

# Maze Suite 1.0: A complete set of tools to prepare, present, and analyze navigational and spatial cognitive neuroscience experiments

HASAN AYAZ, SARAH L. ALLEN, STEVEN M. PLATEK, AND BANU ONARAL  
*Drexel University, Philadelphia, Pennsylvania*

Maze Suite is a complete set of tools that enables researchers to perform spatial and navigational behavioral experiments within interactive, easy-to-create, and extendable (e.g., multiple rooms) 3-D virtual environments. Maze Suite can be used to design and edit adapted 3-D environments, as well as to track subjects' behavioral performance. Maze Suite consists of three main applications: an editing program for constructing maze environments (MazeMaker), a visualization/rendering module (MazeWalker), and an analysis and mapping tool (MazeViewer). Each of these tools is run and used from a graphical user interface, thus making editing, execution, and analysis user friendly. MazeMaker is a .NET architecture application that can easily be used to create new 3-D environments and to edit objects (e.g., geometric shapes, pictures, landscapes, etc.) or add them to the environment effortlessly. In addition, Maze Suite has the capability of sending signal-out pulses to physiological recording devices, using standard computer ports. Maze Suite, with all three applications, is a unique and complete toolset for researchers who want to easily and rapidly deploy interactive 3-D environments.

Stimulus presentation and measurement of the responses of subjects, using various neuroimaging modalities, play a central role in neuroscience research. Because of the substantial need, there are many commercially available comprehensive stimulus presentation systems on the market. E-Prime (Psychology Software Tools), Stim (NeuroScan), SuperLab (Cedrus), and Presentation (NeuroBehavioral Systems) are well-known experimental psychology research applications. An editor with which to design experiments in which stimuli are presented in a PowerPoint-like fashion is a fundamental feature of stimulus presentation systems. However, it is also essential that stimuli should be presented with accurate timing and that responses from subjects should be recorded with reaction times and types of responses. Accurate timing has become a sizable challenge with the advent of multithreaded modern computer systems. These complex systems are formed by independent layers (kernel, application) and are nondeterministic and, thus, not real time. Timing accuracy has been studied on Macintosh operating systems (Chute & Westall, 1996; Westall, Perkey, & Chute, 1986). Microsoft Windows operating systems have not been studied as much. A recent report by Chambers and Brown (2003) provided promising results for millisecond resolution with the QueryPerformanceCounter (Microsoft, 2006) method. This method is based on a central processing unit (CPU) counter that is incremented at each CPU cycle, using an RDTSC (read time stamp counter) benchmarking instruction. Thus, the counter fre-

quency is almost as high as the CPU frequency. To use this method, first, a counter frequency, which is different on each computer, is acquired, and the counter value difference between two time points is converted to milliseconds, using that counter's frequency.

## MAZE SUITE: A New Set of Tools for Investigating Spatial Cognition

In current research, navigational and visual-spatial skills are investigated primarily through the use of behavioral tasks. For example, spatial tasks involving mental rotation and object location, as well as spatial working memory tasks, have been used to dispute findings of differences in the abilities of males and females. These contradictions were often due, in part, to the inclusion of many different aspects of *spatial abilities* in studies of spatial cognition (Caplan, MacPherson, & Tobin, 1985; Voyer, Voyer, & Bryden, 1995). For example, it may be that males are better at creating mental maps to manipulate their environment, whereas females are better at using objects to navigate space. The field is in great need of a technique that enables researchers to better define and control environments while keeping the ecological value of the spatial surroundings. This would limit overlap in the abilities being measured (e.g., allowing researchers to measure brain activation in navigational tasks without the distraction of external stimuli)

---

H. Ayaz, [hasan.ayaz@gmail.com](mailto:hasan.ayaz@gmail.com)

and would enable investigators to target specific spatial cognitive skills.

