The Graphics Teaching Tool

Anne Morgan Spalter
Brown University
ams@cs.brown.edu

Dana K. Tenneson
Brown University
dkt@cs.brown.edu

Abstract

To be a "literate" citizen one must increasingly be able to critically evaluate digital visual materials, make decisions using digital visual representations of data and ideas, and use computers to create effective visual communications. To be literate in this sense demands at least a basic understanding of computer graphics principles. Existing curricular resources, however, are chiefly aimed at students studying computer science and related fields. For the majority of students, who do not pursue technical degrees, there is virtually no recognition of the importance of basic graphics principles or even reference to their existence. To address the needs of this audience we have developed a novel interactive Graphics Teaching Tool (GTT). The GTT is a Java-based application (and applet) that offers 2D and 3D graphics in a single environment. It is expressly designed for teaching and uses a mental model-based pedagogical approach not found in commercial graphics software.

Keywords: computer graphics, data representation, education, educational software, free software, IT literacy, Java, visual literacy

CCS: K3.2 [Computer and Information Science Education]: Literacy

1. Introduction

What does it takes to be a "literate" citizen of our increasingly digital visual world? One must be able to critically evaluate digital visual materials, make decisions using digital visual representations of data and ideas, and use computers to create effective visual communications. To be literate in this sense demands at least a basic understanding of computer graphics principles, such as the difference between underlying data representations of raster "paint" and geometric "draw" programs. Existing curricular resources, however, are chiefly aimed at students studying computer science and related fields and thus use concepts and vocabulary that are foreign to most non-technical students.

For the majority of students, who do not pursue technical degrees, there is virtually no recognition of the importance of basic graphics principles or even reference to their existence. It is common to find a multitude of application-based courses in Information Technology (IT) literacy venues as well as art and design departments, but it is rare to find one that addresses the principles that would let students easily choose what types of software to use and readily adapt to new brands and versions of standard applications.

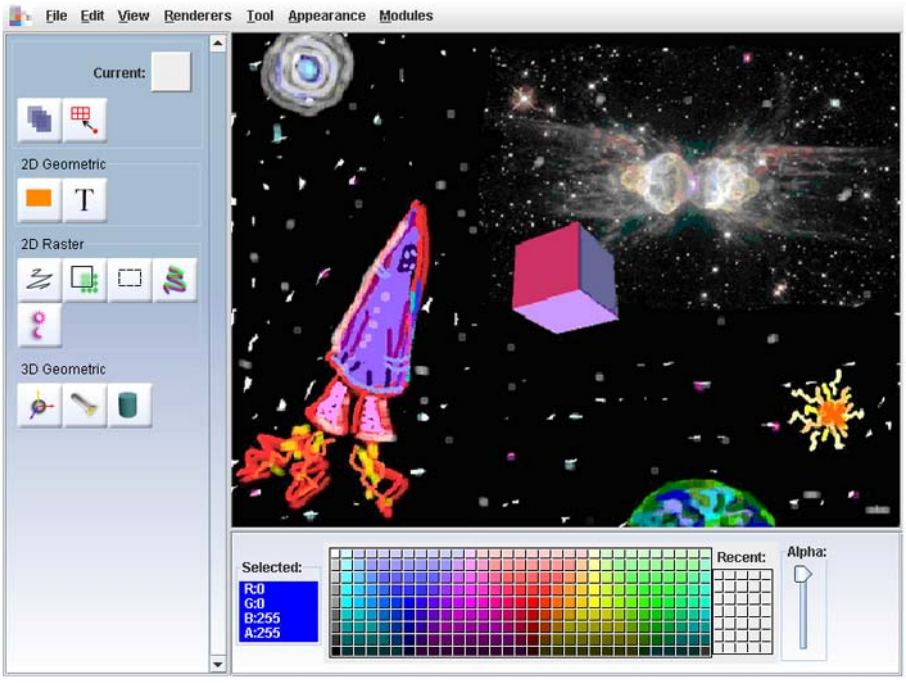

Figure 1: The main GTT interface.

Each type of graphics data representation can be used in its own layer and the layers can be easily interchanged.

