TEACHING MATH AND PHYSICS BY DECONSTRUCTING GRAPHICS

Amit Shesh[1]

[1]*School of Information Technology, Illinois State University, Normal IL USA*
ashesh@ilstu.edu

Keywords: Graphics and math: high school math and science: : interactive virtual environments

Abstract: The foundations of computer graphics theory lie in mathematics, whereas most effects that add realism to a virtual graphical world are a direct simulation of the rules of physics. Complementary to the role of math and physics in learning about computer graphics, this paper proposes using graphics which has widespread appeal to directly encourage learning math and physics. We aim to use graphics to kindle interest in learning basic math, physics and by extension, computer science at the high school and college level, by directly and immediately matching theoretical concepts to how they are practically used in familiar graphics effects. We propose to achieve this by creating a virtual navigable environment of arbitrary complexity that, upon simple interaction, automatically de-constructs its constituent objects and various visual effects to reveal how they were created as well as relevant physics such as light and object behavior. We concentrate on two technical problems (a) setting up a visually appealing environment with ease for users like instructors who are not necessarily familiar with technical details of graphics and (b) de-constructing, illustrating and viewing various effects as automatically as possible so that users like students and teachers concentrate on the concepts and not how best to illustrate them.

1 INTRODUCTION

Almost every aspect of computer graphics is based on sound mathematical principles, with judicious compromise for engineering purposes. Any manifestation of graphics, like movies and games, illustrates a variety of math converging on a set of visual effects. However widespread appeal of such visual effects does not seem to translate into an equally widespread interest in math, despite such a strong dependence on it. Can computer graphics, as a "celebrity offspring" of math, encourage its learning to a wider audience?

Most college-level computer graphics courses use high-school level math. If graphics is used in a pedagogical role in such math courses, students may benefit from seeing a direct application of what they learn. For example preliminary lighting from a game can be used to illustrate the basic vector algebra that it uses, moving characters could illustrate matrix operations like multiplication, inversion, etc. Even college-level courses can use these metaphors by using water/fire in games to illustrate applications of differential equations. Besides math, computer graphics can be used to kindle interest in STEM fields like computer science where math is a strong requirement. At our university we see many students opt out of the computer science major because of its math and science requirements. Even computer science students taking their first course in graphics are often put off by the significant math content. Using graphics effects that students have seen in games and other graphics applications could persuade them to like and specialize in computer science and other STEM fields.

Like math, physics plays a significant role in many graphics applications. Achieving "realism" in computer games rests on the accuracy with which physical phenomena can be simulated. Exploiting the natural "pull" of such applications, can graphics be another reason why a student *ought to be* interested in physics? Even a simple game like *Angry Birds*(ang, 2011) uses object dynamics and projectile motion. Can dynamics be taught with the aim of implementing or at least explaining *Angry Birds*?

Interactive visualization has been used in recent years to teach math concepts (ope, 2011; cal, 2011a; cal, 2011b; see, 2011), but strictly as an enhanced illustrator. Similarly animated simulations have been used to teach dynamics(e.g. Magic Paper(mag, 2011)). We believe computer graphics can be promoted from a role of *illustrator* to that of a *motivator*, moving from visual depiction of math and physics to being the reason they should be studied. Furthermore to achieve this we believe that the pedagogical role of graphics should be more implicit, in that applications written with a different and entertaining goal can be used to illustrate and even teach underlying math and physics. In this paper we present a proof-of-concept idea of a navigable environment that may outwardly act as a game, but is able to deconstruct itself upon interaction.

2 RELATED WORK

The mathematics community has investigated using visualization to teach mathematics: we refer to a summary by Gutierrez *et al.* (A.Gutierrez and P.Boero, 2006). Many online visualizations that illustrate Math concepts are available (Java applets (ope, 2011; cal, 2011a), tools to learn calculus (cal, 2011b; see, 2011), etc.). All of the above use graphics as an illustration tool: in contrast, we propose a more direct role of using graphics as the *reason* to learn Math by showing how it is practically used.

