

Building Blast!

User's Guide

Matthew Berland

Rajiv Ramaiah

Michael Talis

Brown University
Computer Science 92
Spring, 1999

Acknowledgements

Developing a good piece of educational software is a tremendous undertaking. Neither programmers nor educators nor students can find success working alone. The nature of educational software demands that these groups work together, and for that reason, we have many thanks.

Professor Roger Blumberg's *Computer Science 92* offered us insight into the complexities of educational software development. His teaching gave us a strong understanding of work done in the field before us and offered us opportunity for dialogue on the topics. Roger's feedback was also very helpful in defining and redefining our ideas and designs. For all the attention he paid to his students of CS92, we thank him.

Andrea Tartaro served as the teaching assistant responsible for our project. We appreciate her commitment to keeping our heads in the game and our project on schedule. She also deserves a round of applause for being a great audience to our late-night comedy routines.

After every meeting with Mrs. Claudia Pietros at the Vartan Gregorian Elementary School, we walked away thinking, "I wish I had had an art teacher like that." Mrs. Pietros is one the most energetic, innovative, and helpful teachers we have ever met. Much of our work stems from her curriculum, and we hope we have done it justice with *Building Blast!*. We thank her for her willingness to try new things, her flexibility in meeting with us, and her overall excitement. None of this would be possible without her.

Lastly, we thank the students at Vartan Gregorian Elementary. In the final stressful weeks of *Building Blast!*'s development, we worked very hard but with wavering enthusiasm for what we were doing. Then we watched a few students use the program. Their excitement, humor, and learning restored our faith in our work. We couldn't have asked for a greater thanks.

Table of Contents

Introduction.....	4
Getting Started	5
System Requirements.....	5
Start the Program	5
Using the Software.....	6
Who and How Long It Takes.....	6
The Opening Screen	7
The Units.....	8
Classical and Romanesque.....	8
Byzantine and Gothic.....	8
Renaissance	9
The Basic Window	10
Navigation	10
Navigation Cues	11
Hotlinks	12
Pop-up Boxes.....	13
Troubleshooting.....	14

Introduction

Building Blast! is educational software designed to teach the architectural history of several different time periods to elementary school students. By using the software, students learn to recognize different architectural styles, those styles' well-known buildings, and the concepts associated with those periods in the history of architecture.

This software is the final project of *Computer Science 92*, a class in educational software development at Brown University. We worked with Mrs. Claudia Pietros at the Vartan Gregorian Elementary School at Fox Point in Providence, Rhode Island to develop software to meet her students' needs. *Building Blast!* is intended to supplement a core-knowledge curriculum, specifically the curriculum of Mrs. Pietros.

This documentation is intended to help the teacher use the program well with his or her students. We hope to explain the content, the rationale behind some of the program's design, and the actual operation of the program.

For further information on the development of *Building Blast!*, point your web browser to <http://www.cs.brown.edu/courses/cs092/cs92.course99.html> and choose the *Building Blast!* link.

Getting Started

System Requirements

Before running *Building Blast!*, be sure that you have the following system requirements:

- PC-compatible computer with at least 16 megs of RAM.
- Microsoft® Windows 95/98 or compatible platform
- A screen working at resolution of 640x480 or above
- CD-ROM
- (optional) Sound card with speakers and/or headphones

The software is designed to be run entirely from a CD-ROM, which means that there is no actual installation. No files need to be copied to your hard drive. However, if you find that the software runs too slowly on your system, it may in fact have to do with the speed of your CD-ROM drive. In that case, we encourage you to copy the file *Building Blast.exe* to your hard drive and run it from there.

Start the Program

1. Orient yourself on the Microsoft® Windows desktop.
2. Double click the My Computer icon.
3. Double click the drive of your CD-ROM.
4. Double click the only file on the CD, *Building Blast.exe*.

A new window will open to the main screen of the program.

Using the Software

Who and How Long It Takes

Building Blast! is designed to allow students to work by themselves, with their teacher, or with each other. In testing the software, we find that students get the most from the program by working in pairs. Working alone, a student is more likely to miss exploratory options of each screen, and this means fewer opportunities to engage in learning. When students work in pairs, one student usually catches what the other student misses. We also find that students are sometimes their own best teachers, and this peer teaching only happens when students use the software together.

