Abstract - Educational gaming has spurred a surge of interest in the educational arena. There exists now a renewed interest in the science of gaming and its application to education. The idea of having curricular materials that are just as ‘fun’ and hard to put down as games are needed as well as intriguing. The design and development of this type of tool to contain important and solid educational content is in even greater need by educational professionals. It is common to see students spending endless hours playing games while the same is not imaginable studying. Games are engaging and fun and it is worthwhile researching their use in education. They provide the ability to process information quickly, utilize parallel processing and visual processing techniques, improve data organization, critical thinking and problem solving skills. These are just a few skills gained through playing games. The purpose of this paper is to present gaming as an educational tool and its use in Engineering Education. The paper will present and review tools that can be used in the development of educational games as well as present the Materials Education Game (MEG) environment, this is an environment that was developed for use in a Materials science educational game. This environment uses the principles of gaming for teaching a core concept in Materials Engineering.

Index Terms – Educational gaming, game development, gaming engines, 3DSMax.

INTRODUCTION

Game based education has recently become a topic of great interest to educational researchers. In 2002, the Internet and American Life Project [1] surveyed college students to “learn about their use of video, computer and online games, and to determine the impact of that use on their everyday life”. The group found that 65% of college students played, on a regular or occasional basis, some form of computerized games. The research also showed that the students used gaming as a form of socializing. It was found that students make time during the day and between classes to play their games. Most of the students surveyed associated gaming with positive feelings such as “pleasant” and “exciting”. Almost half of the students admitted that gaming sometimes keeps them from studying.

The Horizon 2006 Report [2] states that “The past year has seen a subtle shift in the way educational gaming is perceived in higher education. A number of interesting examples have shown anecdotaly that games can be very effective tools for learning. As a result, there is an increasing interest among scholars in researching the subject, not only to quantify the actual effect of using games to teach, but also to define the essence of gaming itself in order to better apply its principles to education. Educational gaming is no longer a fringe activity pursued only by extreme technophiles—it is emerging as a discipline unto itself, multifaceted and rich.”

The Entertainment Software Association (ESA) [3] reports that computer and video game sales grew 6% in 2006 with 69% of Americans play computer and video games, 38% of which are women. ESA also report that 85% of games sold in 2005 were rated E (Everyone), T (Teen), and E10+ (everyone aged 10 or above); 31% of game players were under 18 years old and 44% between the ages of 18 and 49. These statistics prove that gaming is a rapidly growing industry both in sales and in interest to the younger generation. It also provides a sound basis for researching the use of gaming in education, in particular for engineering education.

The 2007 Horizon report states that two elements are fast growing (virtual worlds and massively multi-player educational games - MMPEG). Virtual worlds are not games but an environment where the user can venture and is able to perform many types of activities. Virtual worlds can be applied to any context whereas game worlds are specific. MMPEG can take place in virtual worlds but not necessarily so. Virtual worlds such as Second Life (SL) are becoming very popular due to their “social networking; the ability to share rich media seamlessly; the ability to connect with friends; a feeling of presence; and a connection to the community”[4]. The interest in virtual worlds has increased in the past year and virtual worlds such as SL are now a place for courses to meet, train Emergency personnel, real time weather data, mathematical modeling and more. The advantage of virtual worlds (VW) is their generality. This generalization allows for diverse applications to run on VWs.

VWs in education allow the user to: address cultural and societal issues, stage theatrical productions, learn through the simulation of problem solving activities, and use role playing as a learning pedagogy (such as being a doctor working to perform certain activities). Example educational applications have been in the theatrical arts [5,6], Courses to teach VW
The term “serious games” is used when communicating about gaming environments whose main purpose is education rather than entertainment. MMPEG is being viewed as a possible environment for teaching and learning. Some examples of MMPEG have been piloted that address literature as well as immersive interactive environment development for the classroom [12]. The Media Grid's Immersive Education learning system is an application of “3-D technology and digital media that brings distance learning to a new level”[12]. Traditionally, courses deliver material via the web through simple pages and video streaming. An Immersive Education provides an interactive virtual reality environment along side elaborate digital media for course delivery. Interest in educational game development is at a high and “developments in the open-source arena are bringing them closer to mainstream adoption year by year”[4].

