

# Cichlid Simulator v 3.0 (Fall 2015)

File: d:\Cichlid\_Simulator\_3.0\Cichlid\_Simulator\_3.0\_specifications.wpd

Date: September 4, 2015

Client:

Dr. Ron Coleman, Professor, Biological Sciences, California State University,  
Sacramento

Office: 119 Humboldt Hall

Email: [rcoleman@csus.edu](mailto:rcoleman@csus.edu)

Tel: 916-705-2606 (cell)

Your goal:

To create the essence of a computer game involving cichlid fishes in an aquarium. This involves working with a database of players/scenarios; putting 3D graphics on the screen; allowing different viewpoints; having players (fish) move; detecting collisions with static objects and boundary conditions; setting up rules about how players interact.

This project should appeal to students interested in computer graphics and gaming. While this is a simple game, it has the essence common to all computer games, and is easily expandable.

This project is actually the first part of a longer term goal to make a fish game/simulator usable in teaching fish behavior to biology students.

Overall Specifications:

A set of scenarios

1. One fish in the fish tank (display, movement, boundary conditions)
2. One fish and a structural object (rock) in the fish tank (collision detection)
3. Two fish and a structural object (rock) in the fish tank (rules of interaction)

A menu system will allow a user to select one of the above scenarios to populate the tank

The user can Save the game to a named state (and Restore it) at any point in time

Key Requirements:

The program will run on the Department of Biological Sciences laptop computers and any requirements of those machines, e.g., particular version of Java, etc. need to be clearly

specified. Note that these are quite modern and relatively fast machines.

At all stages, the development group must be able to bundle all the necessary pieces into a single file (zip, jar?) to be distributed to the sponsor (Dr. Coleman) for testing, either via flash drive, dropbox or email. This means that at every stage there should be a "working" version of the model. This requires that the team members work together to ensure that the various pieces (interface, graphics, database, etc) integrate with each other.

#### Components:

I imagine that producing a working product will require creating/managing several key components, e.g.,

- Interface (overall look and feel, menus)

- Database (Microsoft Access?)

  - Save and Retrieve

- 3D Graphics (representing objects)

- Collision detection and boundary detection; interaction rules

#### Milestones:

Here is a possible way to break down the overall project into different "phases" each of which could serve as a "milestone".

##### Phase 1: The tank setup

- Set up a 3D world (a fish tank) in the computer

- Display that world on the screen

- The tank back and sides should be portrayed by a non-transparent grid. This will allow perspective, and boundary checking, and also the fish should move "in front of" the grid.

- The view point should be movable

- At the end of phase 1, the development team must distribute a single file to the sponsor that runs.

##### Phase 2: The fish

- Put a "fish" in the tank. The size of the fish (and patterning, see below) and its initial position should be selectable from some sort of interface, possibly allow mouse action to position fish.

- The fish should be represented by a truncated ellipsoid where the round end is the front and the blunt end is the back. Coloration on the ellipsoid (eyespot, stripes? shading) will allow the viewer to see the orientation and direction of the "fish"

##### Phase 3: Moving the fish

Have the fish move around the tank, detecting walls/surface/bottom. Devise rules for how the fish should interact with each of these surfaces.

Phase 4: Add a structural object to the tank

Add menus to add objects to the tank, i.e., a "rock" (cube) of a certain size, at a particular location.

Phase 5: Have the fish move but detect (and react to) the "rock"

Phase 6: Add a second fish to the tank.

Have both fish moving (avoiding the "rock") and interact with each other (devise rules for fish-fish interaction, e.g., big fish swims towards small fish, small fish swims away from big fish, i.e., a chase)

- END