

HAMSTER BALLS

DAVID COSTANZO, ALEXANDER THOMSON, HARLEY TRUNG, AND MATT SILLS

INTRODUCTION

Hamster Balls is a multiplayer first person shooter implemented entirely in Haskell and built on the Functional Reactive Programming framework Yampa. To the best of our (considerably extensive) knowledge, Hamster Balls is the first 3D network game ever implemented in Haskell.

BASIC GAME MECHANICS

Each player controls a hamster armed with a 10-terawatt laser and a 100-terawatt protective hamster ball. Mouse-look and standard W-A-S-D controls are used to aim and move. The game's objective is to shoot and destroy other players. Every laser hit a player takes reduces that player's protective hamster ball strength by the wattage of the laser fired, and when a laser hit takes a hamster's protection down to 0 strength, the hamster dies a gory death and is respawned at its original start location. Once a player achieves a kill, his name is displayed in the top-left corner of the screen along with his score.

ADVANCED COMPUTER GRAPHICS FEATURES

We include two advanced features in our current implementation of Hamster Balls: a Mandelbrot-generated waterfall comprised of 1058 triangles flows (well, hangs) from the top of a crudely-rendered Watson Hall and a lightning-fast N^2 time collision detection algorithm bounces uncoordinated players off walls.

FUTURE WORK

Over the course of the summer we intend to improve the game further. Our ultimate goal is to improve it to the point where it is played competitively in the zoo by bored undergrads, then release it as an open source project. Features currently under development: 360° vision, powerups, faster and more robust collision detection, additional maps, team play, variable mouse sensitivity, gravity, sound effects, and cuter hamsters.

RUNNING THE GAME

Run `/home/accts/agt5/hamsters/binServer` from any zoo machine to start up the server. Then run `/home/accts/agt5/hamsters/binClient <playerName>` to join the game. Note that the game is full screen, so only one instance can be played per computer. To fully appreciate the game, you will want to have at least two players logged in. Note that once any player exits the game (by pressing Esc), there is a very high probability that the server and other clients will crash in the near future. Also, make sure you restart the server with each run. We anticipate that these issues will be resolved soon.

We also have a dynamically-spawnable waterfall in our game. It's generated by computing values of the Mandelbrot set, so it's a fairly complex structure. Unfortunately, this also means that it takes much longer to render than do other shapes. Thus we allow the user to choose to spawn or despawn it. Press L to spawn and K to despawn the Mandelfall/Waterbrot.