Proposal Draft

Angel Carvajal and Sarah Leu

After implementing the simple pong game from Lab 3, the next obvious step was to program a more complicated, entertaining game. This project will create a virtual, two player table tennis game on hardware. The computer screen will display the environment including a split screen with video of the opposing player on the other side of the table. The players will play with paddles whose area will be projected on to the display and sensors on the physical paddle will indicate speed and tilt. Using the graphics implementation knowledge from Lab 3, this project will explore creating objects which move in 3D space and object recognition tracking combined with the fun of table tennis.

In total there are 9 top-level modules that comprise this project. Three of them consist of input/output intractability with the system: Accelerometer sensor, gyroscopic sensor, and the camera. The following three deal with paddle and ball movement throughout the game: Master Ball Module, Paddle Recognition, Paddle Projection. The last three sum up and compile all the movement to generate a playable game with them: Rules, Background module, and Display. Each module has it's own intricacies and complexities discussed in the descriptions below.

1. Accelerometer Module

The accelerometer will be the means by which the players impart the speeds of their swings to the ball. There will be analog devices that will detect the swiftness of each individual's paddle when the ball is within proximity. It' will be attached to each paddle and will send information wirelessly to the FPGA. This information will be computed and processed to convert it to signals the FPGA can handle. This module will then send its information to the ball module when a ready signal is enabled. It'll be continuously working, computing speeds, and sending them to the rest of the system, but the actual data will not be used until the Ball module needs it. There will be two receivers, one for each player.

The accelerometer device will transmit analog data to the FPGA. The transmission will take place wirelessly and be outputted to a receiver on the board. An analog-to-digital converter would then be implemented to translate the incoming data to signals the FPGA board can manipulate. The analog signals the device emits will be discretized and mapped to another set of numbers to make computations easier. Every digitized input will floor estimated to the nearest 50^{th}. For example: a value of 1236 will get mapped to 1200. This will make the arithmetic in the ball module and other devices simpler to manage. This discretized signal will be transmitted to the ball module. This is only one module concerning IO. Now that the ball has speed, it also needs direction to determine its trajectory.

2. Gyroscope Module

The gyroscope will be the means by which the players impart the angles of the paddles to the ball. Without it, the ball will only be able to travel in the same straight-line path forever. Analog devices will detect the tilt of each individual's paddle when the ball is within proximity. A gyroscope will be attached to each paddle and will send information wirelessly to the FPGA. These sensors will detect the change in angle from a calibrated state of equilibrium on a plane horizontal to the screen. This module will then send its information to the ball module when a ready signal is enabled. It will be continuously working, computing tilts, and sending them to the rest of the system, but the actual data will not be utilized until the Ball module needs this information. There will be two receivers, one for each player.

The gyroscopic device will transmit analog data to the FPGA. At first the gyroscope will have to be calibrated so it reads zero degrees when the paddle is parallel to the screen. The transmission will take place wirelessly and be transmitted to a receiver on the board. An analog-to-digital converter would then be implemented to translate the incoming data into signals the FPGA board can manipulate and handle. Now that the paddle movement has been inputted to the board, the camera visuals of the players themselves must be integrated to complete the IO of this system.

3. Camera Module

Two cameras will be implemented to pick up signals from each player. These camera videos will be used in two different ways. One version of the video will be used for paddle placement recognition to detect the position of the paddle so it can later be projected onto the screen in the form of a crosshair. The other version will be downgraded, cropped, and minimized which will be shown on the monitor displaying one's opponent at the end of the table. So, the camera image will be sent to the paddle recognition module (4) and the background module (7).

Two webcams will be implemented to capture a video of each player. Since the board can not intake two video signals, we will use two FPGAs to transfer the images into the system. The situation would run as follows: Player one's camera will be connected to the second player's FPGA and paddle recognition would happen on that same board and result sent to the first player's FPGA and vice versa. The reason for the necessity of sending the video of player one to player two's board is due to the video being displayed on player two's monitor and it would much more difficult to send a full fledged video across a wire to another board. This concludes the input/output modules of the system. Next come the modules managing the movement of the ball and paddle and how they are projected onto the screen.

4. Paddle Recognition/Paddle Projection Module

This module will take in a copy of the video captured by the camera and perform an algorithm to find the paddle within the image. It will determine the coordinates of the paddle in the image and map them onto coordinates of the monitor display. These coordinates will be transferred to the ball module (8) to make ball hits possible. They will also be sent to the display module (9) so to act as a center for a crosshair to indicate the position of the paddle in the world of the game relative to the real world.