**WHY EDUCATIONAL GAMING?**

With 44% of game players being within the college age range, educational gaming becomes a very promising methodology for increasing retention rates in colleges and universities. It also becomes a promising venue for improving high school SAT scores (31% of game players are under the age of 18). Games are perceived as fun and exciting [3] for young people. A person willingly spends on average $50 on a game and between 50 and 100 hours solving it. This is more than a semester’s worth of work. Games are perceived as engaging and fun. A review of the use of computer and video games for learning [13] concluded that computer games “engage” and are “seductive” with their rich graphics and aesthetics providing “awe and pleasure”. Games motivate via fun, challenge, and instant gratification through feedback within the virtual environment. The review states that a greater understanding of the games culture is needed in order to find ways of designing learning games that would appeal to the younger generation while at the same time contributing to their educational process.

**ADVANTAGES OF GAMES**

Using games for education have advantages as well as disadvantages. Games enhance the player’s information processing ability, enable quick decision making on what is necessary for solving the task and what is not, and enhance data organization skills. Game players also utilize the following while playing the games: Parallel and visual processing, Critical-thinking, and Problem-solving. Games also enhance the player’s eye-hand coordination. Marc Prensky's [14] presents “ten of the main cognitive style changes observed in the Games Generation:

1. **Twitch speed vs conventional speed**
2. **Parallel processing vs linear processing**
3. **Graphics first vs text first**
4. Random access vs step by step
5. Connected vs stand alone
6. Active vs passive
7. Play vs work
8. Pay-off vs patience
9. Fantasy vs reality
10. Technology as friend vs technology as foe”

Well-produced simulation games encourage “visualization, experimentation, and creativity in finding new ways to tackle a game” [4]. They provide a means of accommodating different learning styles such as verbal, visual, logical, and aural. Games also allow the ability to introduce environments that would otherwise be too costly to experiment with. All these advantages provide a sound reason for the investigation and development of games geared towards education while retaining the “excitement” found in games for play.

**GAME DEVELOPMENT PROCESS**

Development of a game is a complex task. Andrew Rollings et al. states in his book, Game Architecture and Design, that a game is not:

- “A bunch of cool features
- A lot of fancy graphics
- A series of puzzles
- An intriguing setting and story” [15]

Rollings et al. state that a good game is described as where you can do the unexpected and win [15]. A somewhat formal process exists for game development. The development of games requires a number of steps – from initial idea inception to released product. The game development steps are as follows [6]:

- **The Game idea:** Here the developer decides on what the main game idea will be about. The game idea presents what type of game it will be and what type of audience will it be for. The specifics of how the game will be presented is decided on later.
- **Genre:** Here the decision is made of whether the game will be a ‘first-person’, puzzle, etc.
- **Story and Plot Elements:** Here the game storyline is developed. If scripted events are expected, here is where they are developed.
- **Game Design Document:** As with any project, documentation is key to the proper development of games. A major game will have a number of individuals working on it. In order to keep everyone on the same page, a game document needs to be developed. This way the programmers can intervene if the graphics designers seem too optimistic and design graphics that would be impossible to render at an appropriate rate. The game document includes elements such as: plot, storyboards, cast of characters, etc.
• The Core Engine: This is where programming starts. The programmers develop a basic engine that would perform basic tasks such as draw tile based levels and scroll them around, provide collision for the player objects, provide a simple test level build with solid colors, a static player, and one or two frame animations. The objective here is to build the core engine framework and worry about adding on levels and graphics to build the game later. The engine design needs to be modular and expandable, quite similar to a bottom up design.

• Level and Graphic Design and integration: here the programmer creates the functionality and the graphics designer creates the content. Animations and characters are created here.

• Game Play: This is where specific elements of game play are programmed, along with general functionality that is required by the game engine and any one time events.

• Computer AI: This part requires processing and physical computation for basic collision and motion. The idea here is to make the character logical in its reactions. The more collisions and motion actions of a highly graphic intense character, the higher the processing required which leads to an unplayable game.

• Story Elements: This section takes care of the programming of any scripts developed in the ‘Story and Plot elements section’.

• Sound and Music: This portion takes care of character sound and music and sound effects.

• User interface: is one of the final steps in game design. It allows the user to interface with the game elements for example: play, load/save, exit, and options buttons.

• Testing and debugging: involves alpha and beta testing of a somewhat finalized product.

Game development is by no means an easy undertaking. However, the steps in developing a game, mentioned above, involve the development of the gaming engine which is quite programming intensive. In order to use gaming as an educational aid, one cannot spend years in development time. There exist a number of open source gaming engines [17-19] that can be used to aid in this portion of the development process. This step will still involve programming but most of the collision actions and motion builders are available through the engines.

