

Upgrade Your AI

SimBionic® is an AI middleware tool that simplifies the process of creating intelligent behavior for games and simulations. Using SimBionic, you can specify how game characters and simulated entities dynamically react to game events and player actions. SimBionic enables everyone on the development team to write smarter behaviors that are more realistic, challenging, and engaging. By accelerating development and encouraging collaboration, SimBionic increases productivity and frees your team to focus on innovation.



Smarter games, faster

Visual AI Authoring

SimBionic's intuitive visual behavior editor enables designers to create AI behaviors without programming expertise, letting your programmers focus on coding new features.

Modular, Reusable AI

SimBionic AI is naturally modular, making it easy to repurpose and extend behavior created during past projects - meaning more time for you to spend raising the bar on your next project.

Easy Integration

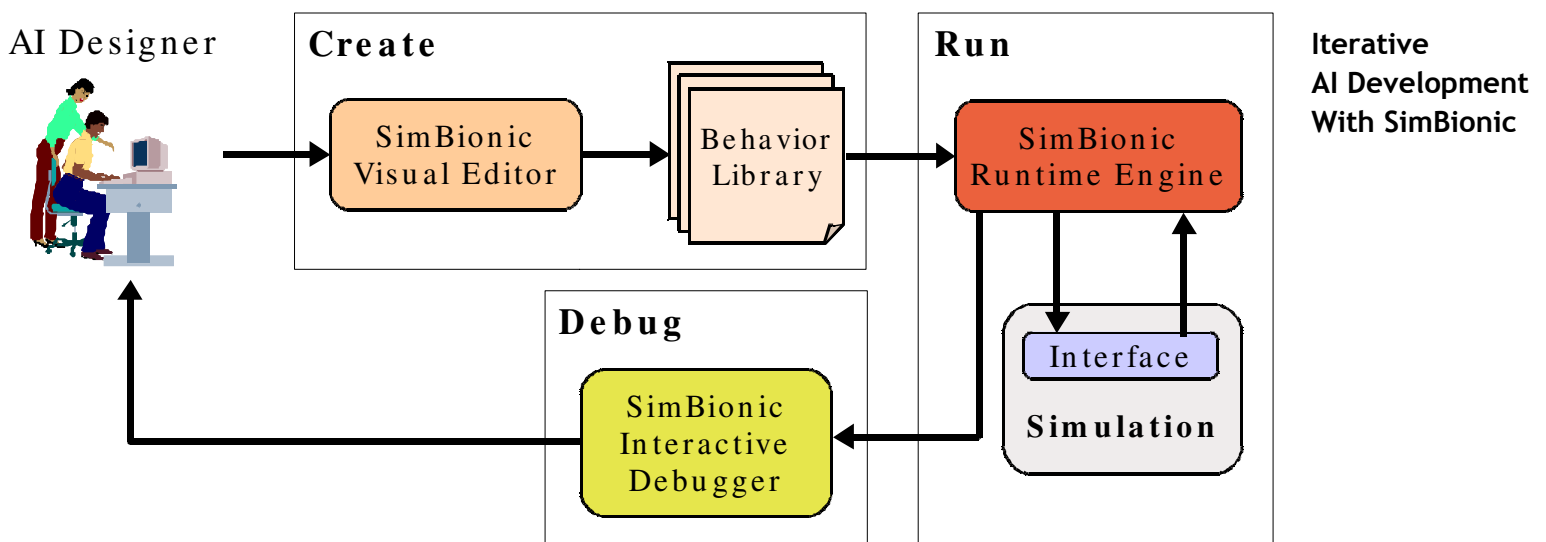
SimBionic's runtime engine integrates easily with nearly any game or simulation, and the SimBionic toolset will drop seamlessly into your development pipeline.

Sophisticated Decision-making & Complex Behavior

SimBionic lets you concentrate on what you want your characters to do rather than how you're going to get them to do it. The result is characters and computer-generated forces with richer, deeper behavior capable of grabbing and holding the player's attention.

