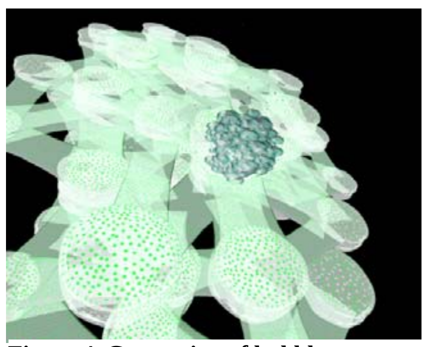

Figure 4. Generation of bubbles.

2.2 CHARACTER ANIMATION

Animation is an important aspect of a realistic video game character because humans can easily distinguish between natural and unnatural movements in an animated object (in this case a video game character). People can get disappointed very easily because of an effect known as the "uncanny valley" [6, 8]. The creation of a hierarchy of bones and limiting movements or joints can make characters motions appear more realistic. Kinesiology studies this motion to acquire data to describe joints movements [11]. Skeletal animation and inverse kinematics (IK) are brought together to create the animation of a character.

3.2.1 Skeletal Animation

Skeletal animation is the application of movement to a series of bones ordered in a hierarchical structure. This hierarchical structure is built from of a root bone or father bone, followed by its set of children bones (Figure 5).

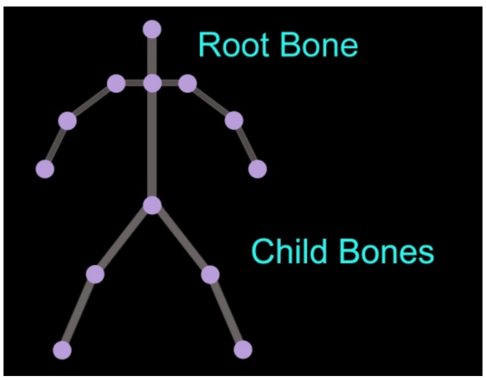

Figure 5. An example of a skeleton hierarchy. The root bone and its children can be seen.

A good example of a bone hierarchy is the human spine. The spine is the root bone, and the shoulders are its children. The shoulder's children are then the humerus bones. This skeletal animation is beneficial because of its similarity to the human skeleton structure. It makes the animation process more identical to human motions. It also requires little memory usage [3,13]. This computer graphic technique also has a 3D surface (some cases can be 2D) called a mesh or skin. The mesh can be represented in different ways. For example, the mesh can be a polygon, or in the case of Meta!Blast, a 3D model of a character. The first step of character creation is to build the skeleton hierarchy. Each bone has been named, in order to simplify the search in getting a bone to be animated. The root bone has been assigned to the head, followed by the neck, the sternum and so on, till forming a structure just like in the figure above. After creating the skeleton structure, the animation is added to each bone by separate. This process makes it easier constrain the joints. Once the skeleton animation is set the mesh is attached to the bones.

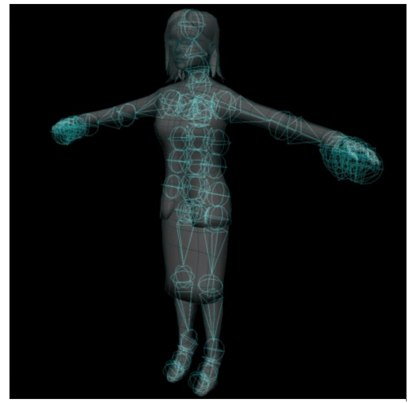

Figure 6. A character's mesh from the game attached to the skeleton.

Figure 6 shows the mesh of Clara Phyllton, one of the characters of Meta!Blast. The mesh has no movement and is formed out of vertices. The vertices are positioned in a 3D space in respect to the bone transformations, making the mesh move at the same time that the bones move, exactly like human anatomy.

The skeleton animation process mentioned above was used for animating the characters in Meta!Blast. The process was first tested with another character. This character (now called Stick Guy, Fig. 7) is also made of a skeleton hierarchy, and has a mesh made out of squares. The animation assigned to the character as can be seen in figure 7, based in a person doing jumping jacks. The process has been use in other video games with success. In an article review, Eike F. Anderson mentions, "character animation in games like Tomb Raider is based in skeletal animation" [3]. The methods used to animate Stick Guy can now be applied to the others characters of Meta!Blast. With this system, a more life-like character animation can be created in less time and with more efficiency.