Maze Suite is a set of applications that enables researchers to have full control over designing, running, and analyzing spatial navigational environments for use in cognitive neuroscientific experiments. Until recently, there has been a lack of a common, task-independent 3-D software tool for navigational studies. Many neuroscientific experiments have involved pencil-and-paper tasks, such as the Vandenberg Mental Rotation Test (Vandenberg & Kuse, 1978), whereas some groups have developed task-dependent custom software for their experiments, such as Astur, Tropp, Sava, Constable, and Markus (2004). In such virtual reality studies, experts are needed to program the necessary environment. Also, NeuroBehavioral Systems, Inc., has made an effort to include programmable 3-D functionality in their well-known product, Presentation. Presentation 3D is Version 0.6, and its features appear to be more oriented toward the display of 3-D objects, as opposed to navigation in a virtual environment. Necessary features for 3-D engines, such as collision detection, are missing, but another major drawback is that users need to write a custom code called PLC (as defined by NeuroBehavioral Systems, Inc.), which requires more time to develop scenes and train users.

Maze Suite, unlike other systems, enables scientists themselves to create extendable 3-D environments in which research subjects can navigate. Because of exponential growth of the video game market, many comprehensive 3-D content generation and rendering tools, such as Maya (from Alias), 3D Studio Max (from Autodesk), and Virtools Dev (Virtools SA), and advanced 3-D video games such as Quake (from ID software), have been developed. However, there are drawbacks to these tools in experimental deployment. First, they are not designed for this purpose (high-quality modeling tools are not real time) and lack specific features, such as precision timing,

synchronization signaling, behavior tracking, and tools for analyzing logged data. Moreover, they are expensive. But the most limiting drawback for experimental social scientists is likely to be the difficulty with usage—for example, the need to master scripting languages. Tools such as 3D Studio Max, which is a state-of-the-art modeling tool, have a large number of functions, and it takes expertise in programming to be able to use the application. However, in Maze Suite, the aim is to incorporate a user-friendly and simple editor yet have enough functionality to create plain 3-D environments for experimentation. Below, we will outline in fuller detail the component applications of Maze Suite.

### Experimental Setup

Maze Suite is composed of three integrated applications that help researchers create, conduct, and analyze spatial navigational experiments. Figure 1 summarizes these tools and shows a common experimental setup for use in conjunction with functional near-infrared spectroscopy (fNIR, a newly emerging neuroimaging technique) data acquisition with a stimulus from Maze Suite. The application in Maze Suite that one begins with is MazeMaker, which is the editor program for creating 3-D maze environments from scratch. In the experimental design phase, MazeMaker is used to create maze files, and then these maze files can be placed appropriately in an experimental protocol with the MazeListBuilder tool within MazeMaker.

During the experimental run, MazeWalker processes the experimental protocol, visualizes mazes interactively, and keeps track of the activities of the subject. In Figure 1, MazeWalker is depicted as running on a presentation PC in front of the subject. MazeWalker sends time sync signals to the fNIR data acquisition computer via a serial port (RS232). Using a standard serial port allows Maze Suite to interface with data acquisition systems, such as fNIR,