Relevant graphics research includes tools such as JHAVE (jha, 2011) for algorithm visualization and Mathpad (Jr. and Zeleznik, 2004) to create sketch-based tools for mathematical simulations. The Graphics Teaching Tool (Spalter and Tenneson, 2006) investigates teaching graphics to non-computer science majors. Both use graphics more intimately as a teaching tool in education. Also a popular approach in using computer graphics for math and science learning has been to use virtual reality (Geitz, 1991; Taxén and Naeve, 2002; Kaufmann and Schmalstieg, 2003; Moustakas et al., 2005; Kaufmann and Schmalstieg, 2006; Kaufmann and Meyer, 2008), all of which have an explicit learning metaphor. In contrast we propose an application that makes learning more implicit. Similar to using using computer games as implicit learning tools (gli, 2011), we focus on motivating high school and college students towards enjoying math.

3 OVERALL VISION

Our overall vision comprises a 3D virtual environment that allows two types of user interaction: exploration/navigation and inspection. Although the exploratory experience would greatly benefit from more sophisticated interaction such as hand-held devices, etc. we do not regard these technologies as being critical to the teaching power of the environment.

Exploration of the 3D environment is not geared towards learning, and thus the "apparent" purpose of the environment is different. The rationale for this choice is to keep the user as interested as he/she would be when interacting with a regular game, by minimizing any negative effects of it seeming like a learning or pedagogical tool. Interesting possibilities include a first-person or a maze-based game, simulation of an existing physical environment or interactive exploration of an "alternate" virtual universe. During such navigation and exploration, the user may move or inspect objects to which the system responds by creating and augmenting to the virtual world 3D illustrations that de-construct various visual effects related to the objects. At any time the user may switch to other explanations about the same object, select another object or return to the original, "un-annotated" 3D environment. Thus we envision the entire user experience as similar to but more interactive than watching a movie on DVD, where the actual movie is augmented with "behind-the-scenes" footage detailing how certain scenes were created.

3.1 Intended Audience

We target three types of users for our system:

High-school students: High-school students learn the basic math and physics that likely persuade or dissuade them from future careers in science and technology. Typical users would be students enrolled in required and advanced math and physics courses that teach and apply linear algebra and calculus.

College students: We target college students who are interested in a major that uses math in a significant way (e.g. STEM). We believe that such a system may show such students more effectively about the "fun" and practical use of the math that they learn in the class, thereby steering them towards related careers that are in high demand.

Teachers for all above student groups: For such a system to be used in an academic environment, it must be easily customizable by teachers to specific content.

3.2 Objectives

Ease of environment creation: As it is difficult to create a single environment that illustrates a large number of examples on diverse types of math and physics, we envision that a customized environment would need to be created by teachers. For the system to be effective, this should be neither cumbersome nor should it expect the user to be knowledgeable in technical computer graphics. This could be accomplished by allowing teachers to select concepts to be illustrated and using that to select only a relevant subset of possible de-constructions in an existing environment, or customizing a toolbox which the user can then use to create a 3D environment.

As-automatic-as-possible illustrations: The system should not force the teacher to create details of the 3D illustrations, as this will make the process tedious and time-consuming. This will also require 3D placement and annotation tasks that are not suitable for the identified user groups. The system should create 3D illustrations, choose appropriate vantage points to view them and create navigation to view them as automatically as possible, without needing extensive "set-up interaction". We envision achieving this by creating a toolbox of simple 3D entities and effects that are "pre-programmed" to de-construct on command. A complete 3D environment can then be assembled using these tools.

Easy interaction and usability: Since learning is not an explicit goal of the environment we expect users to not have specific topics in mind when they explore it. Thus the interaction should be easy, and non-specific with respect to the actual math concepts or physical phenomena that it is capable of describing. Since the original purpose of the system would be explore or play, the metaphor for inspection would match that of its original application. Any customization due to specific topics would be handled by correspondingly changing the environment as discussed above.

