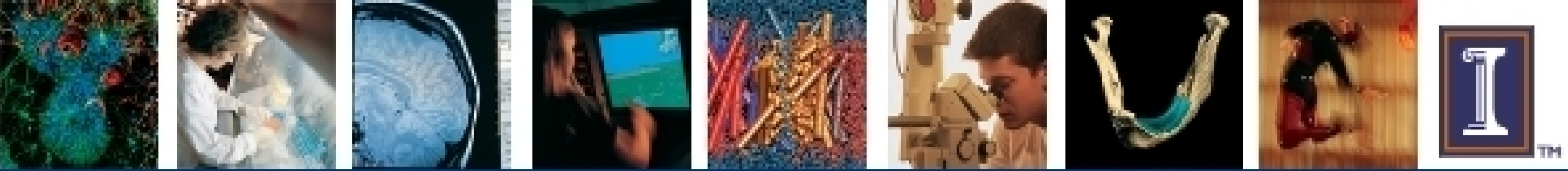


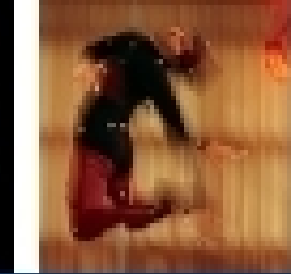
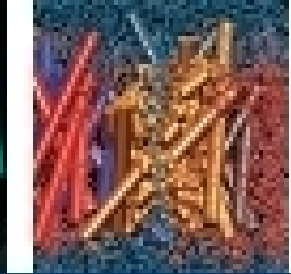
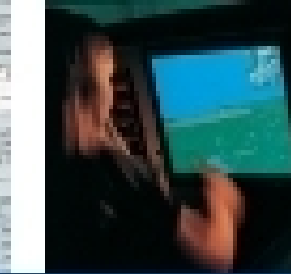
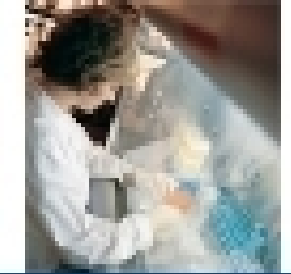
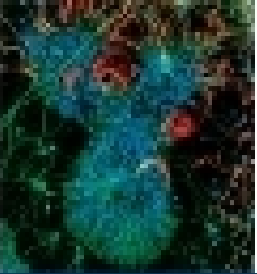
Syzygy: PC-Cluster-Based Virtual Reality