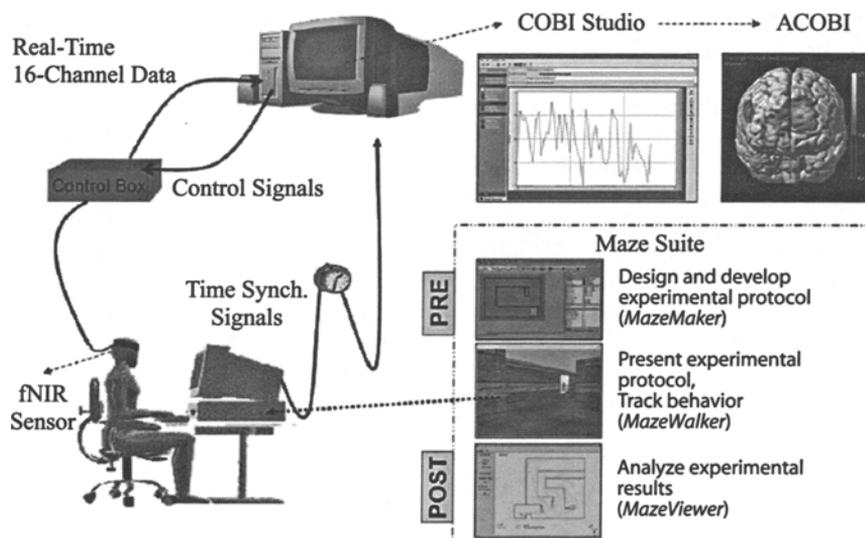


Figure 1. Functional near-infrared spectroscopy (fNIR) experimental setup for a navigational study.

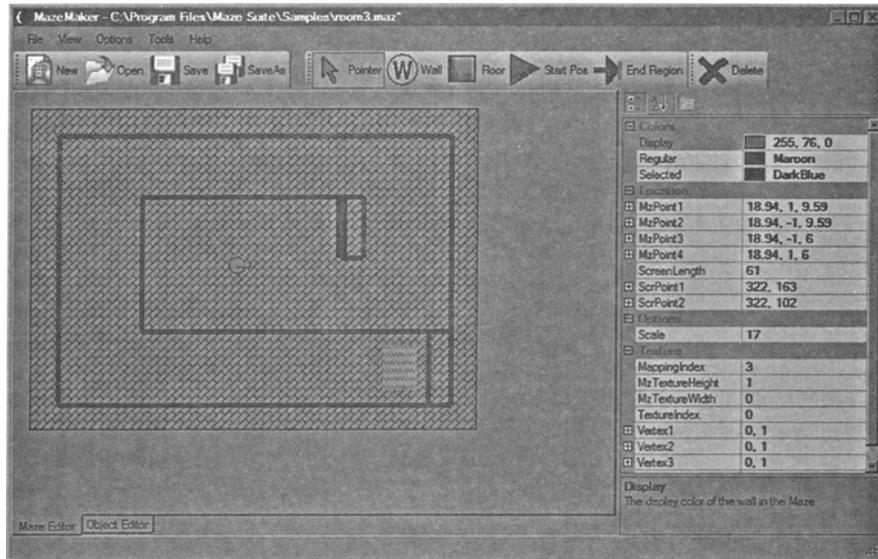


Figure 2. MazeMaker main window.

EEG, and an eyetracker. In Figure 1, Maze Suite is shown as being used in parallel with the fNIR device that is monitoring the oxygenation level in the prefrontal cortex, but Maze Suite can as well be used with MRI devices with virtual reality goggles and a joystick. After the experiment, MazeViewer is used to analyze the path and timing of the subject. COBI Studio and ACOBI (Ayaz et al., 2006) are applications developed in the Optical Brain Imaging Lab of Drexel University, and they are used to acquire, visual-

ize, and analyze fNIR data. Izzetoglu et al. (2005) have provided more information about fNIR data analysis.

**MazeMaker**

MazeMaker is the editor application of Maze Suite for creating and editing experiments. The main window of MazeMaker is composed of a drawing area, a properties list (vertical right-hand side), and toolbars (horizontal, on the top). The main window is shown in Figure 2.

