

Client-side motion and physics scripting in distributed virtual worlds

Will Monroe

CURIS Summer 2011

Mentors: Philip Levis, Ewen Cheslack-Postava, Behram Mistree

Problem:

Writing scripts to coordinate complex types of motion, such as realistic collision and physics, in a distributed virtual world is tedious and error-prone.

A centralized approach (one script controlling many objects) is simpler, but doesn't leverage the ability to distribute computation among multiple clients.



What is needed:

a library to abstract away the burdensome details of motion control while still allowing customization of each object's motion on the client

Challenges:

Usability

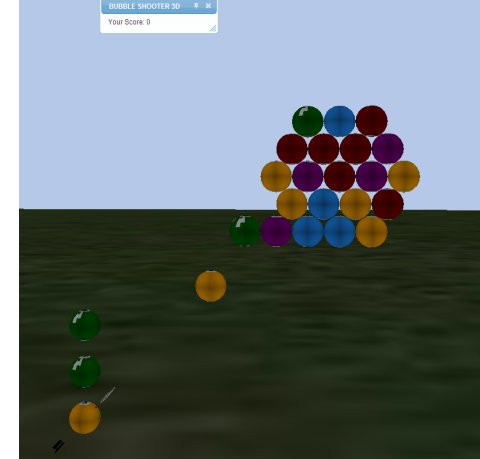
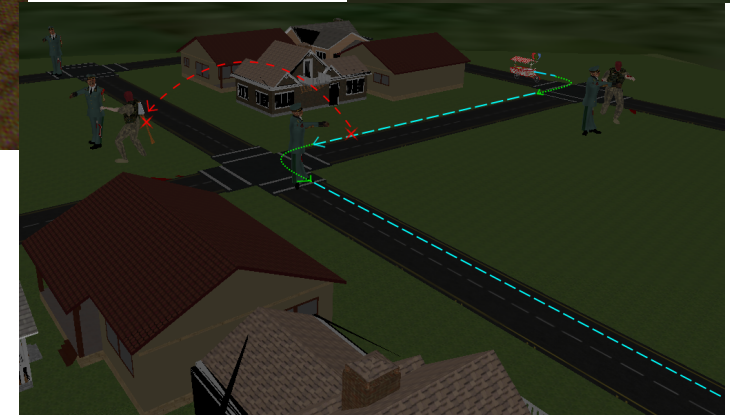
Increasing the amount of allowed customization also increases the complexity of the interface.

Latency

Message sending is much slower than script execution within a single client entity.

Decentralization

Since there is no central authority, every entity has to detect collisions, so frequently a collision is detected twice, once on each entity.



Solutions:

Polling

A polling-based system with regular queries and updates allows painless dynamic modification of the properties of an object's motion in response to events in the virtual world.

Multipresencing – centralize when possible

Host many objects on the same client and only interact across the network when necessary.

Send a snapshot of the world along with messages

Store positions and velocities computed during collision detection and send them with collision notification messages, so the circumstances of the collision can be reproduced on another client after the message has traveled across the network.

Filter out duplicate collisions

Compare the velocities in the message to locally stored velocities, and if the local velocity has already changed, ignore the message.