Game development requires both artistic and programming experience, and applying game theory to education becomes an even bigger challenge. Generating story lines that provide both the interactiveness and excitement of games while inherently providing knowledge is no easy feat. Games can also be used as an assessment tool to assess student learning. This can be done through the development of a puzzle like or trivia like game.

GAME DEVELOPMENT TOOLS

There exist free game development tools that allow for the development of both two dimensional (2D) and three dimensional (3D) games. Example 2D and 3D game development tools are:

• Game Editor: this is an interactive multimedia tool for game development. It has a simple and intuitive interface and allows the user to develop 2D games for personal computers and mobile devices with little programming experience.

• Sphere: is a 2D RPG (Role Playing Game) engine that allows people with not much programming experience to create role-playing games.

• 3D Game Maker: This environment uses a point and click interface. It is very easy to use and allows the user to design a game using one of 8 possible genres. The game developer can pick from pre-made characters. It is a good environment for developing games by changing set capabilities.

Game developers can also chose to develop games that are unique in their look and implementation using graphic rendering software along side gaming engines. This type of game requires knowledge of the graphics rendering environments and also requires high programming skills for the gaming engine portion of the game development. 3D graphics rendering systems are used to generate the game’s graphic environment. Two top level graphics tools are Autodesk’s 3DSMax and Maya. These tools are commercial but provide academic discounts for academic institutions.

• Autodesk 3dsMax [20]: Allows the developer to generate realistic characters, create rich and complex design visualization. It is mainly used for shading, texturing, lighting, and rendering on projects ranging from feature films to television commercials to music videos.

• Autodesk® Maya® [21]: This software is a powerful, integrated 3D modeling, animation, effects and rendering solution. It is used for modeling and creature work.

Why use 3ds Max in educational game development? 3dsMax has been used in a number of commercial game development environments in the past. Microsoft shaped their ‘ultimate racing simulator’ using Autodesk® 3ds Max® software in creating their Forza Motorsport game. Monolith used it to develop core action for their games Condemned, and Fear and Bioware uses 3ds Max to create games such as ‘Baldur’s Gate’ and ‘Star Wars – Knights of the Old Republic.

3dsMax has been used successfully in commercial game development and provides a means for developing unique game environments for implementation on gaming consoles such as Xbox 360.
Once the graphics are generated, a gaming engine is needed to provide for collisions and gaming algorithm implementation. There are a number of open source gaming engines available as well as commercial ones. Example game engines are: Delta 3D, Torque, Irrlicht, and the newly released Microsoft XNA.

- **Delta3D [22]:** is an open source full gaming engine that includes a 3D environment editor. Its architecture is built on a Game Manager, which is in charge of maintaining and organizing the key elements of the application.

- **GarageGames' Torque[23]:** provides for a whole game development approach. It provides the technology, framework, methodology, tools where you provide the creativity and the drive to make your game.

- **Irrlicht Engine[24]:** is an open source high performance real-time 3D engine. It is written and usable in C++ and is completely cross-platform, using Direct3D and OpenGL. It has a character animation system, and has a powerful, customizable and easy to use 2D GUI System with Buttons, Lists, Edit boxes

- **Microsoft’s XNA Game Studio Development Express[25]:** is a cross platform Next generation Architecture that will enable any one to make a game for the Xbox 360 and Windows XP platforms. Games developed using XNA can be distributed with the free XNA Framework to be able to run the games and need .NET 2.0, DirectX 9.0c, and Windows XP.

**MATERIALS EDUCATION GAME ENVIRONMENT (MEG)**

Materials Education Game (MEG) environment is an environment that was developed for use in a Materials science educational game. This environment uses the principles of gaming for teaching a core concept in Materials Engineering. 3dsMax was used for the graphics generation and animation. Following is a detailed description of how 3ds Max was used in the generation of the MEG environment.

The first step in MEG development was defining the concept of the environment. The objective was to create enough physical area to satisfy the game’s requirements. The environment was designed to have two levels (Figure 1). The decision to have two levels was made to visually create contrast to the user. It added more real estate and depth to the environment, instead of having one continuous room. This also built upon the concept of having a nicer “living” area on top, complemented with polished hardwood floors and a marble statue. The lower level is “dungeon-like” in appearance, grouped with stone floors, brick walls and a waterway in the middle. The symmetry of the environment was designed simply for aesthetics, indicated by the open area of the top floor that overlooks the bottom one, the steps that bridge the two sides, and the spiral staircases on either side.

