

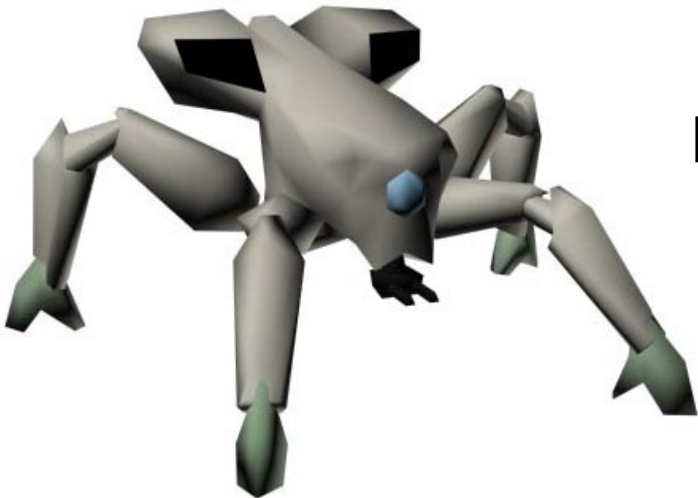
Beyond

A Portable Virtual World Simulation Framework

Jason L. Asbahr

ASBAHR.COM

November 1998



What is Beyond?

Beyond *adj, n*

1: on or to the farther side: FARTHER

2: something that lies outside the scope of ordinary experience

3: a system written in Python for controlling "real time" 3D environments, including video games and educational titles!



What is Beyond?

Object oriented simulation framework

Separates game mechanics from graphics code

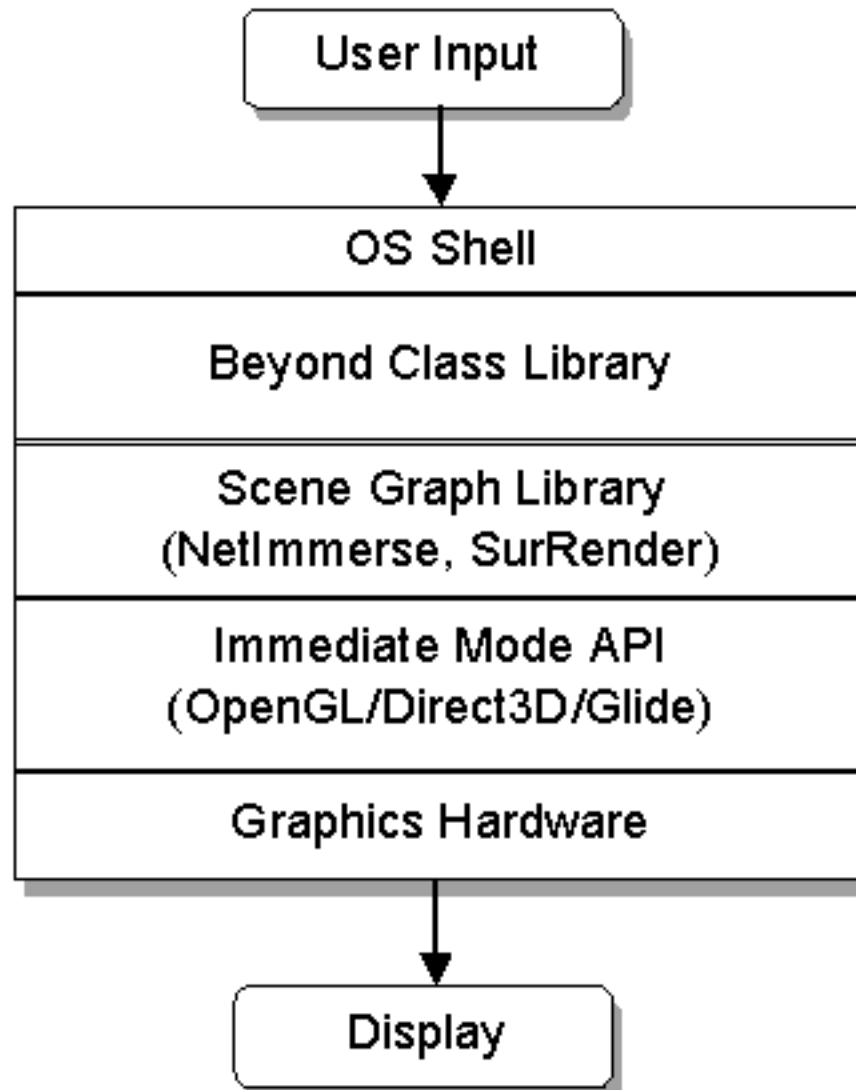
Provides a game-domain logical abstraction

Rooms, Characters, Weapons, Treasure, etc

Composed of elements called "SimObjects"



