

## Establishment of the first Java 3D™ enabled CAVE<sup>01</sup>

---

### A Whitepaper

Christoph W. Sensen, University of Calgary, Sun Center of Excellence for Visual Genomics

Jeff Brum, Fakespace Systems, Kitchener, Ontario

Paul Gordon, University of Calgary, Sun Center of Excellence for Visual Genomics

Mark Hood, Sun Microsystems, Palo Alto, CA

Gary Lindahl, Fakespace Systems, Kitchener, Ontario

Michael Schulman, Sun Microsystems, Palo Alto, CA

Chris Spindler, Sun Microsystems of Canada, Calgary, Alberta

Megan Stuart, Sun Microsystems of Canada, Calgary, Alberta

Stefan Unger, Sun Microsystems, Palo Alto, CA



---

<sup>1</sup> ® CAVE is a registered trademark of the University of Illinois Board of Trustees

## Introduction

The name "CAVE" is a recursive acronym (CAVE Automatic Virtual Environment). The CAVE was researched and developed by the Electronic Visualization Laboratory, at the University of Illinois at Chicago<sup>2</sup> as a tool for scientific visualization. Since its premiere at the 1992 SIGGRAPH conference, the CAVE has achieved international recognition as the pinnacle of immersive virtual reality system technology, providing a compelling display environment for science, engineering, and art. The intent of the CAVE is to facilitate "suspension of disbelief" by making the virtual environment as realistic as possible. The quality of the graphical images is of paramount importance to realism. Depending upon the model or environment being explored, the speed of the computer's graphic system will affect the speed of image update. The layout of the physical CAVE structure itself (typically at least three walls with a 270 degree angle and a floor display) will also affect the sense of virtual presence.

Today, the CAVE technology has been commercialized and is marketed by Fakespace Systems Inc. for universities and industry, where it has proved its worth as a research tool and a virtual prototyping device. The experience and detailing provided by Fakespace Systems have minimized the multiple, ninety-degree corner seams of the physical structure. Optimal projection geometry for the best alignment as well as stereo, brightness, and color uniformity have proven to be key to creating a seamless virtual experience.

Over the last ten years, CAVEs and other immersive rooms have been installed in approximately 100 facilities worldwide. Interestingly, even though the initial goal of the CAVE development was to create an immersive environment for scientific applications, many of the CAVE installations have been in the technical sector, including the aerospace industry, car manufacturing and defense applications. There were two main reasons for this, the usually prohibitive cost of the system installation and the fact that the proprietary operating environment (additions to the operating system, libraries and programming APIs) required developers to be located in close proximity to the CAVE while creating applications. Life sciences applications, including analysis tools for large-scale Genomics data could benefit tremendously if access to CAVE environments would improve substantially.

---

<sup>2</sup> C. Cruz-Neira, D. J. Sandin, T. A. DeFanti. Surround-screen Projection-Based Virtual Reality: The Design and Implementation of the CAVE. ACM SIGGRAPH '93 Proceedings. Anaheim, CA, August 1993.

## Defining the Challenge

In our goal to build the first CAVE environment tailored to Bioinformatics research and development, we faced a number of challenges that needed to be addressed through unique technological solutions. Java™ technology has become one of the major, if not the dominant, programming languages in Bioinformatics, thus the programming environment of the CAVE should include a Java 3D™ API. The compatibility with the Java 3D™ API would have the additional benefit of unlinking the development environment from the execution of the fully developed system.

The operating system of the CAVE server should be UNIX® based, because almost all Bioinformatics applications have been developed for this platform. Especially the Sun Solaris™ Operating Environment (Solaris OE) has proven to be a very popular platform for the development and execution of Bioinformatics applications, because of the 64-bit operating system (OS) architecture, binary compatibility throughout the product line, OS reliability and the built-in remote administration tools. The graphics resolution of the CAVE should be as high as possible, with refresh rates of at least 100 Hz to allow for a flicker-free and seamless viewing experience.

The laboratory layout provided for a 6.50 m by 6.50 m (21 ft. by 21 ft.) interior space for the installation of the CAVE display (four projectors, four mirrors, four display surfaces and tracking gear). This laboratory was much smaller than the standard 8 m (26 ft.) to 10 m (33 ft.) spaces, thus the wall and floor display layout had to be custom designed. Another consideration was the fact that the server room and the CAVE facility were separated by two floors and a distance of approximately 70 m (230 ft.). This called for a connection of the CAVE projectors, keyboard and mouse to the server via a solution that would not degrade the graphics quality and performance.