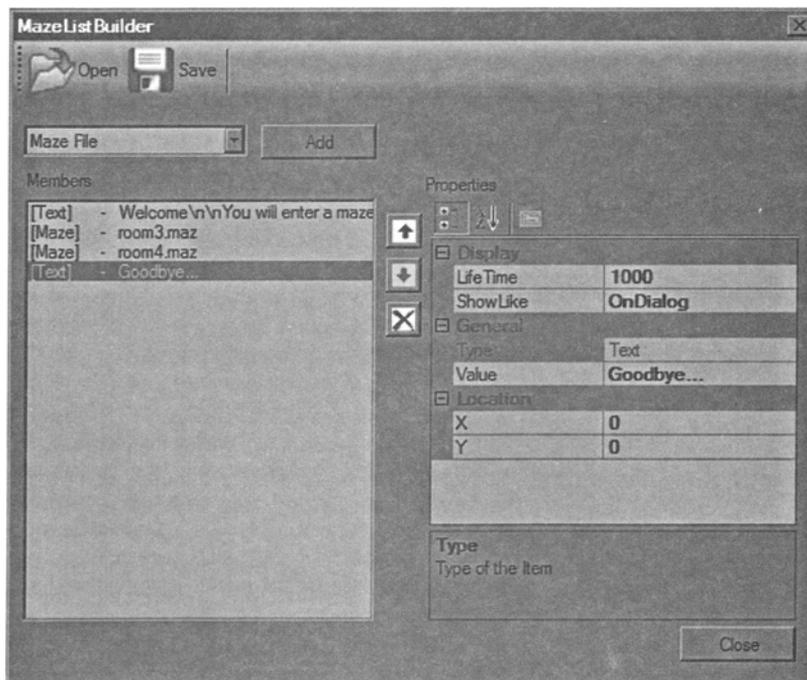


Figure 3. MazeListBuilder tool.

Mazes are built from a bird's-eye view. The drawing area is parallel to the maze floor plane, and thus planes and objects that are perpendicular to the floor are represented with lines and 2-D shapes that are the cross-sections of the original objects. All the items on the drawing area instantiate objects, and their type-specific properties can be viewed and edited from the right-hand side properties menu. Common properties are color, texture, and 3-D space coordinates.

The motivation behind MazeMaker is to craft an easy-to-use application for the creation of 3-D virtual environments. One can create 3-D structures as easily as one can edit pictures in the popular Microsoft Paint application. Unlike scripted presentation systems, this graphical interface is much easier to learn and use. A wall can be created with two clicks of the mouse to specify the two edges of the line. The coordinates of the wall in 3-D space are instantaneously calculated from the screen coordinates of the line. Also, the user can select the object and reset the 3-D coordinates if he or she wants to achieve varied effects.

MazeMaker also incorporates a MazeListBuilder tool that enables researchers to combine multiple mazes and text messages to generate experiment protocols. The tool is shown in Figure 3. MazeListBuilder can create the feel-

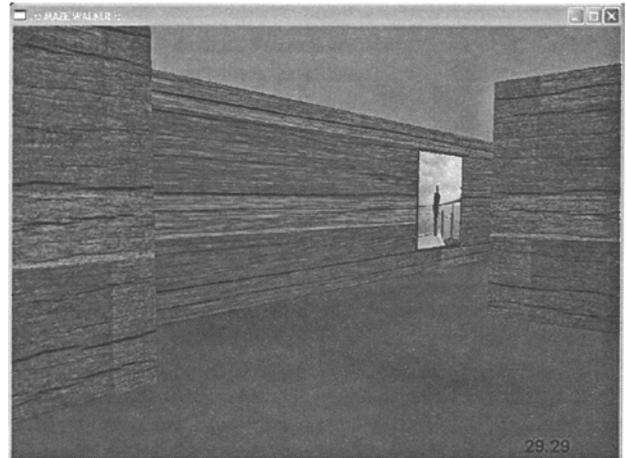


Figure 5. MazeWalker rendering screen.

ing of moving between rooms/floors in a maze. It can also present instructions or any feedback text between mazes (e.g., "Congratulations! You Completed the Maze!").

MazeMaker has been developed with C# using .NET Framework Version 2. Each maze and predefined items (wall, floor, start position, and end region) are implemented as individual classes. 3-D coordinate space and transformation calculations are also implemented in separate classes and instantiated to invoke their methods.

### MazeWalker

MazeWalker is the enabler of Maze Suite, in that it is the application that renders mazes created with MazeMaker. MazeWalker is a Win32 application that uses OpenGL to visualize mazes. MazeWalker has been developed in C and uses Win32 API calls for interaction with the operating system. This minimalist approach is followed in order to reduce processing overhead and increase rendering performance. MazeWalker is composed of a plain 3-D engine that simulates a virtual physical environment whose properties can be customized in MazeMaker. This rendering engine listens to inputs from the user (keyboard, mouse, or joystick) and renders the virtual environment in real time.