For many students, reading about (or hearing a lecture on) graphics principles is often not effective. One of the authors (Spalter) has written a text that includes such content [Spalter 1999], but has found that this mode is often not optimal for visual learners. To address the need for teaching these visual principles in an engaging and visual manner we have developed a novel interactive Graphics Teaching Tool (GTT) for non-computer science majors. The GTT is a Java-based application (and applet) with a modular architecture that permits fast download times and easy modification and adaptation by others. It offers 2D and 3D graphics in a single environment which is expressly designed for teaching; the environment using a mental model-based pedagogical approach not found in commercial graphics software.

The GTT is not designed to replace professional applications or perform deadline-driven production work. It is a teaching tool that will help students become more productive users of computer graphics, a skill now in demand in virtually every IT-related and knowledge-work job they may encounter.

The GTT has been tested in an experimental course at Brown University (CS24: Visual Thinking/Visual Computing) and will be used in an NSF-sponsored Advanced Technology in Education (ATE) project to help introduce hundreds of students in the Maricopa Community College system to key aspects of visual digital literacy.

2. The Challenge

Government reports call for increased IT literacy and better education of the IT workforce [PITAC, PCAST 1997] and [FITness]. In particular, computer graphics is a highly technical discipline that is increasingly integral to many academic fields and industries using IT. Just as basic grammar skills underlie writing in all fields and basic math skills underlie virtually all the physical sciences (and quantitative aspects of social sciences), so an understanding of computer graphics fundamentals is more and more a basic literacy requirement: whether students need to edit a photo, make a business presentation, work with a 3D chemistry model, use a CAD program, or work with medical imaging, they need to be competent users of computer graphics applications.

A grasp of the fundamental computer science aspects of computer graphics is essential not only for working effectively with complex professional photo-editing and graphic design applications, but also for arriving at informed critical "readings" of computer-based images and simulations. Students and IT workers need to make informed judgments about information communicated with computer graphics, from images on the evening news, to medical diagnostic scans or computer-based simulations of mechanical, physical or biological processes.

Students interested in computer graphics who pursue a computer science curriculum learn about the concepts underlying the most popular types of graphic software. However, most IT students do not follow such a curriculum and are instead exposed to the seemingly disconnected capabilities offered by the bewildering variety of commercial application programs. Without understanding the underlying concepts, students are reduced to trial and error in trying to transfer skills within features of a single program and among different brands and types of graphics applications. This lack of conceptual understanding also makes students ignorant of the difference between limitations of a specific program's functionality or user interface and the more unchangeable constraints imposed by the type(s) of data representation being used.

For example, Adobe Photoshop offers users a single "brush" tool whose icon looks like a paint brush. The palette offered by Corel Painter, on the other hand, has icons showing not only oil paint brushes but charcoal, crayons, and magic markers. Without understanding the basic raster graphics concepts, the user might well assume that Photoshop differs somehow in nature from Painter and that only Painter can be used for "natural media" simulation. Although preset icons and tools can make such functionality easier to access, Photoshop has extremely powerful tools for designing one's own brushes and many "natural media" effects can be created. Only a student who understands the underlying raster graphics model will know that brush and media effects are almost all created by simple variations in the brush's shape, alpha mask, and compositing methods, all alterable in professional-level raster-based programs. Only such students will also realize that every operation, from the most basic brush stroke to complex filtering is simply assigning new values to pixels in the image. We aim to make this understanding available to non-experts.

Today's graphics teaching tools do not meet the basic need for graphic concepts literacy. Tutorials for professional graphics packages (see [Adobe]) present the how but rarely the why. Algorithm visualization and mathematical tools useful to computer science majors are usually too technical for most students. "Lite" versions of some programs can provide helpful scaffolding for learning a full version, but do not explicitly teach any of the fundamental concepts students should know.

3. What is the GTT?

The GTT is an integrated environment that gives students hands-on experience with 2D raster, 2D geometric, and 3D geometric computer graphics. Students interact with underlying graphics concepts via explicit models built into the GTT interface (see Pedagogical Strategies subsection "Explicit models for students to interact with"). No previous technical experience is required. Figure 1 shows the current main GTT interface with an image that includes a raster layer (an imported Hubble telescope image and by-hand drawings) and a 3D layer (currently just a white cube with colored lighting).