Rapid Prototyping & Development

With SimBionic, you can design your AI behaviors in the editor and then immediately test them out in your game without recompiling, for a fast development cycle that maximizes creative iterations. You can even begin sketching out your AI in preproduction before you have a running game engine.



Features

SimBionic is a comprehensive integrated development environment for complex behavior and decision-making logic, featuring a visual behavior editor, runtime engine, and interactive debugger.

Visual Behavior Editor

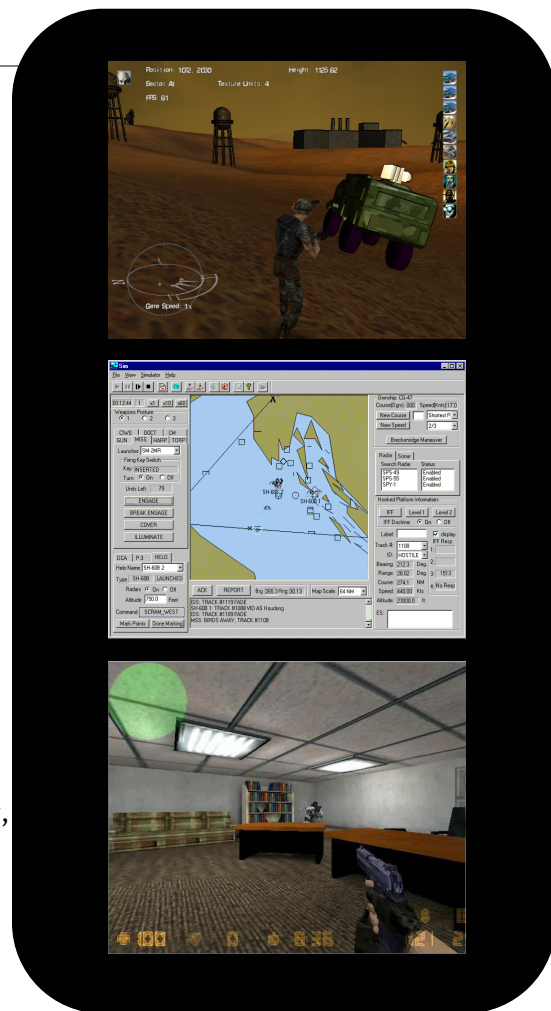
- Intuitive editing approach that lets you draw AI behaviors by dragging and dropping logical building blocks.
- Compile-time error checking.
- Automatically-generated skeleton code for interfacing with your application.
- Automatically-generated HTML documentation.
- Command-line batch mode for integration into your build pipeline.
- Support for multi-user development.
- Export to XML.

Runtime Engine

- C++ and Java versions.
- A thin API for easy integration with virtually any game, application, Java applet, or web server application.
- Impressive speed, backed by a customizable load-balancing scheduling system that minimizes impact on frame rate.
- Efficient use of memory, thanks to both a small footprint and the engine's own customizable memory management system.
- Hierarchical behavior execution, which lets any behavior invoke any other behavior, which in turn can invoke any other behavior, ad infinitum, allowing you to construct sophisticated behaviors from simpler ones and efficiently reuse your work.
- Unique stack-based transition-checking mechanism, which allows higher-priority behaviors to interrupt or override lower-priority behaviors.
- Support for polymorphic behaviors, where the game engine dynamically selects the appropriate behavior to perform based on the entity's current state.
- A library of built-in commands supporting communication between entities via both message queues (that allow entities to join groups and exchange team messages) and virtual blackboards (that allow any entity to post messages that can be read by any other entity).
- Ability to directly invoke methods on Java objects from within behaviors. (Java version only)

Interactive Debugger

- See your behaviors executing via full integration with the visual behavior editor.
- Debug your application remotely.
- Examine the execution stack for any SimBionic-controlled entity.
- Examine the value of any SimBionic variable.
- Step into and through your behaviors.
- Set breakpoints on behavior nodes (including conditional breakpoints).
- Create watchlists to detect when a variable's value changes.



The development cycle