**Timing.** MazeWalker uses the QueryPerformanceCounter method for timing, as described by the Microsoft (2006) reference document. The study by Chambers and Brown (2003) validated that this method provides submillisecond resolution timing and is the highest resolution timing method available on Windows-based personal computers. QueryPerformanceCounter uses the CPU's time stamp counter that is incremented at each cycle. Using high-frequency processors, the theoretical resolution is on the order of nanoseconds. With this method, the first frequency of the timer is acquired, and this frequency is used to convert tick numbers from the counter to milliseconds.

**Synchronization signals.** During the experimental run, MazeWalker processes the experimental protocol created by MazeMaker. The experimental protocol is formed of a list that contains, in mixed order, various numbers of

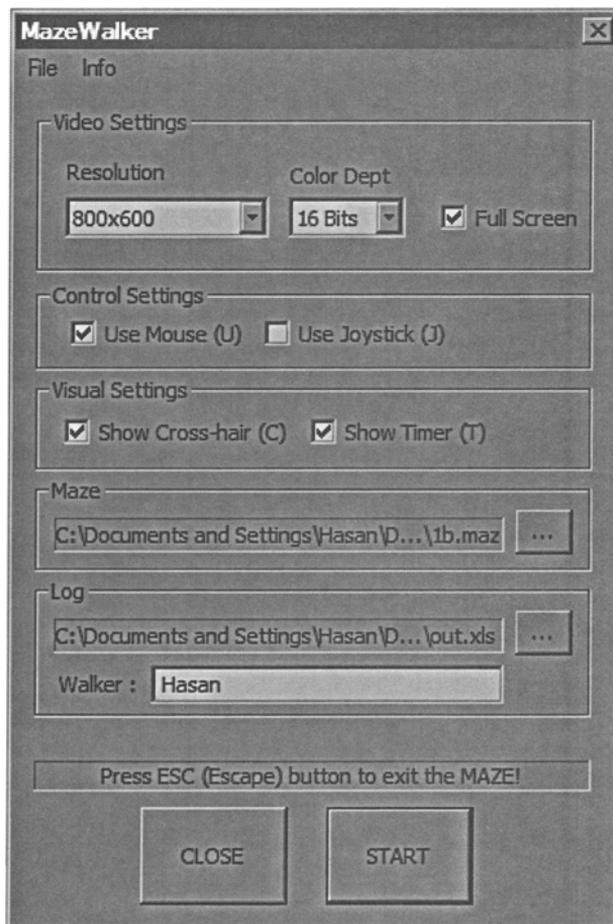


Figure 4. MazeWalker initialization dialog.