The GTT is based on the premise that, to be truly literate in computer graphics and be effective users of graphics software applications, students must acquire a basic mental model of how the software works and how an application's tools generate visual graphic outcomes. A mental model is the user's cognitive representation of the basic elements of a physical system along with the processes and operations that cause these basic elements to change or interact. Students who understand how graphic tools operate will know what to expect of any graphics application and will be able to use their mental models to figure out how to attain their visual goals.

Mental models have been studied extensively in the learning and cognitive sciences, beginning with Piaget's concept of schema [Piaget 1964] and culminating more recently in Gentner and Stevens' seminal book [Gentner 1982]. Mental models have led to a number of technology-based learning tools including White's ThinkerTools [White 1993]. White's work is particularly important in the present context because it is one of the best demonstrations of how complex mental models can be built up by

beginning with a simple causal model and then progressively expanding and deepening that model through purposeful interaction with computer-based tools.

4. Pedagogical Strategies

The environment offered by the GTT is powerful enough to create interesting projects, but is designed to teach concepts rather than provide professional-level production support. Learning is active and meaningful to the student because the GTT can be used in the context of real-world-style assignments for which the student has a sense of ownership and accomplishment. Constructive learning takes place as students pursue their visual goals and use different parts of the tool to create imagery. GTT users can then "see under the hood" to better understand how image data is handled in different application types using the unique "inspection tools," described in greater detail later.

The key pedagogical approaches employed are:

A constructivist vision: Students make things with the GTT rather than read about them. The GTT supports real-world-style assignments in everything from making a poster to "painting" a fine art landscape. Tool help and explanatory text is offered in context, with "how to" information contained in the tools themselves. Within defined assignments, the program aids self-correction. For example, if a student enters text in the raster layer, scaling it and changing it later will be difficult. This vision is inspired by the insight in [Jonassen 2003] that educational technology works best not as a delivery mechanism for information but as a tool that students "learn with."

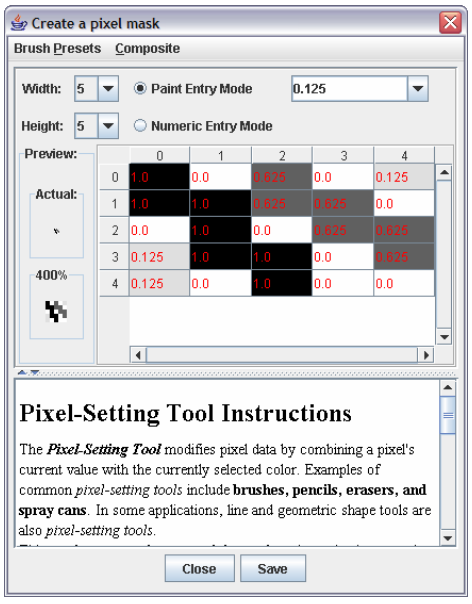

Figure 2: The pixel-setting tool interface.

Users can click and drag to "paint" a brush's alpha mask.

A simplified interface for cognitive scaffolding for both interface and concept complexity: The Graphics Research Group at Brown University, which has specialized in 2D and 3D interface research for over 20 years, has often found (as have many individuals) that users become overwhelmed with the multitude of menus and tools common in today's programs. A quick enumeration of tools and menus in the standard graphics applications for power users, Adobe Photoshop and Adobe Illustrator, shows over 50 icon-based tools and a single menu option (filters) in each program has over 100 sub-choices. The tools in 3D programs are virtually uncountable and students without any prior knowledge are almost immediately lost. In addition, little effort is made to relate tools and features across program brands or types.

As a tool for novices, the GTT provides scaffolding for both more complex interfaces (since students must eventually use them) and for constructing accurate mental models of the underlying data types. First, we have reduced the key features of all the program types, not by choosing a subset of tools from each type of application, but rather by devising new tools based on underlying graphics concepts. These two approaches: simplified interface design and condensation of tools according to underlying models, fuse to create an easy-to-use system that will prepare students for increasingly more complex environments.