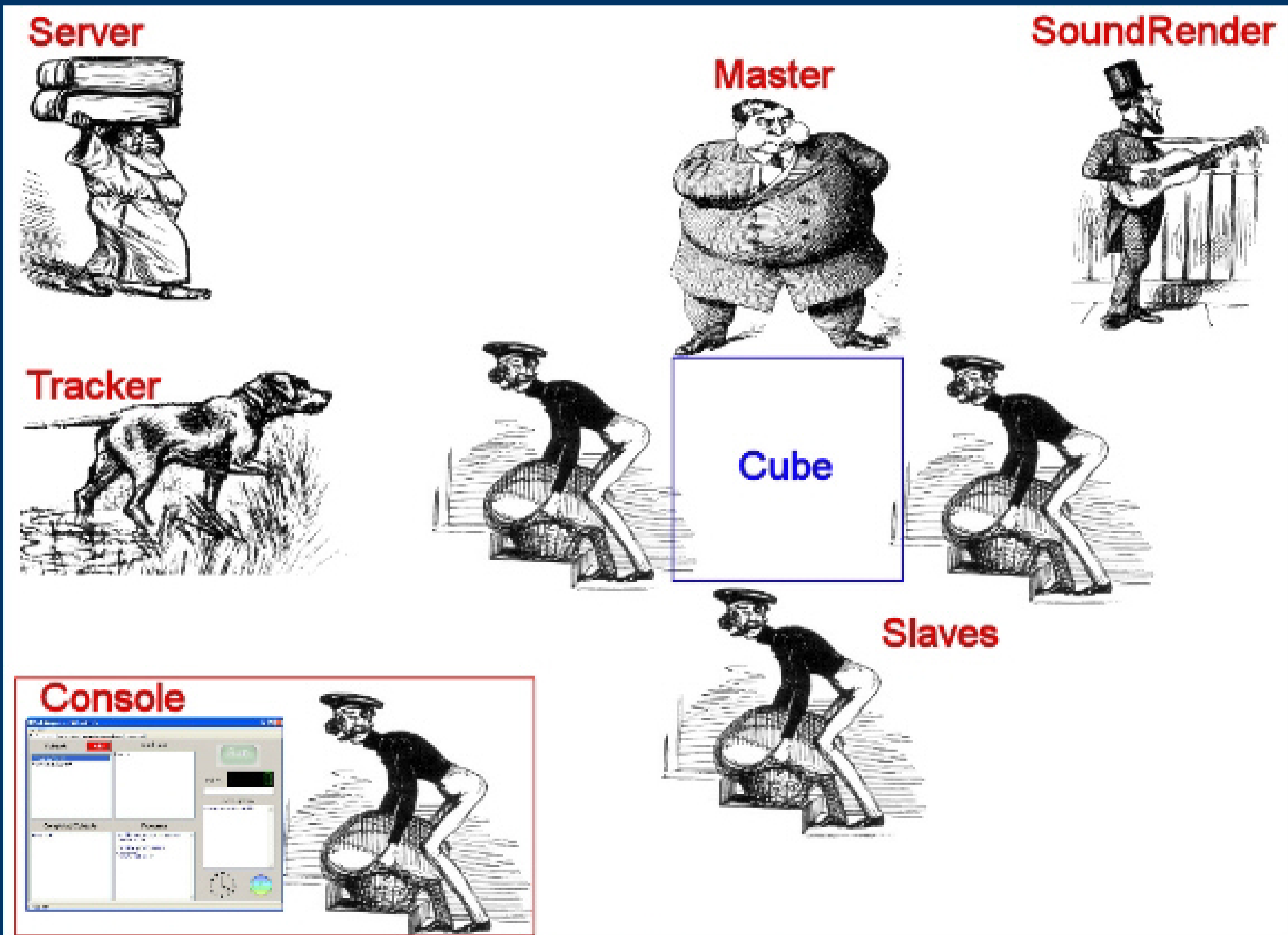


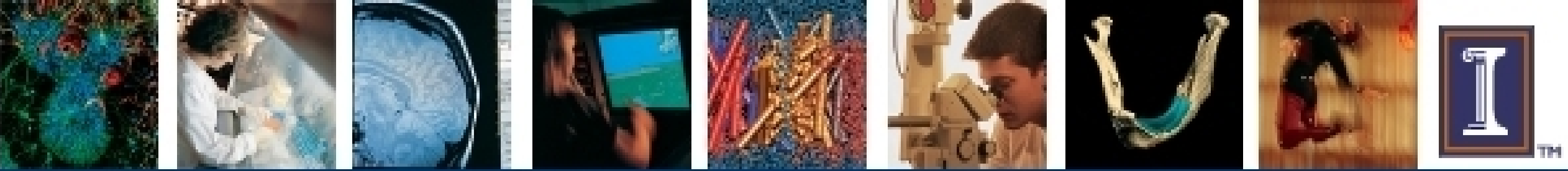
Syzygy Features

- Runs on Windows, Linux, MacOS X, Irix.
- Libraries written in C++, uses OpenGL for drawing.
- Programmable in C++ or Python.
- Supports lots of 6-DOF tracking devices.
- Supports many methods of stereoscopic rendering: active (LCD shutter), passive polarized, anaglyph (red/cyan), side-by-side, over/under.
- Scales nicely: run programs on anything from an SGI to a laptop.
- Programs also run in Standalone Mode for easier development.