Depending on the level of the student(s) and their quantity of interaction with the program, **each unit of the software** can take anywhere from 15 to 25 minutes. This estimate also assumes that the user does not go back in the unit to review any of the screens she or he observed the first time around.

The Opening Screen

Opening the program does not lead you directly into an architectural unit. Instead, you are presented with an opening screen (Figure 1) that allows you to choose which period in architectural history you want to study. Click* on either the picture of the unit's guide or the name of the unit, and the material for that section will load and you will be presented with a new screen.



Figure 1. Choose a unit from the opening screen or quit the program.

The units are arranged by historical periods which correspond to the grade levels in which Mrs. Pietros studies these periods. The third graders study Classical and Renaissance architecture, the fourth graders study Byzantine and Gothic architecture, and the fifth graders study Renaissance architecture.

With this curriculum design in mind, the vocabulary used and methods of teaching in each unit are grade/reading-level appropriate. However, do not let this scheme keep you from using any and all of the units. Students who tested the software were so excited that they wanted to do all the units. In fact, one pair of students did all three units in one long sitting. Students excited about learning are a blessing.

*Nothing in this program needs to be double clicked. Single clicks are fast and easier for younger computer users. If something does not load within five seconds after clicking, click it again. However, be patient with the application for graphics do take a while to load. Encourage students to do the same.

The Units

Below are brief descriptions of the content included in each unit.

Classical and Romanesque

- Unit designed for third-grade students, lead by Alan the Ant
- Concept of architecture introduced: what it is, who creates it, and where it can be found
- World architecture: students explore architecture from around the world by clicking on small pictures of buildings on a world map and having an information window pop up
- Shapes and forms: objects from every day life all have forms and shapes, students explore these objects, then look for shapes and forms in an architectural street scene
- Façade, horizontal and vertical lines, and symmetry: introduced by example of Greek architecture and the Parthenon.
- Rhythm: by example of Roman aqueducts, students see patterns and animation
- Colonial architecture: similarities between Classical/Romanesque architecture and Colonial architecture are recognized, local Providence homes used as examples

Byzantine and Gothic

- Unit designed for fourth-grade students, lead by Robin the Blue Bird
- Emphasis on history: students travel through Europe and through time starting in Constantinople with the emperor Justinian
- Byzantine architecture: Hagia Sophia used to exemplify this architectural style, students get to explore the building by clicking through pictures of it
- Spatial awareness: within the Hagia Sophia, students recognize the height of its dome by seeing how many elephants fit inside
- Abbot Suger: famous architect of Cathedral St. Denis in France introduced, his inspiration from Hagia Sophia
- Building materials: students explore what materials were used to build the Cathedral St. Denis by playing a small game in which they guess which material was used (6 options), click it, then see the material highlighted in the building along with a brief description

Renaissance

- Unit designed for fifth graders, lead by Boris the Beaver
- Emphasis on history: students are led to Italy during the years of the Renaissance, introduced to Florence and its citizens; students can explore a street scene, talk to the locals, Denise, Luigi and Antonio, and explore examples of Florentine architecture up to that point in history
- Leonardo daVinci: introduced as another Florentine citizen, students can ask him four different questions; he discusses his diagram of human proportions, earth/universe conjunction, Greek thinker Vitruvius, and how this all relates to architecture
- Domes: main emphasis of unit. Brunelleschi introduced along with his dome from the Cathedral of Florence
- Architecture construction and geometric concepts: how does one build a dome? students take an in-depth exploration of how it is done through careful visual demonstration; concepts of horizontal construction, herringbone pattern, rib supports, and double-shell construction introduced
- Review: because the unit goes into a lot of detail, students have a one-screen reminder of all that they explored

The Basic Window

Below is the basic window that you will find throughout all three units. In Figure 2, each part of the window is labeled so that you can refer to the subsequent sections of this document that describe these areas in detail.