For example, a single Pixel-Setting tool replaces the brush, eraser, airbrush, and pencil tools found in most "paint-type" programs, thus reducing interface complexity and focusing on the underlying concept (changing pixel values by indicating a path with a mouse). (See brush icon in 2D raster layer in Figure 1 and tool customization area in Figure 2.) The resulting single tool gives the user a simple, explicit model of brush-like tool design that should make any of the above-mentioned tools in professional programs less mysterious. (Interestingly, our Pixel-Setting Tool actually gives the user much more flexibility than standard programs and in informal testing has yielded creative brush designs.) Exploiting mental-model scaffolding concepts, we reuse the interface of the Pixel Setting Tool (in which users customize an alpha mask) in the Filter Tool (in which users customize a filter kernel) and users can see that once again they are designing a method to alter underlying pixel values. Although not planned for immediate implementation, more advanced brushes, such as one using compositing algorithms based on physical simulation of watercolor paint mixing, could build on the concepts integrated so far.

***An integrated environment*:** Unlike commercial applications, the GTT supports all the main types of graphical data representation in a single unified application. Students can work with 2D raster, and 2D and 3D geometric tools seamlessly, rapidly gaining a sense of their similarities and differences and appropriate contexts for each. Each graphics type is used on a separate layer and users can easily reorder the layers.

Perhaps more importantly, the integrated environment lets us convey mental models that will transfer among tasks because they share easily recognized cognitive elements [Bransford 1999]. For example, the methods used in transforming (moving, rotating, and scaling) 2D forms in the 2D vector module are immediately recognizable in the 3D geometric module as part of the 3D transformation interface.

In addition to the pedagogical benefits of this unique integrated approach, it also dramatically reduces overhead for teachers, support staff, and students since multiple programs need not be licensed, installed, and loaded into memory, and results from one program need not be exported and imported into others.

Explicit models for students to interact with: The GTT lets students see and interact with underlying data models for every

important concept, rather than presenting potentially misleading metaphors or assuming any knowledge of such models. For example, although the brush metaphor is useful when one starts using a professional paint-type graphics program, it quickly becomes misleading. For draw-type programs with an "ink pen" metaphor, the metaphor is significantly less useful than the brush for paint programs–drawing spline paths isn't much like drawing with an ink pen. Professional 3D programs abandon most attempts at metaphor and confront the user with puzzling visual icons and technical terms that assume some knowledge of 3D graphics concepts.

The GTT attempts to build on the mental models suggested by professional programs' interfaces and real-world experience with image-making tools, and also to lead the student into interaction with much more powerful, accurate models of image and form data creation. Students can work directly with these underlying models to "see under the hood" without having to program or know any mathematics. (High-school math will help students gain deeper understanding, but the basic concepts can be introduced with no math prerequisites.) This interaction with the underlying models occurs in creation tools such as the Pixel-Setting and Filter Tools, as well as in a tool unique to the GTT that we call the Data Inspection Tool.

The Data Inspection Tool gives students a kind of conceptual zoom tool to see how the image data is stored. In Figure 3 the data inspection tool is being used to inspect an area of raster data. The small rectangle in the drawing shows the area under inspection and the tool's interface shows the pixels arrayed in a grid. Each pixel consists of a location (shown on the grid) and R (red), G (green), B (blue), and alpha (or transparency) values. Students can

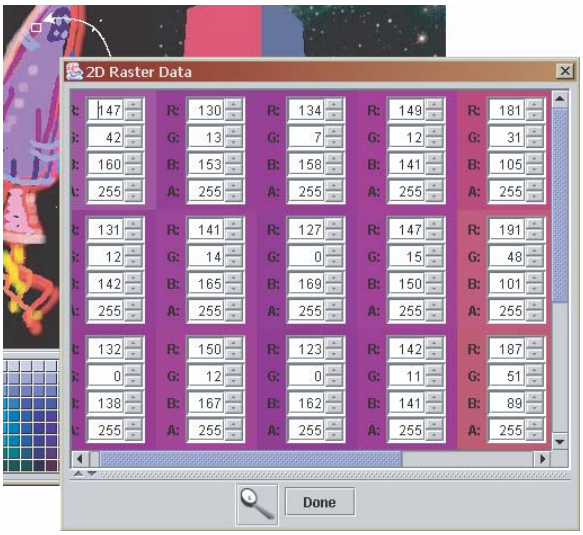

Figure 3: The GTT data inspection tool used to inspect an area of a raster image.