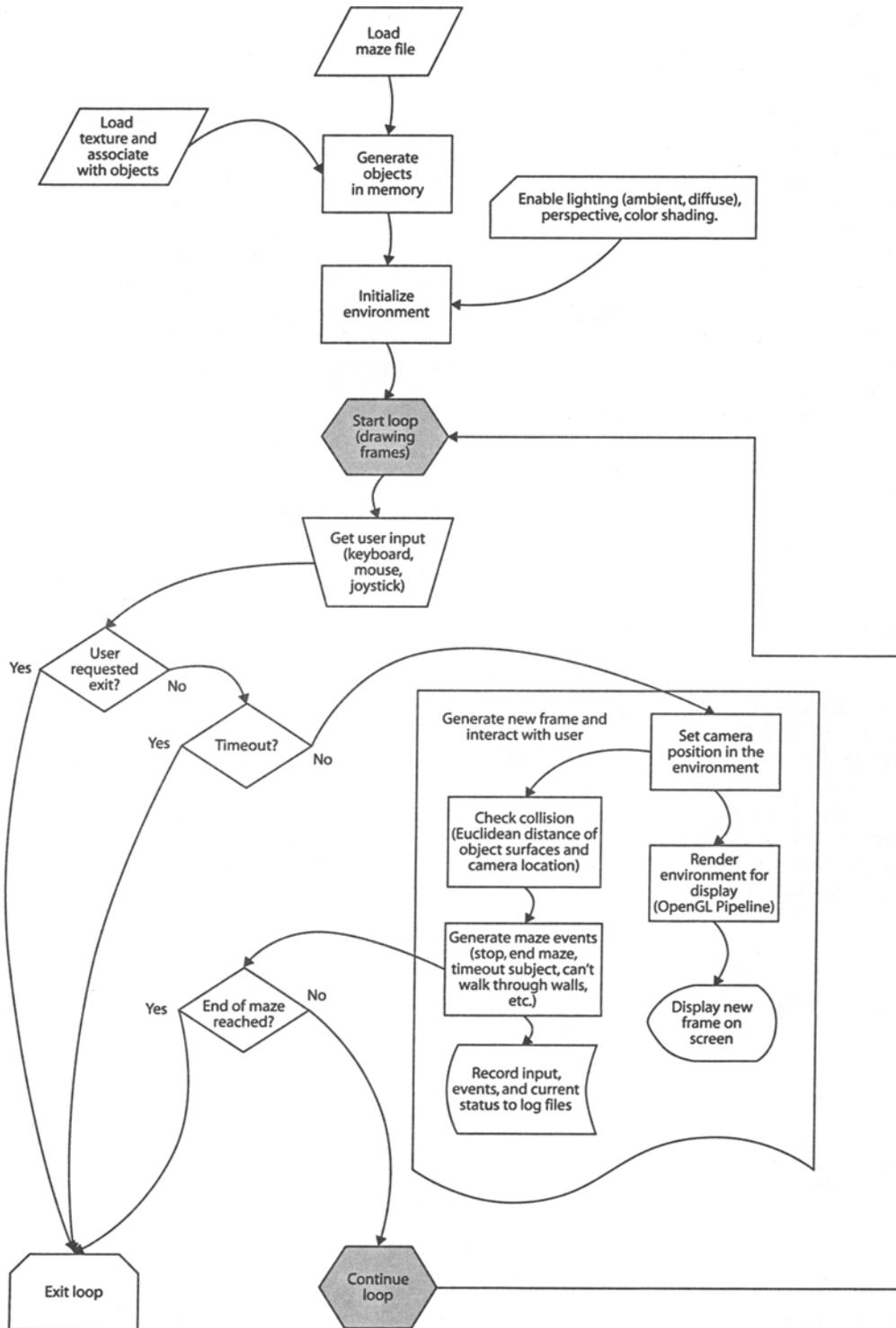


Figure 6. Flow chart depicting basic functionality of MazeWalker with a single maze.

mazes (interactive, 3-D) and text messages (interactive, dialog based). During processing of the experiment protocol, MazeWalker can send time synchronization signals (markers) that mark certain milestones of the experimental protocol to a data acquisition system. In Figure 1, as was described in the Experimental Setup section above,

MazeWalker is depicted as running on a presentation PC in front of the subject and sending time synchronization signals to the fNIR data acquisition system COBI Studio, running on another computer.

A marker is a single byte of integer value with a unique event definition. MazeWalker can be set to send markers

for various events during an experiment by using the initial settings dialog. The serial com port can also be specified. MazeWalker uses a standard baud rate of 9,600 and a byte size of 8, with 1 stop bit and no parity.

**Features.** When the MazeWalker application is started, an initialization dialog box is reached, as shown in Figure 4. Necessary settings—video, control, visual, maze, and logging—can be set here. The start button closes the dialog and starts the rendering window (see Figure 5). Subjects can use keyboard cursors, a mouse, or a joystick to navigate in the maze.