In addition to the graphics capabilities, we were looking for a shared memory server solution, which would allow us to build complex models in main memory, that, in turn, could be efficiently rendered and manipulated in the CAVE. The large scale of the logical data that we needed to host locally also called for the installation of a major data storage facility.

## Server Architecture

The CAVE is powered by a SunFire™ 6800 midframe server using Solaris™ 8 OE as the operating system. The server features twenty 750 MHz Ultrasparc™ III CPUs (8 MB cache each) and 20 GB of main memory. The system layout is fully redundant, using six power and cooling supplies (N + 1 configuration) and two DVD and 4 mm tape units. Two redundant system controllers featuring redundant system clocks eliminate the possibility for any single point of failure.

The redundant boot devices are connected via two differential SCSI host bus adapters. Four dual-channel fiber-channel host bus adapters provide an aggregated throughput of 800 Mb/s to the eight T3 hardware RAID5 disk arrays with seventy 73 GB FC-AL disks (a total of 5 TB hard disk space). In addition, four dual fast and wide ultra differential SCSI host bus adapters connect a L180 tape library equipped with DLT 8000 median and drives for backup purposes as well as a L700 tape library with five DLT 8000 median and drives that is used as a near linear storage unit (hierarchical storage system) featuring twenty TB of “slow” storage.

Network connections to Ca\*net3/4 (the Canadian high speed Internet) are provided in the form of two Gigabit Ethernet network adapters. Two 10/100Mb/s network interface adapters are used to provide administrative connectivity and support for the network of Sun Ray1 thin client machines (currently 14) equipped with flat panel monitors in the laboratory part of the facility. The server also features a USB port card (Ads Tech Turbo Quad PCI I/O assembly) to connect a USB keyboard and mouse. While it is not mandatory, it is useful for the operation of Java 3D™ applications in the CAVE.

Four Sun™ Expert3D graphics cards are at the heart of the CAVE. Each card is installed in one of the four I/O assemblies of the SunFire™ 6800. The symmetrical layout of the graphics cards guarantees the even delivery of the video signals to the CAVE room. The Expert3D cards are configured to operate at a resolution of 1280 x 1024 pixels for stereo-3D mode and a refresh frequency of 112 Hz.

## CAVE Architecture

The CAVE at the Sun Center for Visual Genomics is an example of a customized CAVE, designed by Fakespace Systems to meet customer specifications. This CAVE is the first installation of a Java 3D™ API-based immersive display, requiring the combined expertise of Fakespace Systems and Sun Microsystems to bring together these complex technologies as a completely unified system. The University's facility was not large enough to accommodate a standard CAVE structure. The optical and geometric capabilities of CRT projection, as well as Fakespace designed structural modifications, allowed the custom 2.40 m by 2.40 m (8 ft. by 8 ft.) CAVE system to fit into the available space. The CAVE features four Marquee 8500 CRT projectors operating at full 1280x1024 resolution as directed by the Expert3D cards. Figure 1 shows the CAVE layout.

The stereoscopic projection effect that makes CAVE experiences compelling requires users to wear a pair of electronic LCD shutter glasses (stereo-glasses). The rapid shuttering actions of the lenses operate optimally at frequencies between 96 and 120 Hz (the limit of most projection technology). In this CAVE, the 112 Hz generated by the Expert3D cards is divided into field sequential imagery— the alternating left and right eye perspective views, displaying 56 frames per second per eye. The user's brain is able to 'fuse' the rapidly alternating left and right eye pictures into one image, which is how we naturally see the physical world.

An electro-magnetic motion-tracking system (Ascension Flock of Birds) is installed in the ceiling of the CAVE to provide positional data, allowing 3D images to move in real-time and be manipulated naturally. To enable this real-time interaction, a motion-tracking sensor is mounted on the glasses of the main user, which provides the users head and eye position to the CAVE application. This information can then be used to compute the correct viewer-centric perspective. For example, looking to one side of a virtual 3D box is as natural as one would look at the same object in the physical world. In the CAVE, the virtual image changes to display the side of the box as the user's head moves accordingly. The main user also operates an interface device, a six degrees of freedom, motion tracked 3D mouse called a "WorkWand", consisting of a joystick for navigating and three programmable buttons for interactivity. In addition to the main user, up to four spectators (also equipped with stereo-glasses) can be comfortably hosted in the CAVE at the same time.