Each paddle will have a brightly green-colored piece of paper on their faces. The camera will send over images with these colors standing out amongst the rest of the image. A filter will be performed on the image and a metric will be used to find peaks of green. The approximated center of the peak will act as the main coordinates of the paddle. The coordinates on the image are then mapped onto the possible coordinates for the VGA display monitor. These coordinates are fed to ball module to determine hits. They are also sent to the display module to serve as the center of a crosshair to enable usability and ease when playing.

5. Referee Module

This module will establish the rules of the game and utilize the different conditions to produce an accurate recreation of a realistic table tennis game. It will operate as a score checker, bound checker, round-reset implementer, bounce checker, decider of winner, and overall reset controller.

This module will receive coordinates of the ball and if the coordinates ever leave a certain volume of space within the game, then the ball is considered out and the last player to have touched the ball is the one at fault, giving the other one a point and possession of the ball. If the ball bounces more than once on a return, then it is considered out and the last player to have touched it is at fault, giving the other one a point and possession. The referee module will receive bounce data from the ball module and use it in its computations. When a score reaches a certain value, a winner is decided and in big letters “WINNER” or “LOSER” will be displayed. When the reset signal is enabled, this module will take care of positioning everything to an initial state.

6. Background Module

The background module will be in charge of manipulating everything that's not the ball and paddle on the screen. This involves receiving the camera video from the receivers and morphing them to a suitable picture quality to enable them to be display on the screen of the opposing player without much lag and reduction of game play. This module will also manage the pixels that comprise the game table and net. To top it off, this module will control the surrounding environment as well. The players will have the option of controlling what gets displayed on the sides of the screen.

For the optional surrounding environments, the board will store some select images with good-enough picture quality to be displayed off the sides. Using the buttons on the board, the individual players can choose their own playing environment (eg. One will resemble the lab space, another will resemble the lecture hall, etc.). The modified camera video that this module receives from the camera module will be placed at the right position on the screen. The exact positioning is determined by the placement of the table since the camera image will lie right on top of it. The ping-pong table will be handled using sprites and basic implementation techniques using the blob module from lab 3. A stock image for the net will be stored in memory and used to portray a net partitioning the halves on the table. The same strategy might be used to create the pong table.

7. Ball Module

As in stated in the name, this module will be responsible for the motion of the ball through the 3D space of the display. The inputs from the accelerometer and gyroscope modules will indicate the speed and tilt at which the ball was hit from either player. When one of the players hits the ball, these inputs help determine the return trajectory of the ball to the opposing player. These values will only be transmitted to the module when a ready signal is enabled.

In order to determine if the ball has been hit, the coordinates of the projected paddles are the players is also given as an input. If the paddle location corresponds to the location of the ball when it reaches a table edge, the ball has been hit by the paddle. The location of the paddles will be constantly updated. If time allows, we would also like to implement a curve in the ball's trajectory.

In order for the ball to look as if it is moving toward or away from a player, the size of the ball will change. This will be determined through trajectory calculations. After computing all the possible sizes of the ball, we will implement a look up table in order to reduce the amount of computation the FPGA will have to perform.

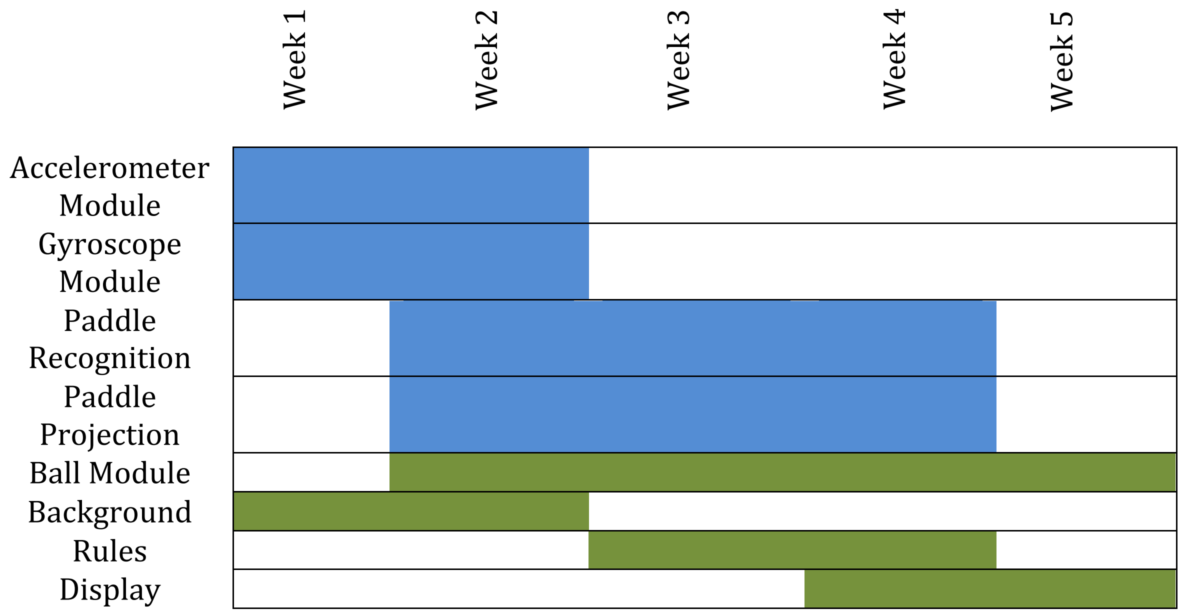
The ball module outputs the ball location to the referee module. This will allow the referee to keep track of the designed rules and will determine when points should be allocated to players. The coordinates of ball will also be transmitted to display for both player one and player two. This is because the visuals shown to the players will differ based on their perspectives.

