

TCSS 491 — Computational Worlds

Winter 2012 — Project 1 (phase 3) + Project 2

Due Date: Saturday, Mar. 17

Guidelines

This project should be completed in small groups of two students. The project deliverables (described below) should be submitted to the instructor electronically by the end of the day on the due date.

Overview

Your group will design and implement a distributed virtual world incorporating a basic physics engine. The project is divided into three phases:

- ◆ Phase 1: Implementation of an interactive virtual world.
- ◆ Phase 2: Addition of a physics engine.
- ◆ Phase 3: Distribution of the virtual world across multiple application instances.

The current assignment addresses only the second phase, the remaining phase will be completed later.

Phase 3

In this phase, your group will extend your existing virtual world implementation to support a virtual world spanning multiple application instances, e.g. executing on multiple computers. Objects that exist in one region (program instance) of the virtual world must be able to “travel” to other regions. To achieve this, several issues must be addressed:

- ◆ Maintaining a network of “peer” regions in the distributed virtual world.
- ◆ Detecting when an object within one region should travel to another region.
- ◆ Transmitting an object from one region to another.

Each virtual world should have an *extent*, a set of boundaries that define the perimeter of the region, e.g. the white cube included in the demonstration programs. An object leaving the extent of its current region may either be transmitted to a peer region or “bounce” if no peer region exists in the appropriate direction (see below).

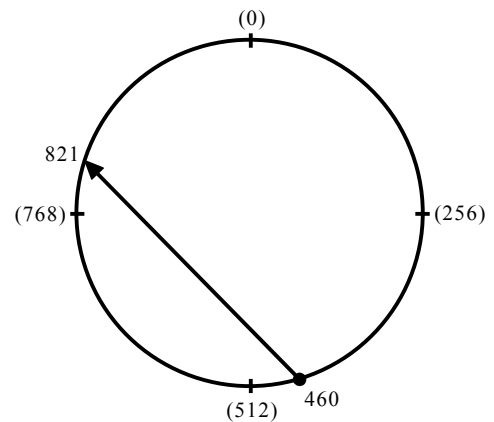
The Peer Network

The peer-to-peer overlay network framework for Chord developed during class should be used to organize the individual regions into a distributed virtual world. Since the Chord protocol organizes nodes into a circle, individual regions will have a position on the circle determined by their (randomly) assigned ID within the network. The framework provides mechanisms for establishing a new network, (dis)connecting to an existing network, and determining which peer network node lays in any given direction. The provided framework is mostly complete, only the portion of the framework specifically dealing with the transmission of objects between regions requires implementation. The bulk of the work in utilizing the peer-to-peer network will be in integrating the framework with your team's existing virtual world application.

Transmitting Objects

There are primarily two issues involved in transmitting objects from one region to another. The first issue is the technique mechanism by which objects can be transmitted from one region to another. Thankfully, Java's serialization mechanism provides a convenient way to transmit objects in their entirety. Modifying the `PhysicsObject` class to implement the `Serializable` interface will fulfill the bulk of the technical preparations. The Chord framework may also be augmented to make transmission of objects more convenient.

The second issue is determining the destination region for a transmitted object. Consider the basic extent used for the demonstration code which has four sides: top, bottom, left, and right. All regions are arranged in a circle, with the top of the region facing inward toward the center of the circle, i.e., a ringworld. The bottom of a region is always solid, thus objects will always bounce off the bottom side. The left side of a region leads to the region's successor in the Chord network. Similarly, the right side leads to the region's predecessor. An object leaving via the top side may travel to any other region in the network based on the angle at which the object leaves its current region. For example, if an object left a region at ID 460 at an angle of 30° left of vertical, the object should be transmitted to ID 821.



Project 2

In addition to incorporating a distributed virtual world, the final form of this project should be more than a “tech demo.” To this end, your group should develop an idea for a game, simulation, or other simple application based on all the technical elements developed in project 1. Email the instructor with your group's idea for validation before investing too much time in implementing the final project.

Deliverables

All Java source code should be archived together, e.g., placed in a `.zip` file, and submitted to the instructor. Only one group member need submit the group's deliverable. Please make sure that both group members' names appear somewhere within the deliverables.