High Level View



SimObjects

Basic unit of simulation

- instance variables represent state

- methods implement behavior

Optionally contain C++ graphic-oriented objects

- scene subgraphs can be arbitrarily complex

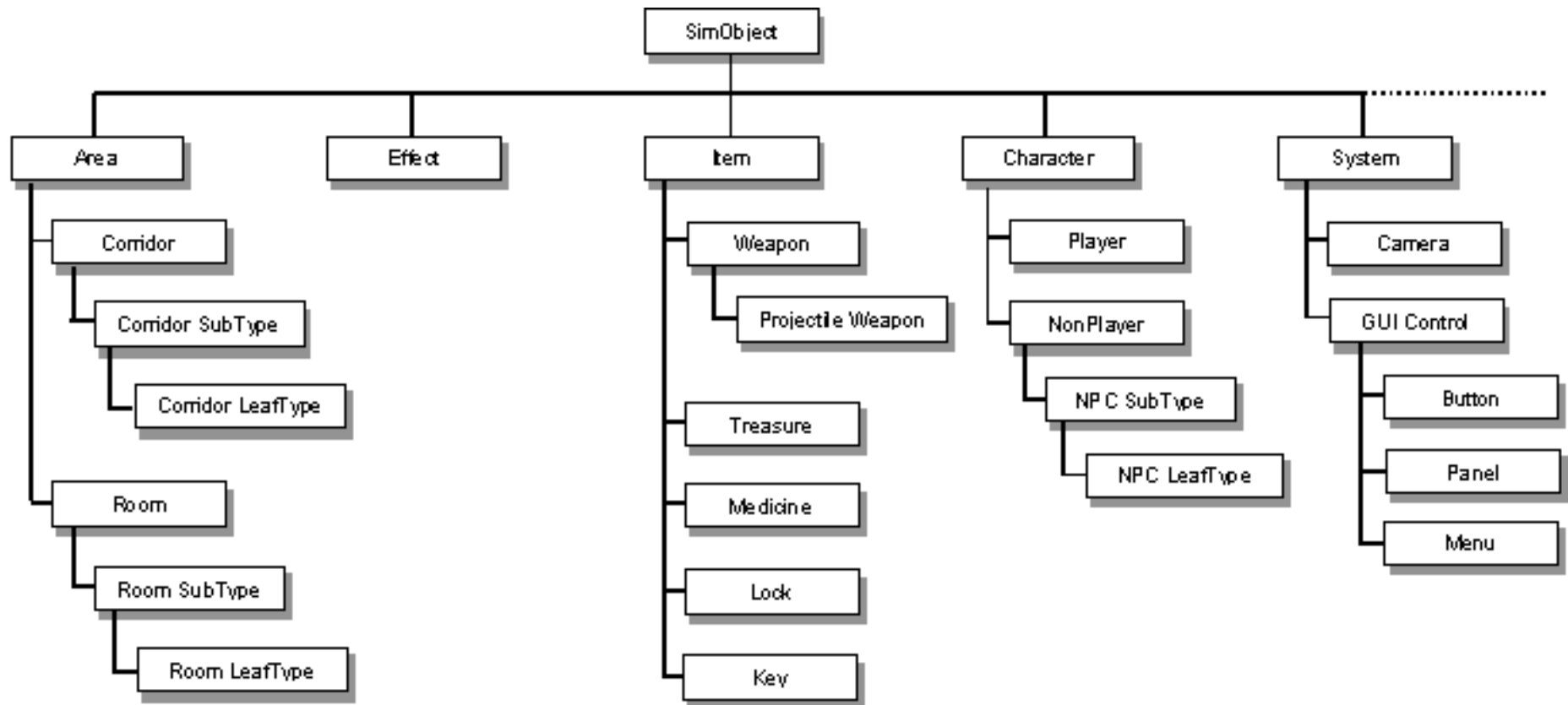
Communicate via discrete events, method calls

Subset of possible messages embodies the

Beyond Protocol



Inheritance Hierarchy



Simulation Structure

SimObjects organized into containment hierarchies

Data flows depth-first through the system

Provides a mechanism for partitioning



Simulation As Scheduling

Actions

- Instantaneous event

- Method call with arguments

Activities

- Time-ordered collection of Actions

- Can be associated with specific graphic animations



Role of Collision Detection

Object communication

Direct messaging

Emergent interaction

Hierarchical collision volumes

Collision Lists

Context-based collision response



Network Console

Mechanism for runtime modification

Allows developer to telnet into the system

User commands are given to the Python interpreter to execute

Useful for loading SimObjects and tweaking object attributes



Subsystem Integration

Graphics, sound, and networking libraries

SWIG is used to wrap C++ classes as shadow objects

Goal: Completely interchangeable subsystems

Consequence: Identification of critical functionality subsets



Performance

Beyond code consumes ~10% of CPU time

Areas for optimization

- interface layer

- moving collision functions to C level

- algorithmic improvements in Python code



What's Next?

Database Integration for Persistence

Better Console, 2D GUI Windows and Controls

Integration of a Free Scene Graph Library

"Headless" Server Mode

Optimization