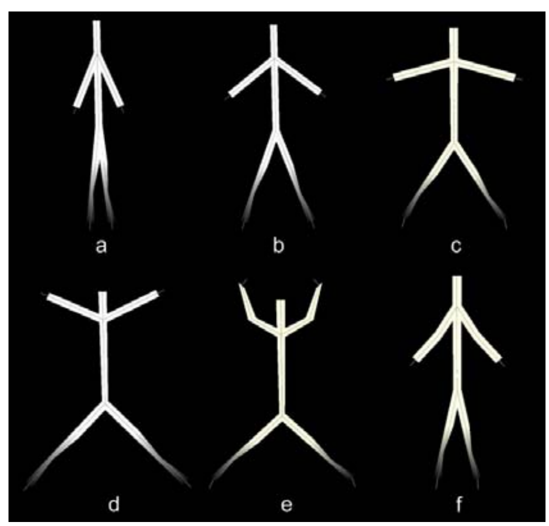

Figure 7. Stages of the animated "Stick Guy" doing a jumping jack.

3.2.2 Inverse Kinematics

IK is applied to the skeleton structure joints in order to eliminate unnatural movements in the bones. IK can be defined as the mathematical problem of calculating the angle of the joints in a hierarchical object based on a desired position for the endpoint. For accurate human-like movements, the addition of IK gives the skeletal structure a smoother and more realistic view and appreciation. The human brain demonstrates its solutions of IK when it calculates appropriate joint positions in the arm for moving one's fingertips to a desired position.

IK is used to animate the joints of a character. The joints of the characters have a certain range of movement. First, the shoulders come up, then the elbow bends towards the head by 120 degrees, and finally the wrist also bends toward the head (Figure 7e). This functionality will be added in the future. Before the IK and the skeleton structure were added to Meta!Blast, the mesh of the characters where static or motionless. The addition of skeletal animation and the future addition of IK will correct the unnatural movements.

3.3 INTERFACE LOOK AND FEEL

The look and feel is determined by the design of the interface components, or widgets, and how they behave. It is used to create a sense of branding and product recognition. A successful widget set, or skin, should be easy to use, easy to learn, attractive, and engaging [12]. Currently, there is a design for the HUD, but it has not been integrated into the game. The user interface look and feel adds widgets created for Meta!Blast. This includes buttons, checkboxes, textboxes, scroll bars, sliders, menus, lists, and progress bars. The skin is designed to look like the game, encompassing organic shapes and colors of plant life, as well as display a 2D GUI in a 3D world. Creating visually pleasing, interactive, and consistent widgets for the game is intended to enhance the user's experience by making it interesting and easy to use and learn.

3.3.1 Unique Design

Just as Windows and Mac use a consistent look and feel unique to their brand, the buttons, text boxes, and other controls of Meta!Blast have been thematically skinned to match the game. The look and feel of the widgets is designed to mirror biology and plant life. Many of the components have rounded corners to give the interface an organic feel. Shades of blue and green are the primary colors used because they are associated with nature. The shapes remain simple as to not take away from the visuals of the game itself. The widgets are designed with the HUD in mind so they will work well together, but they have enough differences to distinguish between the game and the interface. The interface design keeps users engaged with Meta!Blast while they are not performing actions in the game.

The look and feel components will be displayed on 2D panels within the 3D world of the game. While the GUI could have been designed in 3D, it was more effective to use 2D interface that is standard in the computing world [7]. The interface is authored in OpenGL, like the rest of the game, giving it the ability to be attached anywhere in the 3D scene. UI Elements are not restricted to a flat plane (labels will be used directly in the 3D environment), but the majority of the interface components will be displayed in 2D on top of the game. This will help distinguish the GUI from the game and allow for easy interaction because "2D GUI technology is "very mature and well understood" [7]. Although the buttons are not 3D, their design gives the appearance that they can be physically pushed.

3.3.2 Affordances

Effective interfaces allow users to automatically know what they can and cannot interact with. The widgets in the GUI are created to have affordances, meaning the users know how they *should* be able to manipulate the interface components [12]. For example, the buttons in Meta!Blast are shaded so they seem to pop out of the screen, giving the illusion of a physical button that can be pressed. Creating different states for buttons (up, over, down) tells users that the button is dynamic and can be selected. When users click on a button and see the text move simultaneously with a change in color and shade, it appears that the button is being pushed down. Figure 8 shows the different states of a standard button, though the changes are easier to see when users interact with them.

Figure 8. The up, over, and down states of a cancel button.

The users now learn this action and will know how to interact with buttons in the future. Assuming that users have some computer experience, they will not think when they perform the action. Once they perform the action it is stored in their muscle memory [12]. By using rollover states and active states consistently throughout the interface, users can interact with the widgets once and not have to think about them for the rest of the game.

3.3.3 Consistency

Consistency helps users by letting them assume multiple widgets work in similar ways. Users will explore many objects in the game to see what they can and cannot interact with. Looking for game objects is designed to be fun during game-play, but searching through tools and widgets in the interface takes extra time and is counterproductive [12].