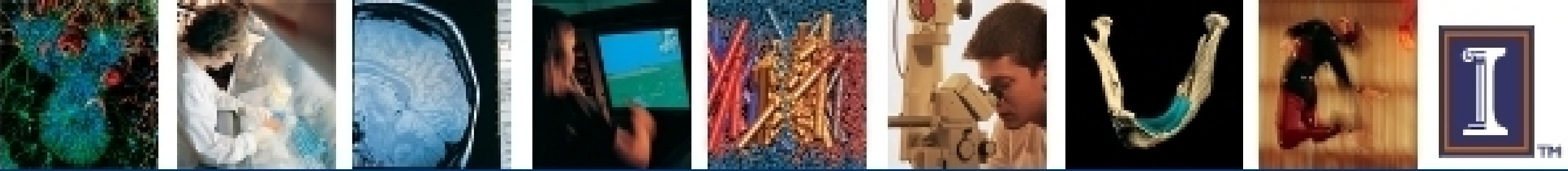
Cluster Components





Virtual Computers

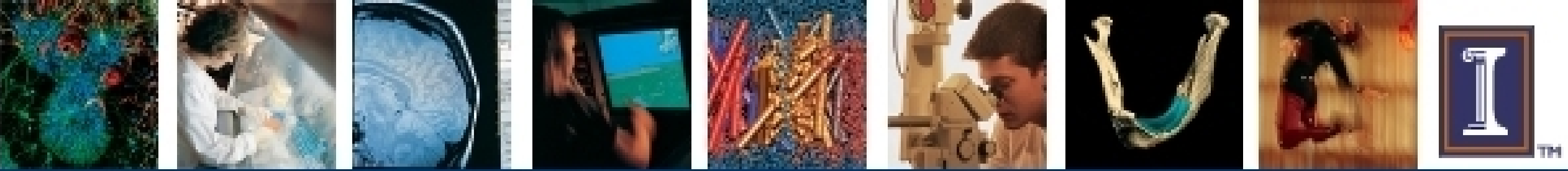
- Specifies which software components run on each computer and how they communicate.
- Multiple distinct virtual computers can be defined on the same cluster.
- Launching an application on a virtual computer automatically causes all incompatible components to be terminated.
- The Syzygy server brokers connections between components.



Standalone Mode

- Syzygy server is replaced by an XML parameter file.
- Application framework can load + play sounds.
- Application framework loads input device drivers or displays an *input simulator*.

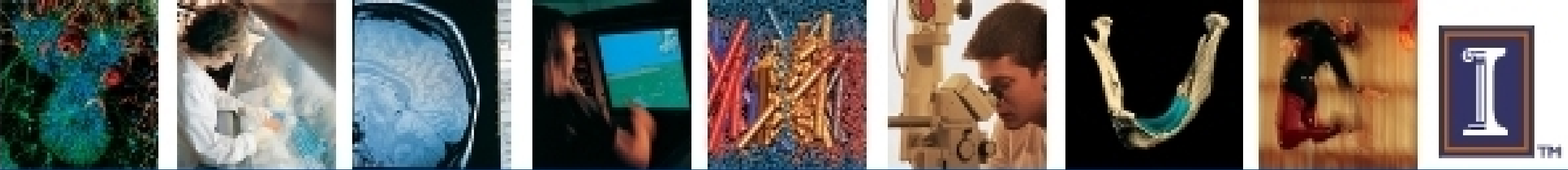
Great for code development, but for final testing set up a single-PC cluster instead.



