6.111 Final Project: Air Hockey
Project Proposal
Team: Justin Graves and Matthew Guthmiller

Overview

Often times when you play a large table game you need a big open space to have enough room to get the required full motion to interact with the game. You also generally have to take time to set up the table to play on and once the table is built it is generally too large to move out of the room without taking it apart. Not to mention the cost of buying a game table is relatively expensive to the lives of a student. This makes playing games like air hockey or pool difficult to do outside of your home or dedicated game hall. To open up playing these games to the masses we need to eliminate the need for a dedicated room and make it so the game is portable.

The idea behind our project is to eliminate the need of large space and reduce the cost so that anyone can enjoy the game of Air Hockey on the go or in a small room such as a campus dorm room. We will first implement the game to take place completely inside of a screen in which players use buttons on the FPGA in order to control the position, angle and force of the paddles. In this virtual mode the paddles will be modelled as sticks. The puck will move on a fixed board on the screen and will model realistic collisions between the paddle and edges of the board. If there is time we will expand on this model and implement a version that combines the virtual game with reality. We will use the screen to display the game but use a physical paddle outside the screen to serve as the paddle. Each player would move the paddle above the screen and interact with the puck on the screen. A camera would be attached to the paddle to track the position of the paddle and an accelerometer would be use to track the speed.

Design and Implementation

The design of our project first relies on getting the visuals working on the display. Once the visuals are done we move onto trying to make sure that the buttons actually move the objects correctly on the screen. Next we move into making sure collisions getting rendered correctly on the screen. Once these are completed we will move onto making sure the timing of the sound corresponds to a specific event.

Week of Nov 5
Implement and visually test Puck / Paddle / XVGA modules

Week of Nov 12
Implement and test Physics / Game FSM (state transitions, basic gameplay) modules. We will use testbenches to simulate physics inputs and verify expected resulting positions and velocities. Game FSM testbenches will verify that state transitions occur as expected on all possible input combinations.

Week of Nov 19

Implement and test Collision detection / Game FSM (scoring)

Week of Nov 26
Implement and test sound, attempt camera paddle position input

Week of Dec 3
Add second player support

Week of Dec 10
Final polishing / testing and final report

Modules

Collision Checker (Matt)
 Input: [10:0]puck_x, [9:0]puck_y, [10:0]paddle_x, [9:0]paddle_y
 Output: [1:0]collision_type

This module will calculate whether the puck has collided with nothing, a wall, the paddle, or entered the goal opening in the wall.

Physics (Matt)
 Input: positions, board, friction, force, angle of paddle
 Output: [10:0]puck_x, [9:0]puck_y, [5:0]puck_vx, [5:0]puck_vy

This module will calculate the next position and velocity of the puck. The physics module will take into consideration how the puck responds in motion during a collision with the edge of the board or paddle.

Puck (Matt)
 Inputs: [10:0]x, [10:0]hcount, [9:0]y, [9:0]vcount
 Outputs: [23:0]pixel

Given the puck position and current hcount/vcount, the puck module determines what pixel should be drawn of the puck.

Paddle (Justin)
 Inputs: [10:0]x, [10:0]hcount, [9:0]y, [9:0]vcount
 Outputs: [23:0]pixel

Given the paddle position and current hcount/vcount, the puck module determines what pixel should be drawn of the puck.

XVGA

Inputs: vclock
Outputs: [10:0]hcount, [9:0]vcount, vsync, hsync, blank

A basic XVGA module will generate all of the necessary XVGA timing signals.

Game Finite State Machine (Justin)

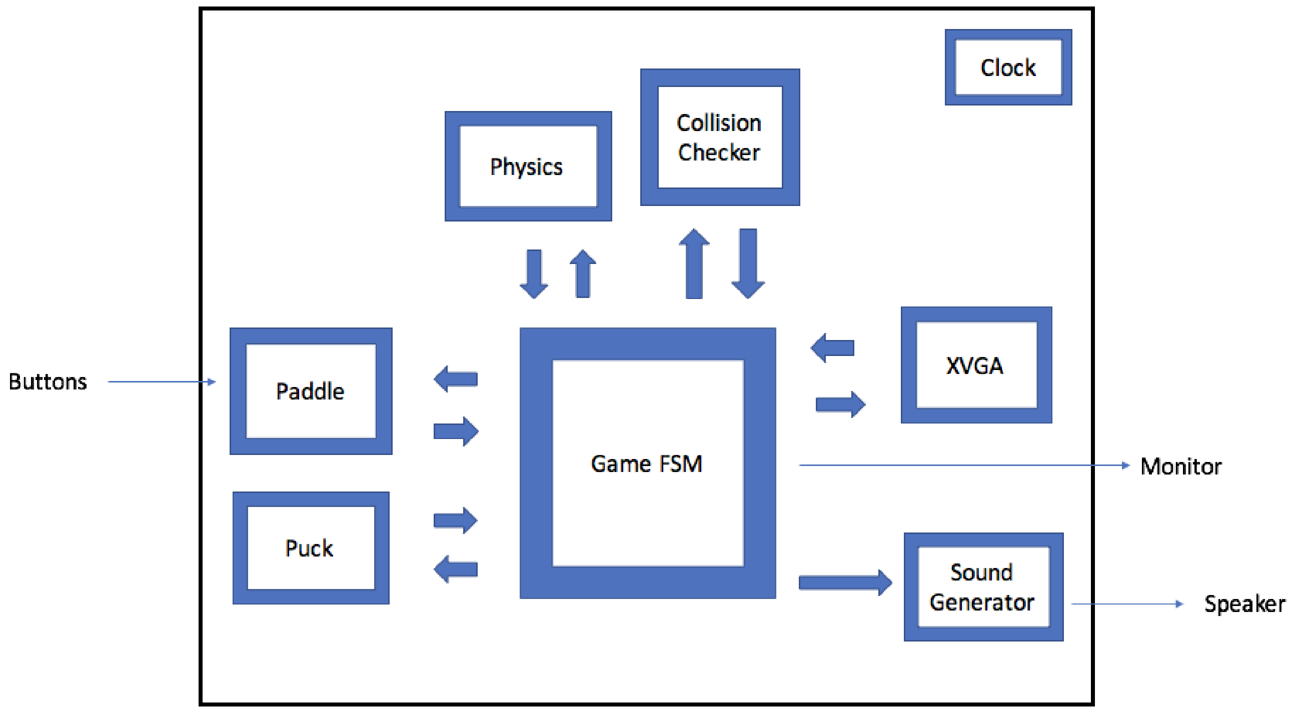
Inputs: puck positions, paddle positions, velocity, friction, board
Outputs: Image to TV/LCD Screen

The Game State Finite machine will be responsible for keeping track of the current state of the game based on the current position of the paddles and puck and score. The Game State Finite State Machine will receive inputs from most of the modules and process the information to in order to show its impact on the current state. The Game FSM will at a minimum of three states The first state is in which the puck is free moving and the only thing affect the puck is the geometry and friction of the board. The second state would be during a collision in which the puck has collided with something. And the third state would be when the puck lands in the scoring region.

Sound Generator (Justin)

Input: Sound signal, Sound type
Output: Sound

This module would receive input from the Game Finite State Machine in the form of whether or not to play a sound (sound signal) and the type of sound to play (sound type). We will use the labkit's on-board ZBT memory to store sounds to represent the different collisions (paddle-to-puck collisions and puck-to-edge collisions) and sounds to represent scoring and winning the game. Initially we will implement simple beeps for sound and add more complex sounds in the memory if time allots. The output of this module will be connected to a speaker to amplify the sound.



Block diagram of Air Hockey Game