

# Networked American Pool Video Game

Group Name: Pool-Maniac

Jiawan Zhang jz2492, Xunchi Wu xw2256, Yichen Liu yl2904, Yuhan Zhang yz2500,  
Zeshi Wang zw2221

- **Overview**

Our team plans to design a 2D American Pool Video Game for two players following the basic American pool rules. The player who shoots in the black ball after finishing all his balls wins the game. This game will be implemented with VHDL and C language and shown on the VGA screen. It can be played by two players in one device or in two devices through the network.

- **Input Sensor Data in the project:**

The input data in our project is mainly keyboard button. The game uses keyboard to get the operation signals from the players, and uses Ethernet 10/100M Port on the DE2 Board to receive data from the other device.

- **Output Data in the Project:**

The kind of output information will be mainly the video data. Audio information, which is the colliding sounds from the balls, would be taken into consideration if possible. The game interface is shown on the VGA monitor. It also uses the Ethernet 10/100M to communicate with other device.

- **Operation Description:**

Use left and right arrow key of the keyboard to rotate the pool cue around the cue ball to set hitting direction. Hold the space key to adjust strength, and release it to hit.

While playing via network, one device sends the package containing the parameters of hitting direction and strength to another device, and displays the results on its screen. Simultaneously, the other device unpacks the package and conducts the same procedure on its screen.

The basic algorithm is to decompose the speeds and moving directions orthogonally. Set the parameters for each side based on the present speeds, directions and positions of balls each calculating cycle. Set the pool table with a constant friction and suppose there's no loss in energy while the two balls collide, and when the ball hits the edges of the table. Use the momentum conservation law to determine the direction and speed of the balls after collision.

To detect ball collision, we first need to work out a fast algorithm to detect if ball collision happens. If not, which is the majority case, go directly to the next cycle, updating positions for the balls. Otherwise, calculate the distance between the center of each ball to find out the collided balls and calculate new moving directions and velocities for them.

- ***Timeline:***

*Milestone 1:*

Setup keyboard

Setup VGA

Setup Ethernet connection

Algorithm design for the ball collisions and movements

Game rule design

*Milestone 2:*

Basic interface design

Algorithm design implementation with C

Game rule implementation with C

Setup Audio

*Milestone 3:*

Finish Ethernet connection

Improve interface (like adding welcome, scoreboard)

Final Report:

Wrap up

Write report

Prepare for presentation