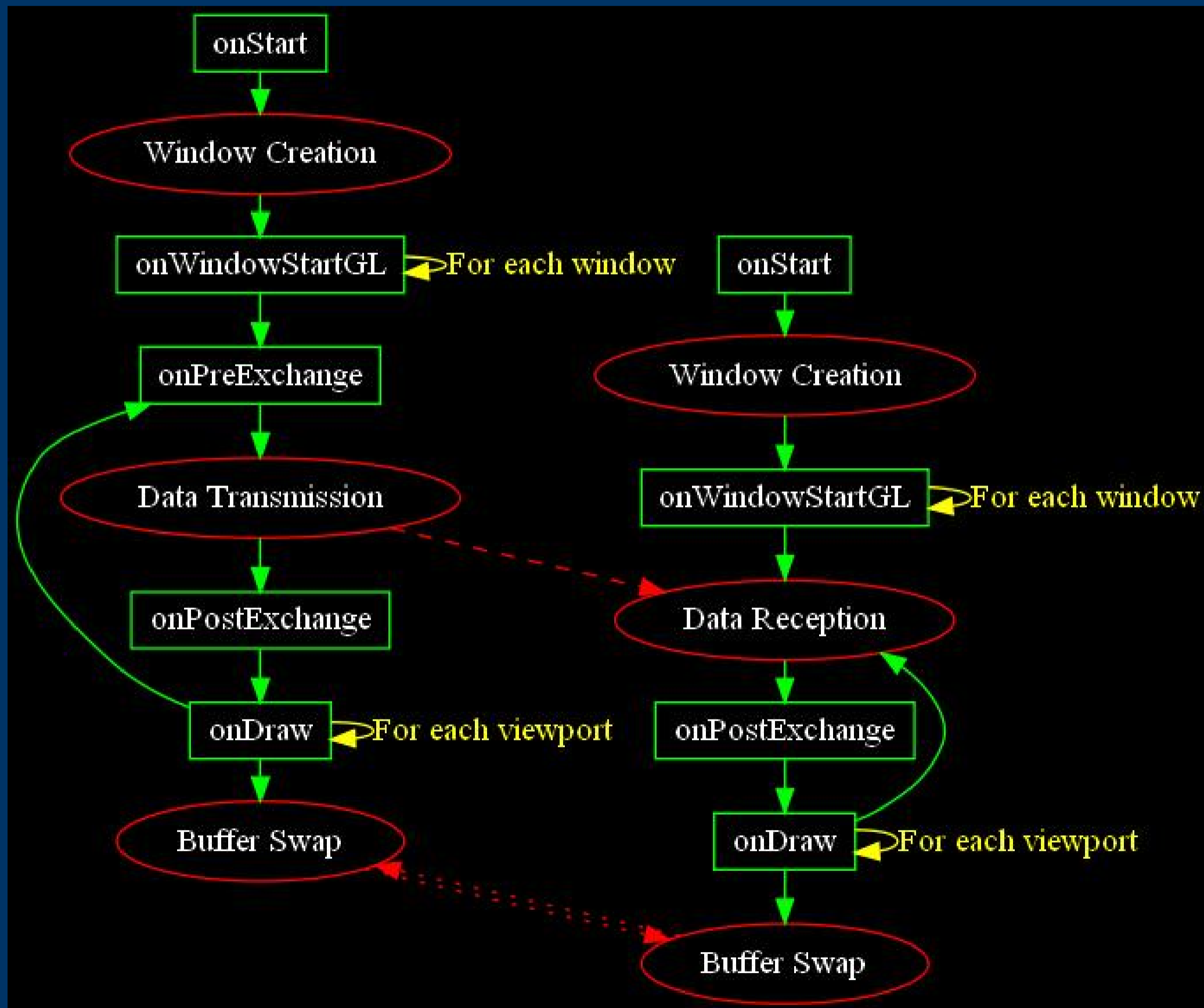
Application Framework

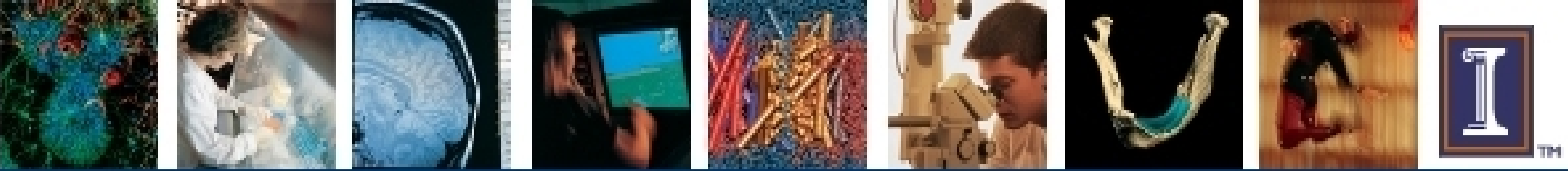
`arMasterSlaveFramework` in C++,
`arPyMasterSlaveFramework` in Python.

- Gets config info from server, sets up appropriate windows, viewports, and viewing frustum parameters.
- Handles event loop, calling user-defined callback methods at fixed points in the loop.
- Provides access to user-generated input events.
- Transfers user-specified data from master to slaves.



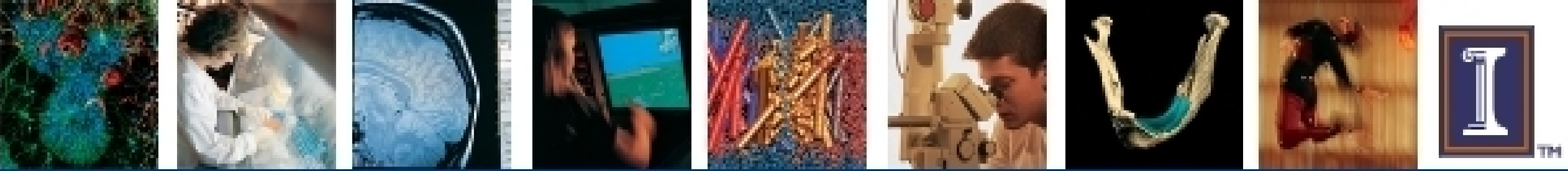