The Meta!Blast GUI is designed with consistent shapes, colors, and interaction with many of its components. For example, any widget using a text box has a background with a green pattern. When users are looking for a certain button they will know to skip over anything with the green background because it is not what they are looking for.

Figure 9. Components with textboxes have similar backgrounds.

Another example is the use of the same arrow buttons for spinners and combo boxes. Users will know that the arrow button means that when they press the button the attached textbox should change. The disabled states of the different components were also designed consistently throughout the GUI (see fig. 10). A frequently used technique for cuing users about the disabled status of a widget is reducing the contrast of disabled objects because less attention is drawn to it [12]. The disabled states of widgets in Meta!Blast are designed the same way so they remain consistent. Users should be able to take their knowledge from other computer programs and apply it to this game, but still enjoy a unique and interesting interface.

Figure 10. Examples of lower contrast on the disabled widgets.

4 FUTURE WORK

Adding skeletal animation, particle system-based effects, and a unique and consistent interface look and feel to Meta!Blast is a work-in-progress. These additions are intended to enhance the user's learning experience and interest in the game. Future work includes visual authoring of the user interface, using particle systems to replace polygons on structural objects, and animations based on real-time motion capture, as well as evaluation of these visual effects on students' levels of engagement and learning while using Meta!Blast. Meta!Blast is being distributed to high school students and teachers for educational and research purposes.

5 ACKNOWLEDGEMENTS

This research was performed at Iowa State University as part of a research internship sponsored by NSF (IIS-0552522), the Human Computer Interaction Graduate Program, and the Program for Women in Science and Engineering during Summer 2008. We would like to acknowledge David Kabala, our graduate mentor, and Eve Wurtele, our faculty mentor. In addition, we would like to thank Stephen Gilbert, Pam Shill, Jim Oliver, Catherine Peloquin, Eva Tao, and the remaining faculty and staff of the Virtual Reality Applications Center.

REFERENCES

1. "2007 Sales, Demographic and Usage Data: Essential Facts about the Computer and Video Game Industry." *Entertainment Software Association.* (2007)
2. "Numbers." Boxoffice.com, LLC. 15 July 2008. Rentrak. 17 July 2008 http://www.boxoffice.com/numbers/
3. Anderson, E. F. "Real-Time Character Animation for Computer Games." National Centre for Computer Animation, Bournemouth University.
4. Call, A.B., Herrnstadt, S., Wurtele, E.S., Dickerson, J., Bassham, D. "MetaBlast! Virtual Cell: A pedagogical Convergence between Game Design and Science Education." *Systemics, Cybernetics and Informatics, 5, 5.* 27-31
5. Garris, R., Ahlers, R., Driskell, J.E. "Games, Motivation, and Learning: A Research and Practice Model." *Simulation and Gaming, 33,4* (2002), 441-467.
6. Gee, F.C., Browne, W.N., Kawamura, K. "Uncanny Valley Revisited." *IEEE International Workshop on Robots and Human Communication* (2005), 155-157.
7. Hartling, P.L., Bierbaum, A.D., Cruz-Neira, C. "Tweek: Merging 2D and 3D Interaction in Immersive Environments." Virtual Reality Applications Center, Iowa State University.
8. Ishiguro, H. "Interactive Humanoids and Androids as Ideal Interfaces for Humans." ATR Intelligent Robotics Laboratories, Osaka University.
9. Jones, Steve. "Let the Games begin: Gaming Technology and Entertainment among College Students." Pew Internet and American Life Project (2003)
10. Kirkley, S.E., Kirkley, J.R. "Creating Next Generation Blended Learning Environments Using Mixed Reality, Video Games and Simulations." *TechTrends, 49,3* (2005), 42-53.
11. Little, James J., Boyd, Jeffrey E. "Recognizing People by Their Gait: The Shape of Motion." *Videre: Journal of Computer Vision Research, 1,2,* MIT Press (1998), 1:2.
12. Olsen, D.R. *Developing User Interfaces.* Morgan Kaufmann Publishers, Inc., San Francisco, CA, 1998.
13. Putz, M., Klaus H. "Character Animation for Real-time Applications." Institute of Computer Graphics, Graz University of Technology.
14. Reeves, W.T. "Particle Systems: A Technique for Modeling a Class of Fuzzy Objects." *ACM Portal Grater Western Library Alliance.* (1983), 1-17.

15. Shaffer, D.W., Squire, K.R., Halverson, R., Gee, J.P. “Video Games and the Future of Learning,” *Phi Delta Kappan, 87,2* (2005), 104-111.

16. Squire, K. “Video Games in Education.” Comparative Media Studies Department, Massachusetts Institute of Technology.

17. (Figure 2) *Wall E.* Pizar Animation Studios. http://www.allmoviephoto.com/photo/2008_wall_e_009_big.html