Navigation in the maze is simulated by the translation and rotation of the camera in the maze space, which is a virtual space. This is achieved by multiplying the location vector by appropriate matrices. Collision detection has been implemented so that the subject cannot go through objects. The collision detection algorithm uses the geometry of an object's location vectors and a calculation of the distance of a point to a plane.

MazeWalker is composed of a compact 3-D engine that simulates a virtual environment. A condensed flow chart of the application is given in Figure 6. A single maze session is started by first loading the maze file and textures. All objects and textures are created as a dynamic list array. Environment conditions, lighting, and shading parameters are set before the application enters the main loop. In this loop, user input is pooled from interfacing devices, such as the keyboard, mouse, and joystick. If the users have not requested exit and timeout has not occurred, a new frame is generated and displayed. Current status information is appended to a log file, and time sync information is sent to the remote computer if "triggering maze event" has been fired. If the end of the maze is not reached, a loop starts from the beginning to

generate a new frame and continues until an "exit case" is satisfied.

The steps described in Figure 6 are a visualization of only one maze file in a typical experimental procedure; there are many maze files in the list. MazeWalker can process a maze list by visualizing mazes one after another. In that case, there is a larger loop that reads a maze file to load and visualize it and then switches to the next maze in the list. There can be text display and pauses for a pre-defined amount of time between two mazes.

Textures can be handy to improve maze reality. One can go out, take pictures of a real-world environment, and then import them into the maze environment. All common image formats (such as bmp, jpeg, gif, png) can be imported. Image editing programs (such as Microsoft Paint or Adobe Photoshop) can be used to create custom images that can realize special effects when imported in maze space.

### MazeViewer

MazeViewer is the analysis tool of Maze Suite (see Figure 7). MazeViewer can open log files created by MazeWalker and enables researchers to analyze the behavior (e.g., time to completion of the maze, number of errors) of the subject in the maze. Log files contain information about the path that a subject has traveled, as well as where the subject looked at each time. All actions are time tagged in millisecond resolution. The "play" feature of MazeViewer can display the animated time-correlated path of a subject, starting with the subject's entering the maze and continuing until exiting. MazeViewer also provides a quick way to view the map of mazes.

The LogParser tool in MazeMaker can segregate multimaze log files into individual maze traces and allows

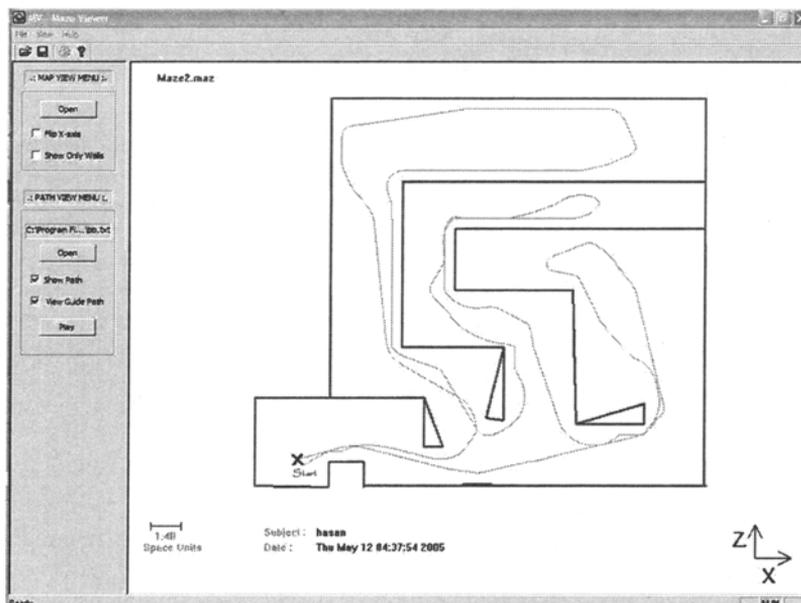


Figure 7. The path a subject has traveled can be displayed by MazeViewer.