3Ds Max’s options were first set to define the relative space. One division on the screen is equal to one foot. This made the environment much easier to visualize. Creating objects was relatively simple because similar processes were used regardless of the type of structure, whether it was cubic, spherical or in one of our cases, a spiraling staircase. Once the type of object to create is selected, the user can manually draw and place the object, or use the keyboard input feature to define its position. For this project, all 3D objects were created as meshes. The lighting objects differ somewhat in the development process. They can still be placed, or have their location defined via keyboard, but instead of meshes, light is vector defined by intensity and direction. A mesh is the vector defined, wireframe model of the object. Each enclosed area within the mesh is known as a ‘polygon’. Each side of the polygon is an ‘edge’ and each intersection of edges creates a ‘vertex’. The speed and ability to render an object depends on the complexity of the mesh and in turn, the number of polygons (also known as faces). The art of the designer is to efficiently manage aesthetics while limiting complexity.

Once the object is on the screen modifications can be made through the use of the modifications interface. There you can redefine the size, shape and position of an object (similar to the keyboard input), but also pick from a vast number of modifications that can be applied to the existing geometry. The modifications make mathematical changes to the vector values of the entire mesh. It should be noted that the visual precision of a modification is directly related to the complexity of the mesh. The modifications can be turned on and off independently and can have the sequence in which they are applied to the object changed. After applying all of the necessary modifications, each object was placed and rotated to create the environment. Figure 2 shows an example of an environment layout. The next step would be to add materials to the layout in order to make it more realistic.
An important aspect of 3Ds Max is the Material editor. A material is the image that is applied to a 3D object, whereas its properties define how the object looks and reacts to its outside environment. Materials are created with a series of maps, where each map represents a specific aspect of the material. Maps can be made from pictures, automatically generated geometries (such as a repetitive ‘brick’ print), or from advanced mathematical calculations that render based on the surround environment. For example, the glass covers for the lamps (Figure 3) have a complex material applied to them. First, the material’s diffuse map was made to be transparent with a smoked blue tint applied to it at 15% opacity. Second, the material’s reflection and refraction maps use the ‘Raytrace’ method, one of the mathematical functions that renders based on light rays reflected from other objects. When applied to the glass mesh, the rendered result is a clear solid material that properly renders its surrounding environment. After materials were created for each of the objects, the static environment was completed (Figure 3).

Then comes the animation process in the 3D environment. Animation in 3Ds Max is based on keyframes along a timeline. Keyframes define changes in an object’s position, shape, size, material and modification properties. Once two keyframes are defined, 3Ds Max automatically interpolates the frames in between. After the objects were created and placed, materials applied and animations defined, the scene was complete and ready to export. Figure 4 shows the animated crystal structures developed for use in the environment.

Once the environment is developed, the gaming engine is used to import the environment and program the functionality of the game. MEG algorithm functionality has been designed and is shown in Figure 5. Three levels have been designed and ready for implementation as follows:

- **Level 1: Identification:** This level places the user in the environment and gives the user a task to pick a specific structure. The user has to navigate the environment to ‘pick’ the correct structure. The scoring will allow for three errors. To complete a level, the user needs to acquire a specific number of points.

- **Level 2: Association:** This level is about associating structures to materials. It is constituted of two sub-levels. The first one is where the user would be given tasks that show a particular structure and the user would have to identify the structure name. The second sub-level is the user is given a material and would have to pick a structure that corresponds to that material. The scoring is similar to level 1.

- **Level 3: Coordinate System:** This level is again subdivided into two sub-levels, the first one is to have the user identify the coordinates of a highlighted atom within a structure. The second option is given the coordinates, have the user navigate through the environment to reach the correct atom.

The game design although simple will provide for a “proof-of-concept” design to be used in future game development by the team.
Interest in the use of game environment for education is at an all time high. The use of games for the education of our college students and indicators of its success is an interesting aspect of educational research. The upcoming task of this research team will be to implement the gaming engine. The game is designed to be an assessment type game to include a number of levels. Impact on student learning will be evaluated.

ACKNOWLEDGMENT

Special thanks go to the School of Science and Technology at Norfolk State University for providing the research funds to enable the performance of this research.

REFERENCES


[14] Digital Game-Based Learning, by Marc Prensky


[19] 3D Game Maker,


[21] Autodesk Maya


[23] GarageGames’ Torque


[25] Microsoft XNA Game Studio Development Express