4 A PROOF-OF-CONCEPT SYSTEM

We illustrate our above idea using a simple 3D maze-based game. The environment of this game has been created using simple implicit shapes such as spheres, boxes, cylinders and cones. The application program uses scene graphs created from XML-based input. The scene graph representation facilitates easier hierarchical modeling (i.e. creating the environment) as well as object-specific and object-relative transformations and object placement for animated 3D scenes. Using XML files provides the flexibility to create the 3D environment part-by-part and possibly programmatically by using maze-generation algorithms (Weiss, 2006). Figure 1 shows some preliminary results from this system.

The user interacts with the 3D environment by either moving in it (camera navigation), moving a specific object in it or inspecting the visual details of an object. Camera movement could be either from a 1st person perspective, simulating a more immersive experience or a fly-by perspective. Any object can be selected by clicking on it. Although our current proof-of-concept system uses manual selections to inspect various properties of an object, we envision either a gesture-based or 3D-menu-based interface that would more closely match the interaction metaphor of the game. Upon selecting an object, the following properties can be inspected.

Lighting: When any point on the selected object is clicked, the scene automatically "augments" that point with vectors relevant to its lighting (i.e. the view vector towards the original camera, the local normal vector and vectors representing the light directions for each light source from that point) (Figure 1(b)). The user can then move the vectors which causes lights to move around the environment (Figure 1(c-d). In this way the user can experiment more with how these vectors affect the lighting of the scene.

Geometry: The geometry of a composite object is shown by automatically creating an exploded view and moving the camera to a suitable vantage point. Thus each part is automatically de-constructed to show its composition using simple shapes (Figure 1(e-g)).

Additional operations that can be supported in future include:

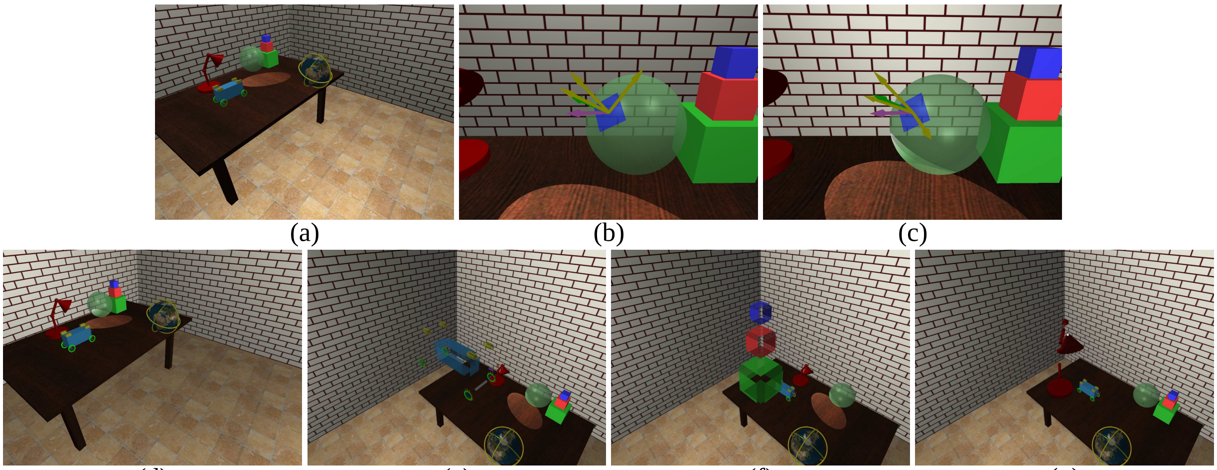
Image pasting: If the object is textured to provide a certain look, this would be de-constructed by showing the image and how it is pasted onto the object. Although this is a specific operation related to graphics, texture coordinates are often related to the geometry of the object.

Object animation: The user could select an object and force it to move in a certain straight line. In this case it would react appropriately with the environment in an animation, showing object-dynamics.

Text-based annotation: Using graphical elements alone may not create self-explanatory annotations. In future it would be possible to annotate the environment with text information. However such text would be automatically placed in a "3D-aware" manner to complement the overall scene. For example, the text may appear as writing on a white board or easel that is part of the scene.

