

Table Air Hockey

Project Description:

A common game found in many parlors is table air hockey, which is played by two players. The game is very simple: try to knock the puck into the opponent's goal using your paddle. The table has regularly spaced vents for air flow, so that the puck rests on a cushion of air and thus can move without much friction. There are four walls, with 2 small openings on two opposite walls used as goals. The table itself is about the same size and dimension as a pool table, and the puck and the paddles are round. Each player holds a paddle in her hand, and she can hit the puck with it. Whenever the puck goes into your opponent's goal, you score a point. The game ends when a certain amount of points are reached.

Table air hockey has two factors that makes it fun and attractive to most people, it is simple and fast. It is a very easy game to pickup because of its simple rules and mechanics, but it also has depth because a person can be skillful at the game despite its simple nature.

Targeted Users:

The Sunlab, with its network of insanely fast workstations, is actually a great gaming platform. As testified by the popularity of Battletris, there is a demand for simple network games among the students who needs an occasional quick break. Unfortunately, Battletris is also unique in fulfilling this demand, since there are no other games that can provide

the same satisfaction. This project, hopefully, can change that. Table air hockey, if implemented correctly, can also be a simple, fast-paced network game that's as fun as Battletris for Computer Science students.

Features:

- Fast-paced game-play. (required)
- Simple control, completely mouse driven. Specifically, the mouse should control the paddles during the game and the GUI outside the game. (required)
- Quick loading time and a simple GUI. (required)
- 2 players over a network. (required)
- flexible rules such as player adjustable maximum score or time limit. (required)
- Graphically pleasing. (very important)
- Correct "feel" of air hockey – good physics modeling. (required)
- Sound effects. (not important)
- 3D graphics. (not important)
- Computer controlled players with AI and different competence levels. (important)
- Many different tables to choose from. Tables of different shapes, with obstacles and pinball like objects(bumpers, teleporters, elastic walls) that affect the movement of the puck. (somewhat important)
- More than 2 players. Have different tables that can accommodate more than the standard 2 players. (somewhat important)
- Chat client for game setup. (not important)
- Outside table designer, let users design their own customized tables. (not important)
- Can play the game from different perspectives, such as a front view or upward-slanted view. (not important)

Scope:

To satisfy the required feature set is not too complicated, at most requiring several thousand lines of code. The challenge comes in what other features we want to add. Adding more tables and players is relatively difficult, but doable. Having a table designer is probably beyond the scope of cs190 in terms of time needed.

Platform Requirements:

This game has to run on Sun workstations over a TCP/IP network. It should have low memory (< 15 Megs) and CPU usage (< 30% utilization) to ensure that it can consistently run reasonably fast in a busy environment.

Testing:

Each large piece of this project, physics, graphics, GUI and networking can stand alone and does not depend heavily on other pieces. For example, you don't need to have networking to test physics and graphics, and physics just needs a rudimentary graphics engine for visual feedback. Thus each piece of the project can be tested separately without much hassle. While each piece of the project can get very complicated, such as a fancy graphics engine, the complexity of the interfaces between the pieces should not change. Once integrated, it should be pretty easy to tell if the program is functioning correctly, since there is a heavy reliance on visual display. Overall, this project should be very easy to test.

Maintenance:

Hopefully this project does not need to be maintained due to its nature.