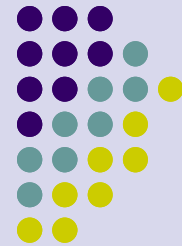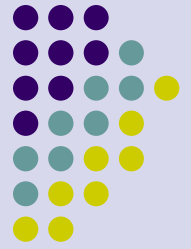# Squash Yourself…

Presented by

Sumit Khatod

Azadeh Moini
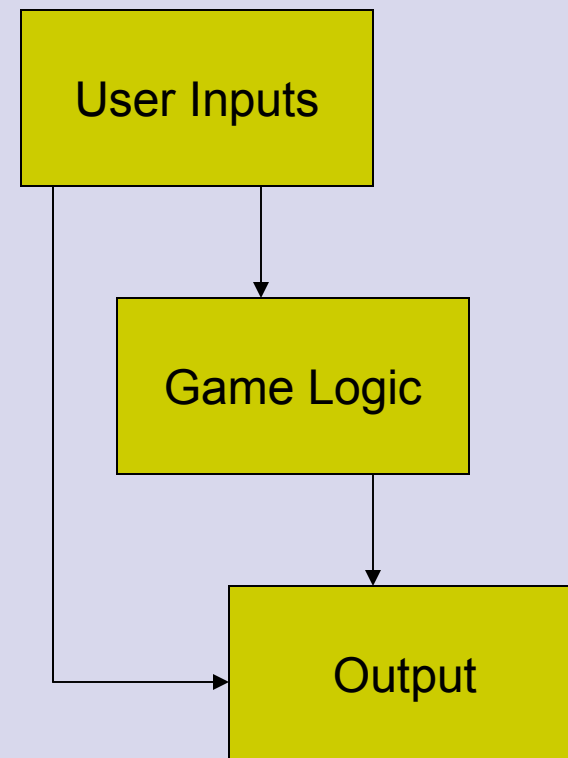
Will Fotsch

# Introduction

- The purpose of our project is to create a two player interactive racquetball game.

- This project was motivated by Nintendo Wii's innovative approach towards making gaming more realistic

- We wanted to create a game that we could appreciate and enjoy playing ourselves. Since sports are something of interest to us and we find using a real paddle more interactive than pressing buttons; we thought this would be fun.
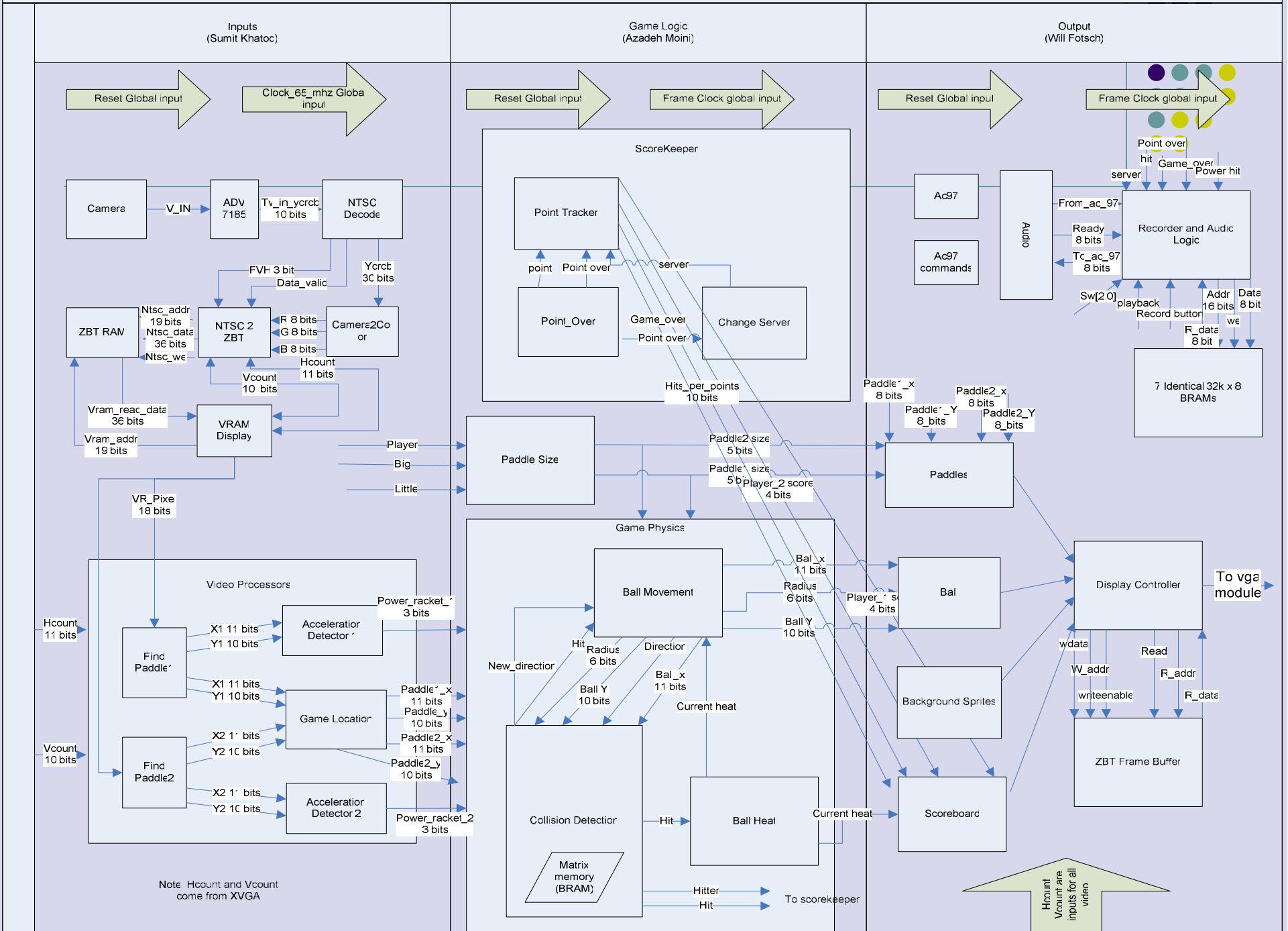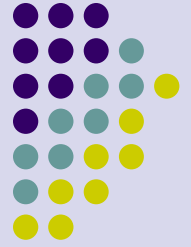
# Division of Labor