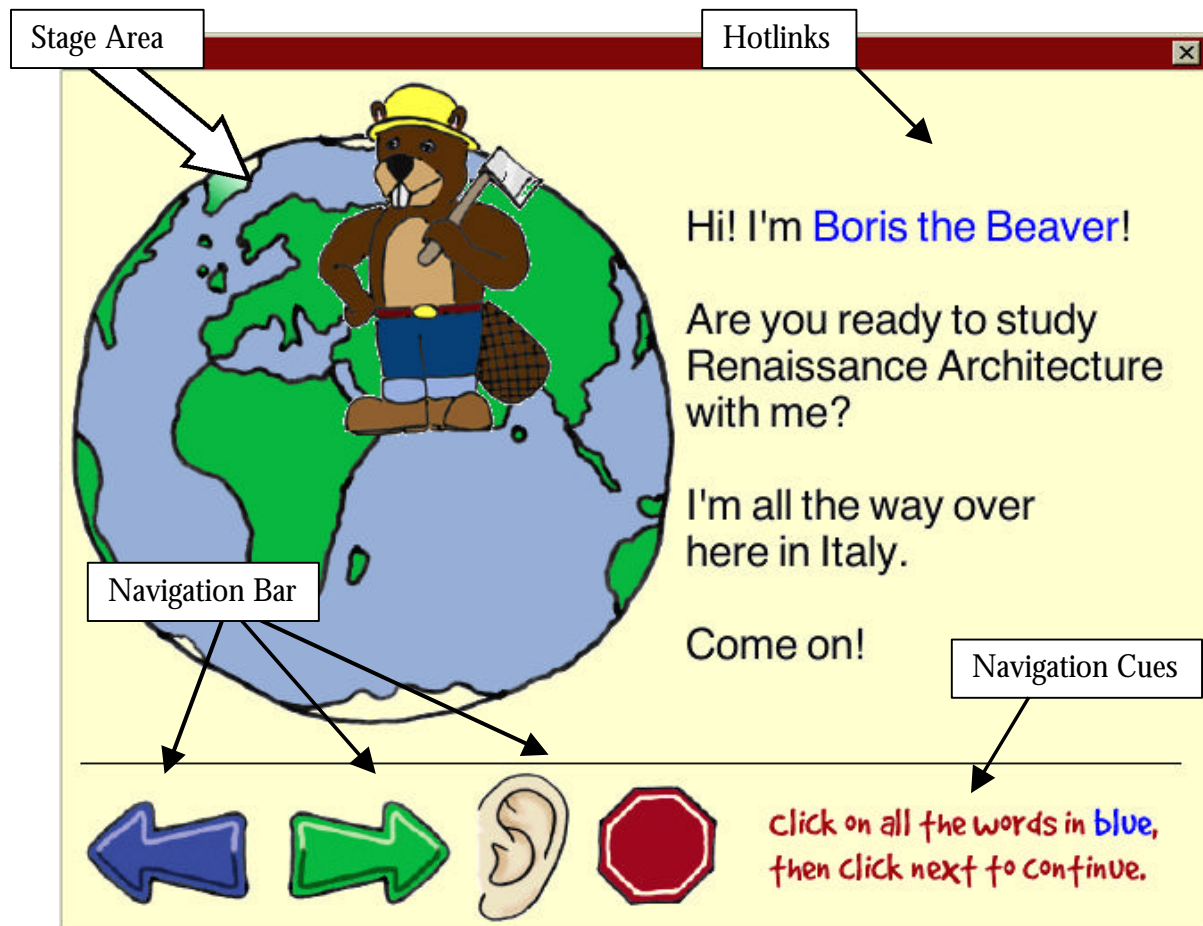


Figure 2. The *Building Blast!* window.

Navigation

Regardless of which unit is selected, there is only one central scheme for navigating the program. Each unit walks you through a series of interactive slides. Some slides are levels deep while others are simply basic informative screens. You navigate through the main slides by means of clicking the navigation buttons. (The keyboard is not used at all.)

A row of simple navigation buttons (Figure 2) allows you to work at your own pace through the program.



Figure 3. The buttons allow you to navigate the program easily.

When the mouse is over one of these buttons, the button highlights, which means that you can click on it. You need only click once, and the buttons will do the following.

The **blue arrow** pointing left will move back one screen.

The **green arrow** pointing right will move forward one screen.

The **ear** will read the text on the screen out loud. (Note: if you move screens or close a window while the text is being read, the sound will automatically stop. This is so that you don't feel caught up in the audio.)

The **stop button** will quit the unit and return to the opening screen, from where you can quit the program.

Navigation Cues

At times you may not recognize your options for navigating and/or interacting with the program. For this reason, we designed navigation cues (Figure 4).