## **Server-CAVE Connection**

The compute server installation requires it to be housed in a controlled and secure environment to ensure adequate provision of power and cooling for its operation. By contrast, the CAVE needs to be conveniently accessible. Thus at the University of Calgary, there is a necessary separation between the server room and the projectors which display the images on to the CAVE walls.

At the Sun Center of Excellence for Visual Genomics, the two rooms are over 70 m (230 ft.) and two stories apart, far beyond the normal reach of a video cable. To resolve the problem of separation, the video signals generated by the Expert3D graphics cards are fed into Video Display Extension (VDE) Systems which act as transmitters, converting the signal from analog copper to digital optical media. This technology extends the monitor display and peripherals, including serial and USB connections (keyboard and mouse) from a server across long distances (up to 3048 m (10,000 ft.)) without any loss of video resolution.

The signal is run through fiber optic cables between the server room and the CAVE, where it is fed into another set of VDE receivers and converted back into an analog video signal on copper media.

## Extensions to the Solaris 8™ Operating Environment

Java 3D™ version 1.3<sup>3</sup> release is at the heart of the University of Calgary CAVE installation. The Java 3D API requires a base implementation of the Java 2 platform, version 1.2.2 or higher. Sun recommends the Java 2 Run Time Environment version 1.3.1 or later for maximum performance (The CAVE is running version 1.4.). In addition, the Forte Developer 6 update provides C, C++ and FORTRAN 77/90 compilers and shared libraries, which are used by several bioinformatics applications. The CAVE is also running the Java 3D Fly Through (J3dFly) application, a Java 3D application used to show developers how to use Java 3D and take advantage of some of the capabilities within Java 3D API. This application is being used as a demonstration platform, as well as providing code examples for developers to incorporate into the bioinformatics applications under development.

Sun Management Center 2.0 is installed as a remote hardware management and monitoring tool. Sun Trunking aggregates network interfaces for increased data throughput as well as redundancy. HPC cluster tools allow several Canadian Bioinformatics Resource (CBR) nodes [for details see <http://www.cbr.nrc.ca>] to work on a single, compute-intensive problem simultaneously. Sun Grid Engine also integrates the distributed CBR computing environment and provides the ability to submit batch jobs. The Sun Ray™ 1 thin clients are operated via the Sun Ray server software. The PAM Smart Card module allows login via smart card (Java Card™) authentication.

Solstice DiskSuite™ software provides the software RAID1 drivers for the boot devices. The hierarchical storage system (L700 tape library) is operated using SAM-QFS, presenting the 5 TB of disk (fast storage) and 20 TB of tape (slow storage) as a single file system to the user. VERITAS NetBackup is used as the backup software that operates the 5 TB L180 tape library.

Trackd is a standard device driver program developed by VRCO to supply data from motion trackers and controller input devices for immersive displays supplied by Fakespace Systems. Trackd is a small "daemon" application on the server, which takes information from a variety of peripherals and makes this information available for immersive 3D applications to use. It is the middle-ware standard for the Immersive Display industry. The Java 3D™ API immersive application used at the University of Calgary employs Trackd to get the real-time peripheral data from the tracking system and wand device. This was accomplished using the Java Native Interface code to the Trackd API, and the creation of a new View Behaviour that may be used with any Trackd compatible input device.

---

<sup>3</sup> Version 1.3 of Java 3D, which the CAVE is using, requires version 1.3.1 or higher of the Java 2 platform. The ConfiguredUniverse utility requires version 1.4 of the Java 2 platform for multi-screen displays such as the CAVE.

## Java 3D™ API

Developers can easily incorporate high-quality, scalable, platform-independent 3D graphics into Java™ technology-based applications and applets. The Java 3D™ application programming interface (API) provides a set of object-oriented interfaces that support a simple, high-level programming model. This enables developers to build, render, and control the behavior of 3D objects and visual environments. By leveraging the inherent strengths of the Java programming language, Java 3D technology extends the concept of "Write Once, Run Anywhere™" to 3D graphics applications.