interact with the data, changing the values and seeing real-time updates in their image. The goal is not to create images by editing individual pixels (although one could) but to understand this type of data—and also to realize that all Photoshop's 50 tools can do is change individual image pixel values. Using the tools is just a lot faster than doing it by hand.

The underlying model is a bit more complicated for 3D data. When used to inspect 3D portions of the GTT work area, the Data Inspection Tool exposes students to the concept of matrix transformations—with no need to understand linear algebra. In fact, the matrices themselves are not shown by default—instead, students can modify x, y, z values for translation, scaling, and rotation via simple sliders and see the new values in "matrix blocks." For example, a rotation is depicted as the degree of rotation around the axis, with the advanced option of seeing a matrix of sines and cosines. For the mathematically inclined, the composite matrix of the currently selected node is depicted at the end of the chain of transformations. Transformations can easily be dragged about to change the order of transformations with the resulting changes to the scene immediately apparent.

In Figure 5, the Data Inspection Tool is being used to inspect 3D data. Below the matrix display and interaction area, objects are listed by name, starting with root. Objects can be grouped into a "group node," introducing the concept of a scene graph. This is the primary way to manipulate 3D objects and we believe that this interaction with the underlying model of transformations and their effect on a scene graph provides an invaluable level of understanding that will transfer to professional packages. It immediately becomes apparent, for example, when rotating and scaling objects in this way, that the order of operations matters (i.e., they are not commutative).

The difference between the data and its representation in an image is reinforced through the availability of renderers. By default, the GTT presents image data with standard rendering methods such as Byte Level Transfer (blitting) for raster graphics and Ken Perlin's Java 3D Renderer [Perlin] for 3D graphics. Choosing a different renderer, such as JITAC [Rieck] or a ray tracer (yet to be implemented for the GTT) shows how the same set of raster and 3D data can create markedly different images depending on the rendering style.

5. The GTT in a Classroom Situation

In the spring of 2005, we introduced the GTT in an experimental course in Brown University's CS Department called Visual Thinking/Visual Computing. This interdisciplinary class introduced students to the chief fields contributing to digital visual literacy, covering topics in computer science, media theory, design, perceptual psychology, and the "image economy." Course

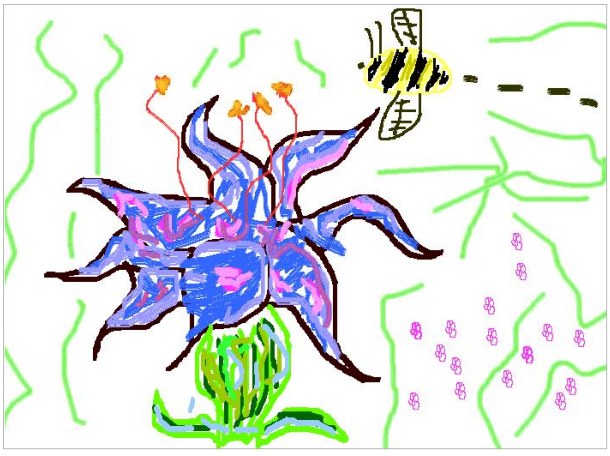

Figure 4: Example of student work made with the GTT within the first lesson on raster graphics.

materials can be seen at [CS24 2005]. The GTT was used to teach concepts in computer graphics, both as a tool for in-class demos and for in-class exercises. All of the students had at least passing experience with Adobe Photoshop and Adobe Illustrator and some were quite proficient with these programs. Only three of the 24 in the course had any CS background.