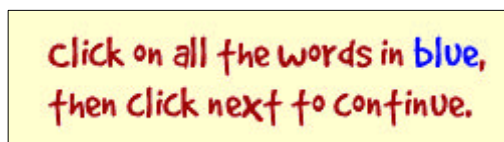


Figure 4. The Navigation Cues keep you from feeling lost.

Navigation cues are small notes in red text at the bottom right hand side of your screen that instruct you on your options. They are in the form of an imperative but are in no way meant to insist that you do anything. Students oftentimes appreciate a gentle encouragement here and there, and in testing the software, we found that without these cues, students occasionally missed the opportunities for interaction with the program.

Hotlinks

Mentioned above were interactive screens that can either be simple informative screens or screens layers deep. Those that are more interactive than others will encourage this interaction by means of hotlinks.

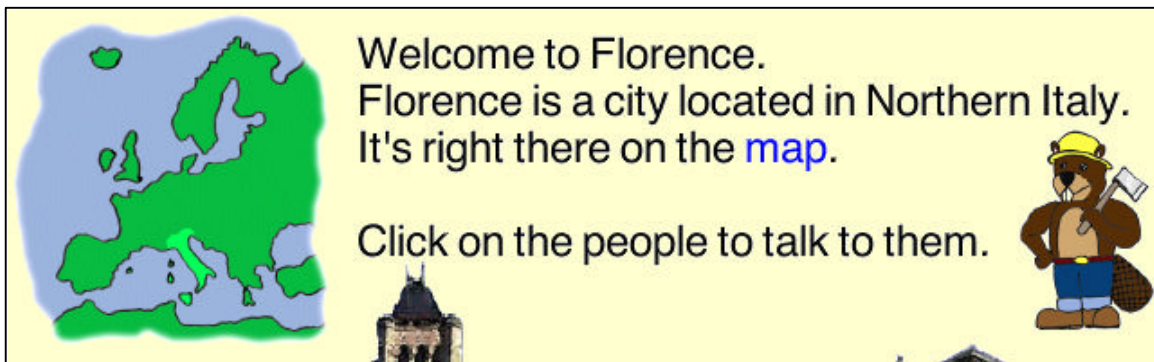


Figure 5. Hotlinks keep you moving.

Most of the hotlinks in this software take the form of blue-colored text (Figure 5). When the user clicks the blue text, many different things can happen. A new window can pop up, such as in the case of Figure 5 when “map” is clicked, going into more detail about the highlighted text. The hotlink may take the user to a different part of the program altogether, or the highlighted text itself might be repeated out loud for the student. Hotlinks can also take the form of images that change when the mouse rolls over them. For instance, in Figure 5, when the mouse rolls over the map of Europe, the map becomes darker. This indicates that if you were to click that area, you would be taken elsewhere. This is also a common way to indicate a hyperlink.

Ultimately, the hyperlink scheme becomes familiar and rather instinctive to users fairly quickly, if it already isn't so. This hotlink scheme is similar to that of hyperlinks found on the World Wide Web, and we chose to use this scheme because of its familiarity and ease of use. Although hotlinks may present information in a less linear format and sometimes be frustrating to the user, this structure is valuable because it then becomes the responsibility of the user to construct knowledge in a way that enables him or her to understand it.

Pop-up Boxes

As was mentioned earlier, one of the possible reactions of a hotlink is the display of a pop-up box (Figure 6). These boxes do not usually offer opportunities for interaction; instead, they do a bit more explaining and demonstration of ideas. Pop-up boxes cannot be moved nor their size manipulated, and you cannot proceed within the software until you have closed a pop-up box. To close a pop-up box, you simply click once anywhere inside it. Clicking outside the box, except for on the ear icon, will do nothing. If you do click the ear icon while the box is open, the text of the box will be read to you. All this aside, you will very quickly get used to pop-up boxes and won't think twice about how to use them.