5 FUTURE WORK

We propose creating a 3D world with hidden pedagogical capabilities, by which we hope to channel existing widespread interest in graphics and games into one in math and science. We believe using a 3D environment that de-constructs itself on command is an effective way to achieve this.



(a) (b) (c)

(d) (e) (f) (g)

Figure 1: Representative results from our proof-of-concept system. (a) A view of a virtual room in our navigable environment. (b) When the user switches to "Vector" mode and clicks on a point on the translucent sphere, the system adds relevant vectors to illustrate lighting at that point and animates the camera smoothly to a close-up vantage point. The blue quad shows the local orientation of the surface of the clicked object, with the green arrow showing the normal at that point. The pink arrow points to the original camera position. The yellow arrows point to the three sources of light. (c) Upon clicking and dragging any row, the arrow and the corresponding light move to observe the relationship of the vector to the illumination of the point. One of the room lights and the table lamp are changed in this way. (d) The first view point with the changed lighting. (e) When the user switches to "Geometry" mode and clicks on the car on the table, its exploded view is automatically generated. The camera smoothly animates to show the exploded view suitably. (f) Clicking on the stack of boxes from the new view shows it in exploded view (g) same result for the lamp, showing its different parts along with the light source. Please see the accompanying video for a real-time capture of this interaction.

The proposed framework can be customized to illustrate specific effects and geared towards specific audiences. There are many research challenges in this proposed idea: how to create and customize such an environment as automatically as possible while retaining the appeal of a game or other well-known applications and how to gear the underlying system towards users who do not possess technical knowledge of computer graphics.

REFERENCES

(2011). Angry birds. `http://www.rovio.com/en/our-work/games/view/1/angry-birds`.

(2011a). Calcsee3d. `http://web.rollins.edu/~dchild/`.

(2011b). Calculus.org. `http://www.calculus.org`.

(2011). Games for learning institute. `http://research.microsoft.com/en-us/collaboration/institutes/gamesinstitute.aspx`.

(2011). Jhave. `http://jhave.org/`.

(2011). Magic paper. `http://icampus.mit.edu/projects/NaturalInteraction.shtml`.

(2011). The open directory project. `http://www.dmoz.org/Science/Math/Education/Java_Applets/`.

(2011). Paul seeburger's dynamic calculus site. `http://web.monroecc.edu/pseeburger/`.

A.Gutierrez and P.Boero (2006). *Handbook of Research on the Psychology of Mathematics Education*. Sense Publishers.

Geitz, R. (1991). Algorithms and images: computer graphics as an introduction to science. In *Proc. SIGCSE*, pages 82–86.

Jr., J. L. and Zeleznik, R. (2004). Mathpad2: a system for the creation and exploration of mathematical sketches. *ACM Trans. Graph.*, 23(3):432–440.

Kaufmann, H. and Meyer, B. (2008). Simulating educational physical experiments in augmented reality. In *Proc. SIGGRAPH ASIA 2008 educators programme*, pages 3:1–3:8.

Kaufmann, H. and Schmalstieg, D. (2003). Mathematics and geometry education with collaborative augmented reality. *Computers & Graphics*, 27:339–345.

Kaufmann, H. and Schmalstieg, D. (2006). Designing immersive virtual reality for geometry education. In *Proc. Virtual Reality Conference*, pages 51–58.

Moustakas, K., Nikolakis, G., and Tzovaras, D. (2005). A geometry education haptic vr application based on a new virtual hand representation. In *Proc. IEEE Conference on Virtual Reality*, pages 245–248.

Spalter, A. M. and Tenneson, D. (2006). The graphics teaching tool. In *Proc. SIGGRAPH Educators program*, page 41.

Taxén, G. and Naeve, A. (2002). A system for exploring open issues in vr-based education. *Computers & Graphics*, 26(4):593–598.

Weiss, M. A. (2006). *Data Structures and Algorithm Analysis in C++*, pages 331–334. Addison Wesley, third edition.