We began the first in-class project (making one's own brush tools) in what was supposed to be the middle of the raster graphics lecture. Students found the GTT so engaging, however, that it was difficult to return to the lecture and we extended the exercise to accommodate student experimentation instead. Many students expressed amazement at the customized effects they could achieve and reported that it was a compelling intellectual and aesthetic experience. They used the inspection tool to alter parts of a raster image and subsequent testing showed a consistently high level of understanding of this data type. Final feedback forms at the end of the course also showed enthusiasm for the tool**.**

6. Technical Details

The GTT is designed with reusable and interchangeable Java modules, all downloaded as .jar files and loaded as needed. The design increases speed over the Web in situations where not all modules are needed and also makes it easy for us and other educators to extend the program and create lessons that focus on specific aspects of the tool. For examples, educators choosing to use the GTT for a lesson focusing on raster graphics could choose to deploy the applet with only the raster graphics active.

Modules can be built to contain new graphics types or to provide new pedagogical tools and renderers for existing graphics types. It is our hope to develop a community of educators creating and sharing GTT modules and lesson plans.

Technical details of class structure and other aspects of program design such as import and export features can be found in [Tenneson 2003].

7. Future Work

Our future work includes both technical goals and more rigorous assessment.

Features to be added include some interface additions and changes to the raster module (for instance, adding a way to easily re-edit stored brushes and augmenting the compositing feature), a more flexible text tool for the 2D vector module, and many aspects of the 3D module (including more primitives, more powerful object definition tools, lighting, camera functions, and texture mapping). A 3D raster (i.e., voxel) module is currently on hold. In addition, although basic explanatory "tool text" has been written, text for new tools as well as some of the deeper descriptive text remains to be created.

While we were pleased with the initial response to the GTT in our course at Brown, we realize this is anecdotal evidence. A more rigorous assessment will be completed as part of an NSF ATE

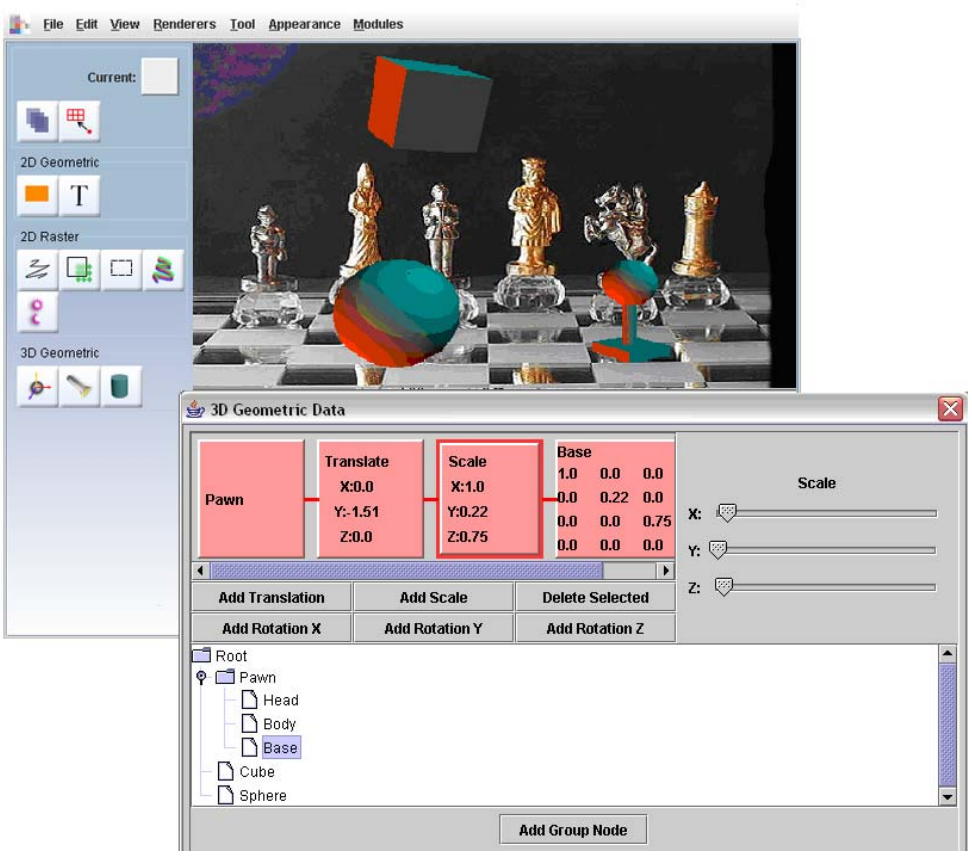

Figure 3: The Data inspection Tool reveals the scene graph for all 3D objects in the scene and shows any transformations applied to an object.