MazeViewer to trace a single maze with path and timing. Finally, a review of the segregation process is saved into a summary file that includes a list of all the mazes, entering and exiting times, and total time spent in each maze. This summary file can be imported to statistical analysis tools, such as Excel and SPSS, for further analysis.

### CONCLUSION

In this article, a new set of tools for spatial and navigational research experiments has been introduced. Maze Suite is copyrighted by Drexel University and can be downloaded from the official Web page of Maze Suite ([www.hasanayaz.com/maze](http://www.hasanayaz.com/maze)) at no cost for noncommercial use.

Maze Suite enables investigators to more objectively measure spatial cognition. Recent use of this technology has been explored at the Evolutionary Cognitive Neuroscience Laboratory at Drexel University in research on sex differences in navigational abilities. Maze Suite allowed our team of investigators to develop complex realistic mazes through which subjects were asked to navigate. The researchers utilized Maze Suite's ease of use to import and export objects enabling the study of spatial working memory as a tool for spatial navigation and for direct comparison with "pure" navigational capacities. Future studies would benefit from the use of Maze Suite to add validity to spatial cognitive research in both behavioral and neurophysiological experimentation.

### AUTHOR NOTE

Correspondence concerning this article should be addressed to H. Ayaz, School of Biomedical Engineering, Science, and Health Systems, Drexel University, 3141 Chestnut Street, Philadelphia, PA 19104 (e-mail: [hasan.ayaz@gmail.com](mailto:hasan.ayaz@gmail.com)).

### REFERENCES

- ASTUR, R. S., TROPP, J., SAVA, S., CONSTABLE, R. T., & MARKUS, E. J. (2004). Sex differences and correlations in a virtual Morris water task, a virtual radial arm maze, and mental rotation. *Behavioural Brain Research*, *151*, 103-115.
- AYAZ, H., IZZETOGLU, M., PLATEK, S. M., BUNCE, S., IZZETOGLU, K., POURREZAEI, K., & ONARAL, B. (2006, August). *Registering fNIR data to brain surface image using MRI templates*. Paper presented at the 28th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, New York.
- CAPLAN, P. J., MACPHERSON, G. M., & TOBIN, P. (1985). Do sex-related differences in spatial abilities exist? A multilevel critique with new data. *American Psychologist*, *40*, 786-799.
- CHAMBERS, C. D., & BROWN, M. (2003). Timing accuracy under Microsoft Windows revealed through external chronometry. *Behavior Research Methods, Instruments, & Computers*, *35*, 96-108.
- CHUTE, D. L., & WESTALL, R. F. (1996). Fifth-generation research tools: Collaborative development with PowerLaboratory. *Behavior Research Methods, Instruments, & Computers*, *28*, 311-314.
- IZZETOGLU, M., IZZETOGLU, K., BUNCE, S., AYAZ, H., DEVARAJ, A., ONARAL, B., & POURREZAEI, K. (2005). Functional near-infrared neuroimaging. *IEEE Transactions on Neural Systems & Rehabilitation Engineering*, *13*, 153-159.
- MICROSOFT (2006). *QueryPerformanceCounter Function* (Microsoft Developer Network Library). Retrieved March 30, 2006, from [msdn2.microsoft.com/en-us/library/ms644904.aspx](http://msdn2.microsoft.com/en-us/library/ms644904.aspx).
- VANDENBERG, S. G., & KUSE, A. R. (1978). Mental rotations, a group test of three-dimensional spatial visualization. *Perceptual & Motor Skills*, *47*, 599-604.
- VOYER, D., VOYER, S., & BRYDEN, M. P. (1995). Magnitude of sex differences in spatial abilities: A meta-analysis and consideration of critical variables. *Psychological Bulletin*, *117*, 250-270.
- WESTALL, R. F., PERKEY, M. N., & CHUTE, D. L. (1986). Accurate millisecond timing on Apple's Macintosh using Drexel's MilliTimer. *Behavior Research Methods, Instruments, & Computers*, *18*, 307-311.

(Manuscript received May 2, 2006;  
revision accepted for publication May 25, 2007.)