The Java 3D API takes advantage of existing hardware accelerators through low-level APIs such as OpenGL® and Direct3D. This allows applications written using the Java 3D API to run on any platform with a Java virtual machine (JDK™ version 1.3.1 or higher software) and OpenGL or Direct3D implementation and Java 3D. To accommodate a wide variety of file formats, runtime loaders are supported. The Java 3D API benefits developers in diverse markets including scientific visualization, animation, Web site design, simulations, virtual world construction, training, games, and design automation. As an integral part of the Java™ Media API, the Java 3D API makes it easier for programmers to integrate 2D and 3D graphics, video, audio, and image processing, as well as multimedia and visualization features, in a single application.

The Java 3D API incorporates a high-level, scene-graph model that helps developers focus on objects and scene composition. This speeds application development, because programmers do not need to design specific geometric shapes or write rendering code for the scene display.

## **Configured Universe**

ConfiguredUniverse is a new utility in the Java 3D version 1.3 release which allows an application to set up an interactive viewing environment based upon the contents of a site-specific configuration file. Using this utility, an application can be configured to run across a broad range of viewing configurations and available input devices, without any changes to the source code, simply by editing a configuration file to describe the local viewing environment.

Viewing configurations that can be described in the file include windows on conventional desktops, stereo-enabled views, fullscreen-immersive displays on single or multiple screens, and virtual reality installations such as CAVE and head-mounted displays incorporating head tracking 6DOF sensor devices. Input devices and navigation behaviors can also be specified within the configuration file to allow further tailoring of the application to available resources and user preferences. Table 1 shows the configured universe configuration files used for the Calgary CAVE.

## Conclusion

Using existing technologies and combining them in an innovative new way, the team has succeeded in building the world's first Java 3D™ API enabled CAVE. The immersive virtual reality environment performs seamlessly and the effects on first-time viewers can only be described as stunning. Reuters correspondent Jeffrey Jones said in his report: "The result is so realistic one cannot resist reaching out to try to touch the image."

We envision numerous new applications for the new platform. Starting with images derived from advanced imaging technologies like confocal light microscopy, electron microscopy or functional MRI, we will be able to build object-oriented models of cells, tissues, organs and entire organisms.

These models will over time be populated with complex Genomics data sets, including DNA chip results (expression studies) and Proteomics data. Being able to observe the expression of all genes in an organism in their compartments and using temporal data to model time sequences, will allow us to obtain new insights into the complex organization and functionality of life.

The major benefit of the new architecture is the use of Java 3D™ API and Java™ technology has become one of the major programming environments for Bioinformatics application development. Being able to completely disconnect the program development from the program execution in the CAVE will allow our collaborators from around the world to benefit from the new system. Because the Java applications are translated automatically into the new viewing mode, the programmers will be able to almost completely ignore the fact that they are creating their new application for an immersive environment, rather the two-dimensional screen that they work on during the programming phase.