Application Event Loop (Master and Slave)





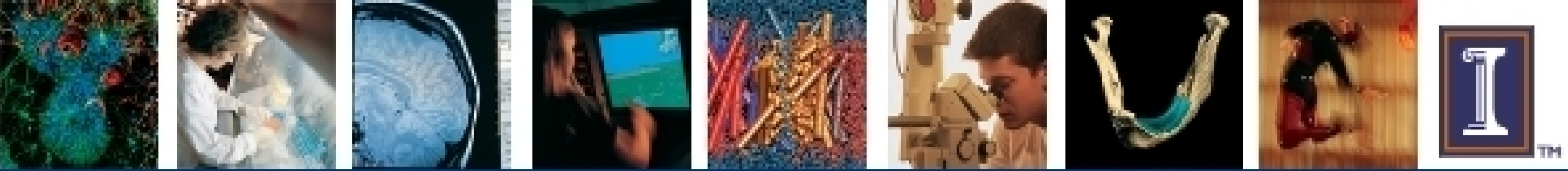
Other Useful Callback Methods

- `onUserMessage()`: Allows you to send text messages to your app in Cluster Mode using the `dmsg` command-line utility.
- `onKey()`: Allows your app to respond to keypresses in Standalone Mode. You can also use `dmsg` to get the same effect in Cluster Mode.
- `onDisconnectDraw()`: Called on a slave that does not have a connection to a master. Good for drawing startup splash screens.
- `onCleanup()`: Called when your app exits.