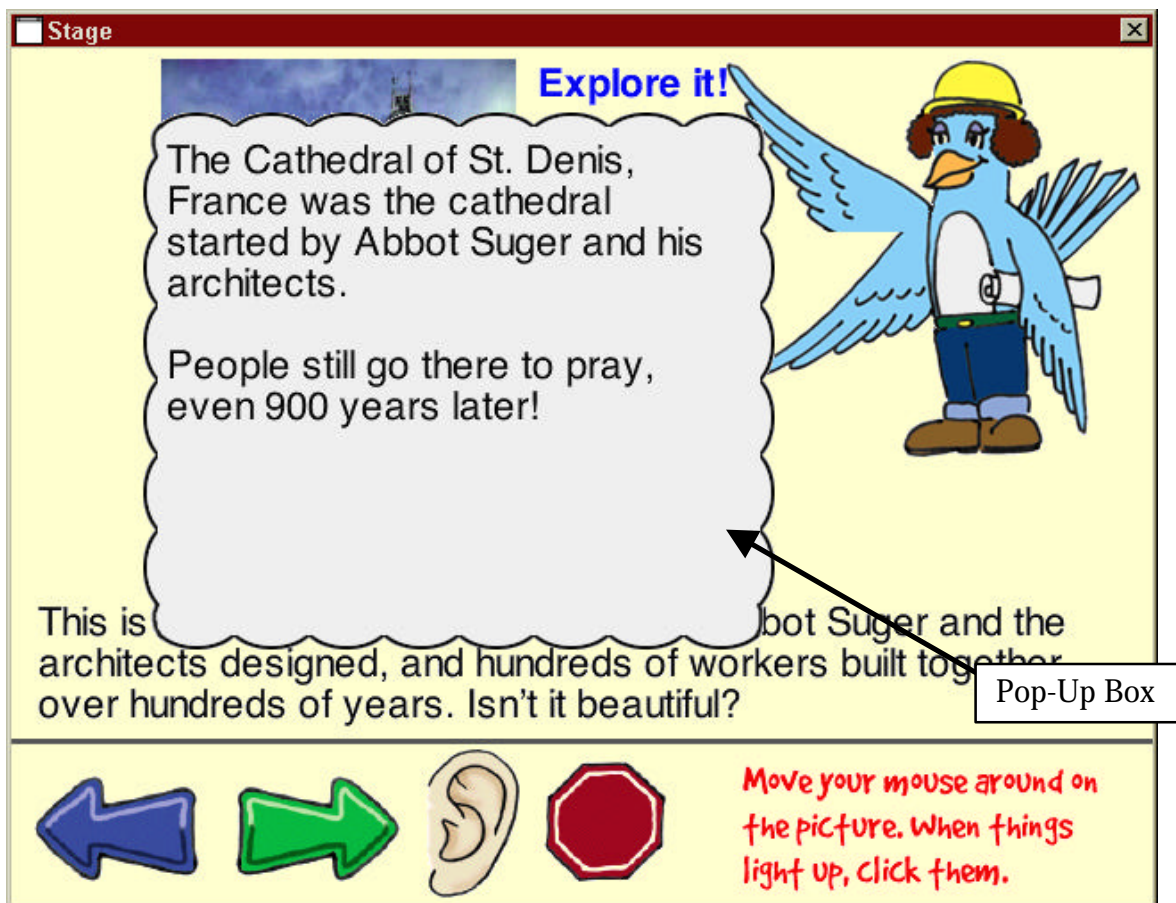


Figure 6. Pop-up boxes are simple, but helpful

Troubleshooting

Considering the simplicity of this software, not much can be said in the way of troubleshooting. However, as with any piece of computer software, bugs are bound to turn up. We have tried our hardest to eliminate those bugs through testing, revising, and retesting, but you may find something that we missed.

Firstly, we encourage you to be patient with the software. If screens don't change and things don't pop up instantly, it is probably more a matter of the computer system than the software itself. Do not click frantically trying to solve the problem. This will only make it worse. Secondly, make sure that no other running applications are hogging your system's resources. Keep things simple, for if you start running more than one piece of software at a time, it is likely that they will do battle over your computer's precious resources such as processing power, screen, sound and more.

If, after having checked these things, your software still does not seem to be working properly, we encourage you to restart the unit you were working on. Simply click the quit button, go back to the main screen, then reselect your unit. This should reset any bugs that might have been fouling things up. If this persists, take it to a grander scale and restart your computer.

If you do this and problems persist, we either created bad software or your specific copy of *Building Blast!* is in some way corrupted. In this case, it is best to simply not use that copy of the software anymore. Find a working copy of the software and use that CD.

A page worth of troubleshooting tips invites you to think that the software you have is poorly designed and executed. We do not believe that it is; in fact, we expect that you will have no problems with the software. However, we do want to be sure that if problems arise, you have a way of solving them.