**Table 1: Java 3D™ configuration file for the Calgary CAVE.**

```

/*
*****
*
* Java 3D configuration file for the Calgary CAVE.
*
* The display consists of four 8' x 8' screens to the left, front, and
* right of the user, along with a floor, all at 90 degrees to each other.
* The origin of each vertical screen is the lower left corner as the user
* faces it within the cave. The origin of the floor screen is the left
* corner closest to the entrance.
*
* Two motion-tracking 6DOF sensors are available in the CAVE. One is
* attached to the glasses of the main user to track the head and eye
* positions. The other is attached to a 3D mouse used as an interface
* device. This device also provides a third joystick-like 2D sensor and
* three buttons. These devices are all accessed through a VRCO Trackd
* implementation of the Java 3D InputDevice interface.
*****
*/

// Create new screen objects and associate them with logical names and
// numbers. These numbers are used as indices to retrieve the AWT
// GraphicsDevice from the array that GraphicsEnvironment.getScreenDevices()
// returns.
//
(NewScreen left 0)
(NewScreen center 1)
(NewScreen right 2)
(NewScreen floor 3)

// Set the available image areas as well as their positions and orientations
// relative to the tracker base.
//
// Dimensions in the Java 3D configuration file are in meters. The scale
// used here is 2.54 cm = 1 inch.
//
(ScreenAttribute left PhysicalScreenWidth 2.438)
(ScreenAttribute left PhysicalScreenHeight 2.438)
(ScreenAttribute left WindowSize NoBorderFullScreen)
(ScreenAttribute left TrackerBaseToImagePlate
(( 0.0 0.0 -1.0 1.219)
 ( 0.0 1.0 0.0 0.000)
 ( 1.0 0.0 0.0 1.219)))

(ScreenAttribute center PhysicalScreenWidth 2.438)
(ScreenAttribute center PhysicalScreenHeight 2.438)
(ScreenAttribute center WindowSize NoBorderFullScreen)
(ScreenAttribute center TrackerBaseToImagePlate
(( 1.0 0.0 0.0 1.219)
 ( 0.0 1.0 0.0 0.000)
 ( 0.0 0.0 1.0 1.219)))

(ScreenAttribute right PhysicalScreenWidth 2.438)
(ScreenAttribute right PhysicalScreenHeight 2.438)
(ScreenAttribute right WindowSize NoBorderFullScreen)
(ScreenAttribute right TrackerBaseToImagePlate
(( 0.0 0.0 1.0 1.219)
 ( 0.0 1.0 0.0 0.000)
 (-1.0 0.0 0.0 1.219)))

(ScreenAttribute floor PhysicalScreenWidth 2.438)
(ScreenAttribute floor PhysicalScreenHeight 2.438)
(ScreenAttribute floor WindowSize NoBorderFullScreen)
(ScreenAttribute floor TrackerBaseToImagePlate
(( 1.0 0.0 0.0 1.219)
 ( 0.0 0.0 -1.0 1.219)
 ( 0.0 1.0 0.0 0.000)))