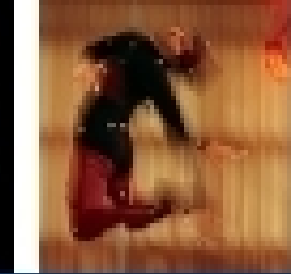
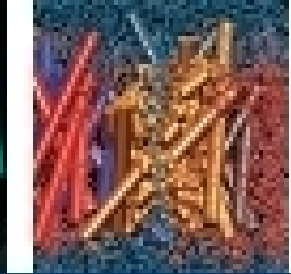
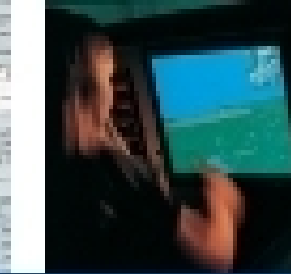
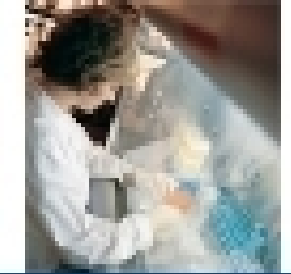
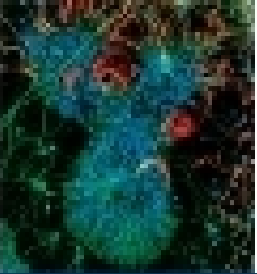
Viewing Configuration

- Viewing parameters are normally not set in user code.
- Determined from parameters set in the Syzygy server (Cluster Mode) or XML file (Standalone).
- Each viewport has an attached **camera**, which relates it to a **screen** (a window into the virtual world).
- Used together with eye position to determine the viewing frustum in each video frame.



User Input 1

- 3 types of input events:
 1. **Button**: 0 or 1.
 2. **Axis**: a single floating-point number, typically a joystick axis.
 3. **Matrix**: 16 floats, representing a 4x4 placement (position + orientation) matrix.
- E.g., a tracked wand might provide 1 matrix, 2 axes (joystick X and Y), and 8 buttons.
- Events of the same type are distinguished by index, e.g.:
 1. Matrix **0** is the head-tracker placement matrix.
 2. Matrix **1** is the wand.
- There's a special purpose filtering language--*PForth*-- for modifying streams of input events.



User Input 2

The application framework provides methods for querying the current state of any input event:

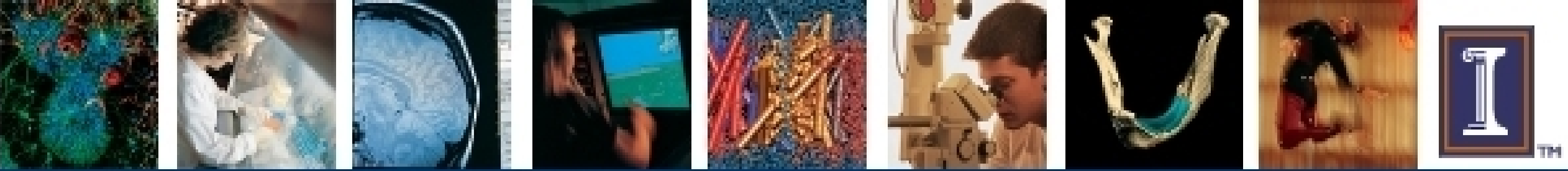
```
getMatrix( int index )  
getAxis( int index )  
getButton( int index )
```

...and for detecting button presses and releases...

```
getOnButton( int index )  
getOffButton( int index )
```

...and for determining position and orientation of the midpoint between the two eyes.

```
getMidEyeMatrix()  
getMidEyePosition()
```



Getting Syzygy

<http://syzygy.isl.uiuc.edu/>

...but you should get it directly from us, as there are some optional dependencies that we're not technically allowed to redistribute...

Windows users are recommended to get *Aszgard*:

<http://syzygy.isl.uiuc.edu/aszgard/>

(but again, class members should get it from us).

- Based on Movable Python (no installation required).
- Contains all tools and external dependencies.
- Wherever you put it, you get a shell with all environment variables and paths configured appropriately on the fly.
- Contains scripts to e.g. easily set up a single-computer Syzygy cluster.