

## Columbia University

Marc Eaddy, Erik Peterson, John Waugh, Hrvoje Benko, Sean White, Steven Feiner

Goblin is a research platform for building augmented reality and virtual reality applications and games. It is written in C# and uses Managed DirectX. Goblin leverages the Common Language Runtime and .NET Framework to provide innovative application features, including Edit-and-Continue and, soon, Aspect-Oriented Programming.

### Features

- Scene graph
  - Animation
  - Collision detection
  - Pathfinding
- Devices
  - Sony LDI-D100B optical see-through head-worn displays (800x600 resolution)
  - InterSense IS900 and IS600 6DOF tracking devices
  - EssentialReality P5 gloves
  - 6DOF device abstraction
- Application plug-ins
- Edit-and-Continue

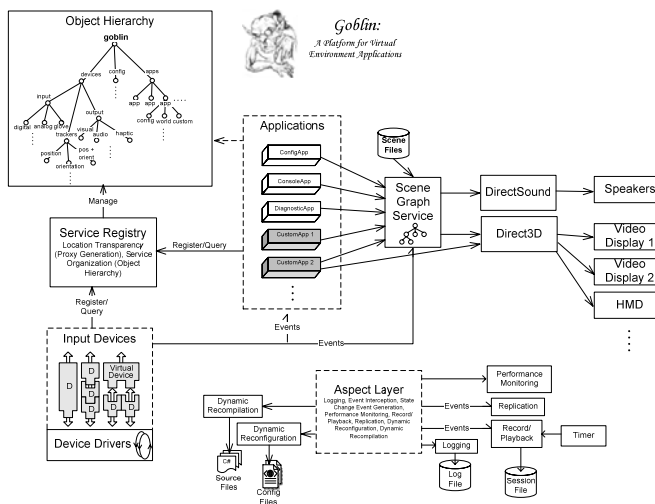


Figure 1. Goblin system architecture.

In addition to enabling the development of 3D applications and games, Goblin serves as a proving ground for research in software architecture, programming languages, virtual machines, and compilers.

Edit-and-Continue.NET is a technology that we developed for Goblin that allows you to **modify the source files** of a running application written in C#, VB.NET, or JScript.NET (or a combination). Changes are automatically **compiled in the background** and the running application is **updated on-the-fly**. The entire update process is **very fast** (< 1 second) and suitable for interactive development and debugging, with very low overhead. This even works for changes made to dynamically loaded plug-ins.

For example, in Goblin we use Edit-and-Continue.NET to tweak calibration and configuration code while Goblin is running. This allows us to quickly prototype small changes without stopping the application.

### Edit-and-Continue.NET

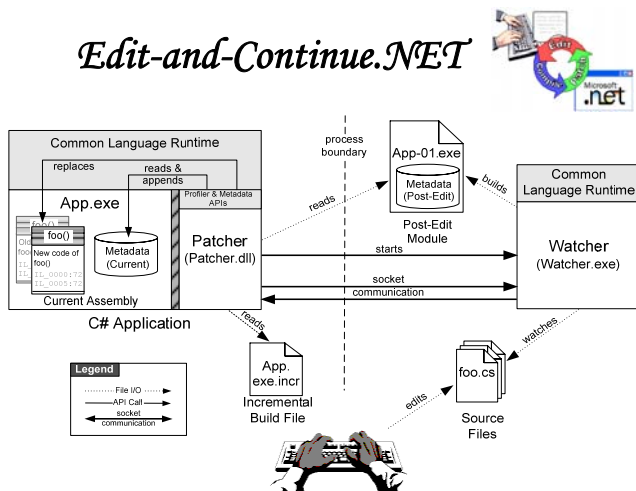


Figure 2. Edit-and-Continue.NET system architecture.

### Future Directions

We are working with Microsoft's **Phoenix** researchers to extend their compiler backend infrastructure to enable non-native C# language constructs, such as **Open Classes** and **Aspect-Oriented Programming**. The goal is to provide techniques for implementing certain features that require time-consuming, laborious, or error-prone development, or adding features that were not originally anticipated. Goblin will serve as a testbed for these techniques. Examples of features we would like to implement using Aspect-Oriented Programming are state change notifications, data flow visualization, plug-ins, persistence, replication, logging, and profiling.

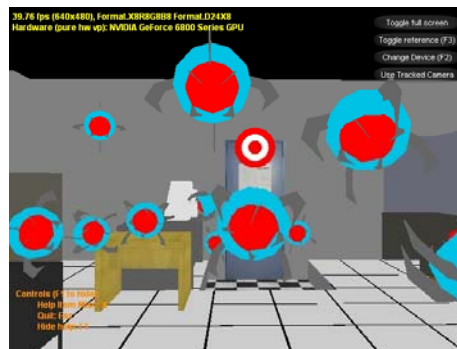


Figure 3. Botica—A prototype 3D game built using Goblin, and playable in augmented reality or virtual reality.