8. Display

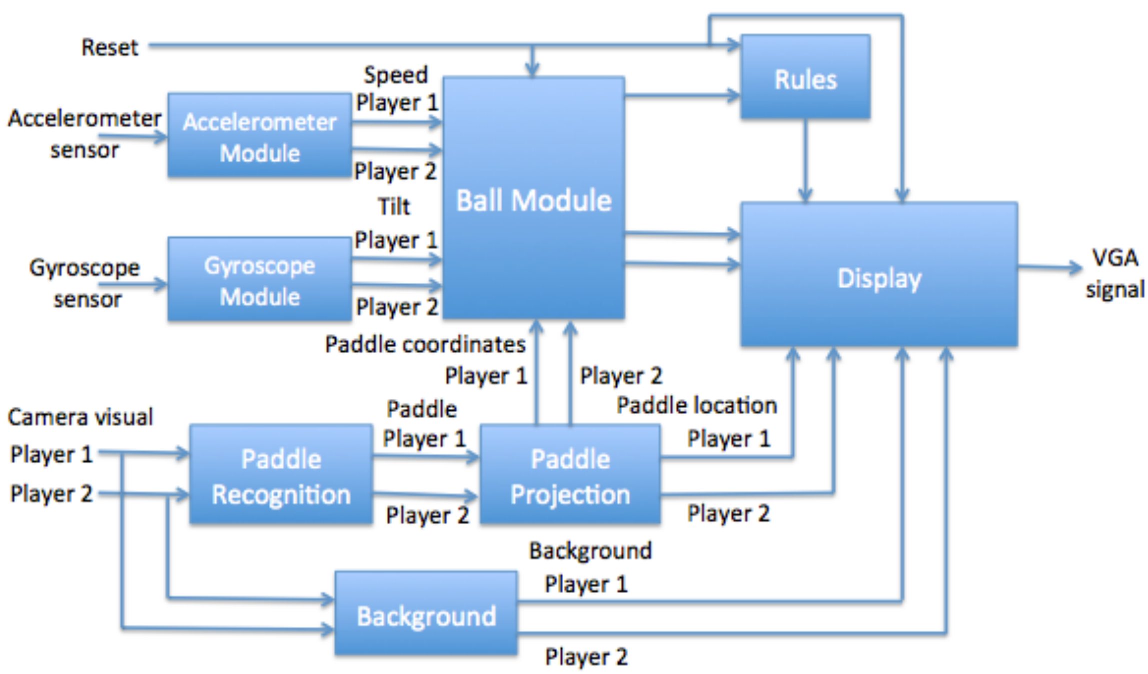
The display module will compile all the visual outputs and render them on the screen. It'll display the balls trajectory, the paddles cross-hair, the surrounding environment, the table, the opponents image, and the scoreboard. It'll take all these signals and send them over the the monitors via VGA ports and produce the final image of the game.

The display modules receives a ton of data from the other modules. From the Rules module, it'll take the reset signals and reprogram the game to an initial state. Along with those signals it'll produce a score board from the score received. From the ball module, it'll take the coordinates of the ball and, using a look-up table, produce an image corresponding to the right depth of the ball. From the paddle projection module, it'll take in the paddle coordinates and produce a symbol to indicate to the players the position of the paddle in comparison to the real world. Finally, from the background module, it'll produce an image of the environment and will tie the game together. When all these pieces are mapped onto the screen and functioning, a pleasing game of virtual ping-pong can begin.

With the plan and design described above, the output of this project will be a table tennis game implemented on hardware. The foreseeable problems have been considered and a number of design decisions were made with these in mind. This game will be an entertaining simulation of a real world two-player table tennis match.



The Gant Chart above shows the week-by-week project timeline from the week November 3rd through the week of December 5th. The blue represents the modules that Sarah will program, and the green represents the modules that Angel will program.



This is the block diagram of the overall system to implement the table tennis game.