grant that will use the GTT in DVL curricular modules in introductory IT courses in the Maricpoa Community College System (for more information see [DVL 2005-2008]). Currently the largest community college system in the country with approximately 280,000 students, Maricopa is known for its innovative use of technology. We have received funding from Sun Microsystems to implement the features mentioned above and test the robustness of the application.

8. Conclusion

We hope that the GTT and accompanying materials will aid many educators in teaching the basic computer graphics concepts so integral to modern communications. We encourage readers to use the software and provide us with feedback. It is only with community involvement that we will be able to achieve our goal of making the GTT easy to use and effective for diverse communities, from high schools to community colleges to four-year colleges and universities, and beyond to continued learning environments.

Readers can find further materials and download the GTT at http://www.cs.brown.edu/research/graphics/research/gtt/

Acknowledgements

We would like to acknowledge generous support by Sun Microsystems, Inc., user interface design by Julie Kumar, and feedback from Andries van Dam.

References

[Adobe] *Classroom in a Box* and other materials. http://www.adobe.com/education/main.html.

[Bransford 1999] BRANSFORD, J. D., ET AL. (eds), *How People Learn: Brain, Mind, Experience, and School*, National Research Council, National Academy Press, 1999, pp. 53.

[CS24 2005] An experimental course at Brown University entitled *Visual Thinking, Visual Computing*. http://www.cs.brown.edu/courses/cs024

[DVL 2005-2008] The National Science Foundation Advanced Technology Education grant for "Visual Digital Literacy: Curricula and Modules for the IT Worker." http://www.dvliteracy.org

[FITness] *Being Fluent with Information Technology*, Report of the Committee on Information Technology Literacy, Computer Science and Telecommunication Board of the National Research Council. National Academy Press, 1999.

[Gentner 1982] GENTNER, D., AND STEVENS, A. *Mental Models*. Hillsdale, NJ, Lawrence Erlbaum Associates, 1982.

[GTT] A Web site with downloads and supporting documents. http://www.cs.brown.edu/research/graphics/research/gtt

[Jonassen 2003] DAVID, J., HOWLAND, J., MOORE, J., AND MARRA, R. M. *Learning to Solve Problems with Technology: A Constructivist Perspective.* Merrill Prentice Hall, 2003.

[PCAST 1997]: The PCAST (Presidents Committee of Advisors on Science and Technology) *Report to the President on the Use of Technology to Strengthen K-12 Education in the United States*, March 1997. http://www.ostp.gov/PCAST/k-12ed.html.

[Perlin] PERLIN, K. Ken Perlin's Handy Dandy Pure-Java1.0 3D Renderer, New York University, http://mrl.nyu.edu/~perlin/render/.

[Piaget 1964] PIAGET, J. Development and learning. In R. E. Ripple & V. N. Rockcastle, *Piaget Rediscovered: A report on the Conference on Cognitive Studies and Curriculum Development*. Ithaca, NY: Cornell University. 1964.

[PITAC]: (The Presidents Information Technology Advisory Committee) http://www.hpcc.gov/ac/report/.

[Rieck] RIECK, K. Java Image to ACSII Converter. Free University of Berlin. http://www.roqe.org/jitac/

[Spalter 1999] SPALTER, A. *Computers in the Visual Arts*, Addison Wesley, 1999.

[Tenneson 2003] TENNESON, D. *Overview of the Goals and Present Status of the Graphics Teaching Tool Project*, Brown University, 2003.

[White 1993] WHITE, B. "ThinkerTools: Causal Models, Conceptual Change, and Science Education," *Cognition and Instruction*, 1993, 10(1), pp. 1-100.