```

```

// Create a new physical environment and bind a name to it.
//
(NewPhysicalEnvironment CAVE)

// Set the location and orientation of coexistence coordinates relative to the
// tracker base. The view platform origin in coexistence coordinates will be
// mapped to this location and orientation.
//
// Coexistence coordinates are set to and aligned with the middle of the
// front screen here.
//
(PhysicalEnvironmentAttribute CAVE
    CoexistenceToTrackerBase
    (( 1.0 0.0 0.0 0.000)
     ( 0.0 1.0 0.0 1.219)
     ( 0.0 0.0 1.0 -1.219)))

// Instantiate the VRCO Trackd InputDevice implementation.
//
(NewDevice trackd VRCOTrackdInputDevice)

// Create logical names for the available sensors in the specified input
// device. The last argument is the sensor's index in the input device.
// The application can retrieve these sensors via their logical names by
// calling the getNamedSensors() method of ConfiguredUniverse.
//
(NewSensor head      trackd 0)
(NewSensor wand      trackd 1)
(NewSensor controller trackd 2)

// Register the input device in the PhysicalEnvironment.
//
(PhysicalEnvironmentAttribute CAVE InputDevice trackd)

// Register the sensor which will drive head tracking.
//
(PhysicalEnvironmentAttribute CAVE HeadTracker head)

// Define the physical body.
//
// The head origin is halfway between the eyes, with X extending to the right,
// Y up, and positive Z extending into the skull.
//
(NewPhysicalBody      CAVEUser)

// Set the interpupillary distance. This sets the LeftEyePosition and
// RightEyePosition to offsets of half this distance along both directions of
// the X axis.
//
(PhysicalBodyAttribute CAVEUser StereoEyeSeparation 0.066)

// Define the position and orientation of the head relative to the sensor
// mounted on the stereo shutter glasses.
//
(PhysicalBodyAttribute CAVEUser HeadToHeadTracker (( 1.0 0.0 0.0 0.080)
                                                    ( 0.0 1.0 0.0 0.000)
                                                    ( 0.0 0.0 1.0 0.000)))

// Create a view using the defined screens, PhysicalEnvironment, and
// PhysicalBody.
//
(NewView      view0)
(ViewAttribute view0 Screen      left)
(ViewAttribute view0 Screen      center)
(ViewAttribute view0 Screen      right)
(ViewAttribute view0 Screen      floor)
(ViewAttribute view0 PhysicalEnvironment CAVE)
(ViewAttribute view0 PhysicalBody      CAVEUser)

// Enable stereo viewing and head tracking.
//

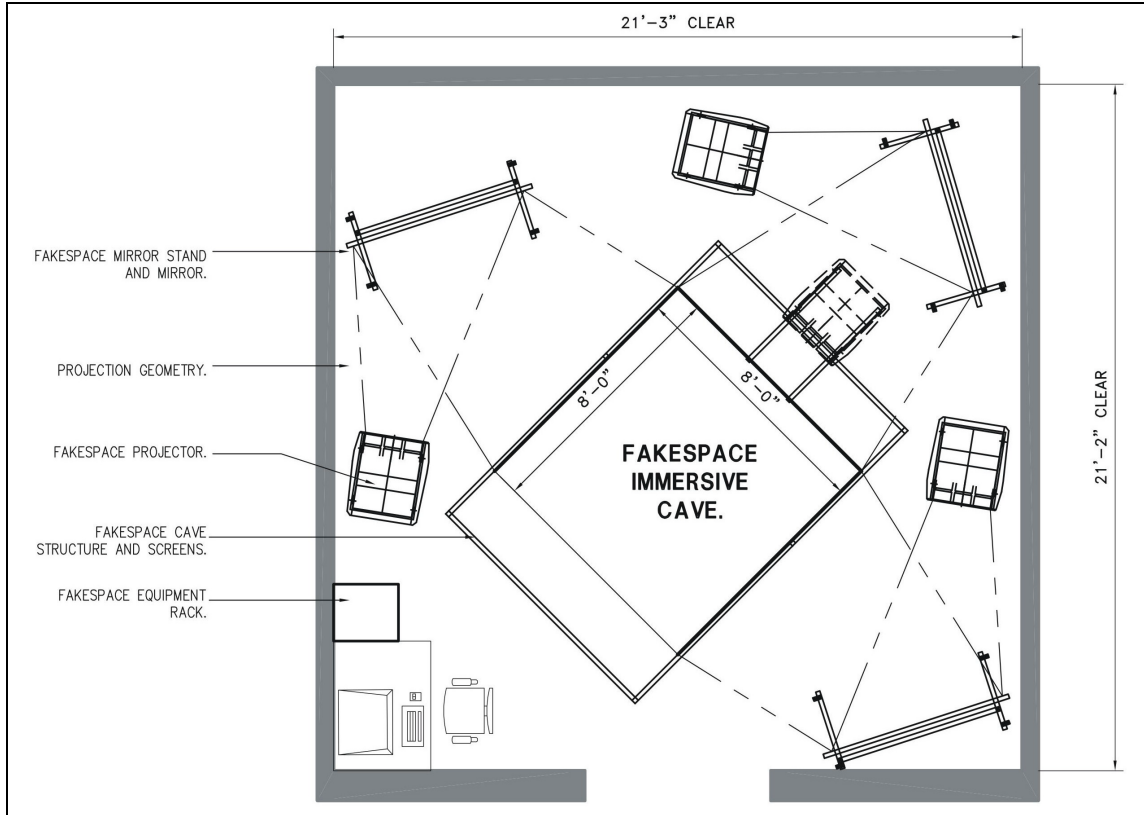
```

```
(ViewAttribute view0 StereoEnable true)
(ViewAttribute view0 TrackingEnable true)

// Set the clip planes from the default 0.1 near and 10.0 far eye distances.
(ViewAttribute view0 FrontClipDistance 0.3)
(ViewAttribute view0 BackClipDistance 500.0)

// Set the nominal user eye position in the display environment, relative to
// coexistence coordinates. It is set here to the middle of the cave,
// 1.737 meters from the floor. This is ignored if head tracking is enabled.
//
(ViewAttribute view0 CenterEyeInCoexistence (0.0 0.518 1.219))
```

**Figure 1: CAVE layout at the University of Calgary**



## **Copyright and Disclaimers**

© 2002 Sun Microsystems, Inc. and Fakespace Systems Inc. All rights reserved.

Sun, Sun Microsystems, the Sun Logo, Java 3D, Java, Solaris, SunFire, Java Card, Solstice DiscSuite, Sun Ray, Write Once, Run Anywhere and JDK are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. OpenGL is a registered trademark of Silicon Graphics, Inc.

UNIX is a registered trademark in the United States and other countries, exclusively licensed through X/Open Ltd Company, Ltd.

CAVE is a registered trademark of the University of Illinois Board of Trustees.