- There are three main components
  - User Controls (Sumit Khatod)
  - Game Logic (Azadeh Moini)
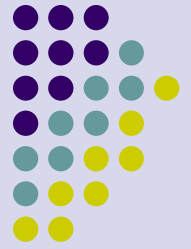  - Video and Audio Output (Will Fotsch)

User Inputs

Game Logic

Output

# Squash Block Diagram

## Inputs (Sumit Khatoc)

Reset Global input

Clock_65_mhz Global input

Camera — V_IN → ADV 7185 — Tv_in_ycrcb 10 bits → NTSC Decode

NTSC Decode — Ycrcb 30 bits →

FVH 3 bit
Data_valid

ZBT RAM

Ntsc_addr 19 bits
Ntsc_data 36 bits
Ntsc_we

NTSC 2 ZBT — R 8 bits / G 8 bits / B 8 bits → Camera2Color

Vcount 10 bits
Hcount 11 bits

Vram_read_data 36 bits
Vram_addr 19 bits

VRAM Display

VR_Pixel 18 bits

### Video Processors

Hcount 11 bits
Vcount 10 bits

Find Paddle1 — X1 11 bits / Y1 10 bits → Acceleration Detector 1 — Power_racket_1 3 bits

X1 11 bits / Y1 10 bits → Game Location

X2 11 bits / Y2 10 bits

Find Paddle2 — X2 11 bits / Y2 10 bits → Acceleration Detector 2 — Power_racket_2 3 bits

Game Location — Paddle1_x 11 bits / Paddle_y 10 bits / Paddle2_x 11 bits / Paddle2_y 10 bits

Note Hcount and Vcount come from XVGA

## Game Logic (Azadeh Moini)

Reset Global input

Frame Clock global input

### ScoreKeeper

Point Tracker

point    Point over    server

Point_Over

Change Server — Point over

Game_over
Point over

Hits_per_points 10 bits

Player → Paddle Size
Big →
Little →

Paddle Size — Paddle2 size 5 bits / Paddle1 size 5 bits

Player_2 score 4 bits

### Game Physics

Ball Movement — Ball_x 11 bits / Radius 6 bits / Ball Y 10 bits

New_direction
Hit  Radius 6 bits  Direction

Ball Y 10 bits    Ball_x 11 bits

Current heat

Collision Detection — Hit → Ball Heat — Current heat

Matrix memory (BRAM)

Hitter
Hit

To scorekeeper

Player_1 score 4 bits

## Output (Will Fotsch)

Reset Global input

Frame Clock global input

Point over
hit   Game_over
server   Power hit

Ac97

Ac97 commands

Audio — From_ac_97 / Ready 8 bits / Tc_ac_97 8 bits

Recorder and Audio Logic — Addr 16 bits / Data 8 bit / we

Sw[2 0]   playback
Record button

R_data 8 bit

7 Identical 32k x 8 BRAMs

Paddle1_x 8 bits
Paddle1_Y 8_bits
Paddle2_x 8 bits
Paddle2_Y 8_bits

Paddles

Ball

Background Sprites

Scoreboard

Display Controller → To vga module

wdata   Read
W_addr   R_addr
writeenable   R_data

ZBT Frame Buffer

Hcount Vcount are inputs for all video

# User Interface

# Summary

- The users uses a ping-pong racquet with an LED placed in the center to play the game
- Both movements and swings are registered by a camera that tracks the movement of the LEDs
- Users can change paddle size using buttons and a switch to select which paddle is being changed
- Why not accelerometers?

# Video Processing
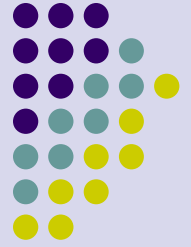
- The camera sends out a YCrCb.
- Using modules provided by the 6.111 staff modified to convert the camera input into RGB, the camera input is stored as a 18 bit color data in a ZBT
- The data is then processed in a video processing unit to detect:
  - Location
    - center of mass of LED
  - Acceleration
    - change in location of led over last few frames
  - Direction of the swing
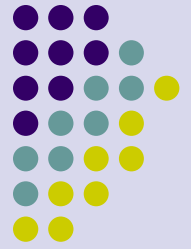    - change in location of led over last few frames
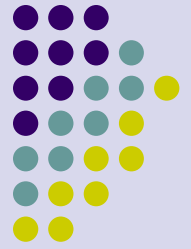
# Game Logic
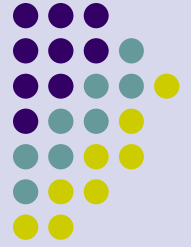
# Game Physics

- Determines:
  - Paddle location
  - Ball movement

- Collision detection (ball & paddle; ball & ground)
  - Current hitter

- Ball heat
  - Paddle power and ball speed
  - Ball movement with respect to angle
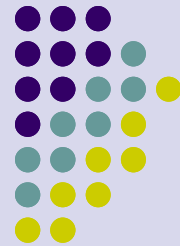
# Scorekeeper

- Whether point awarded or point over:
  - Based on hitter, server, and groundball
- Tracks points:
  - Based on point and server
  - "Rally" for one-player
- Changes server:
  - Based on previous server and point over
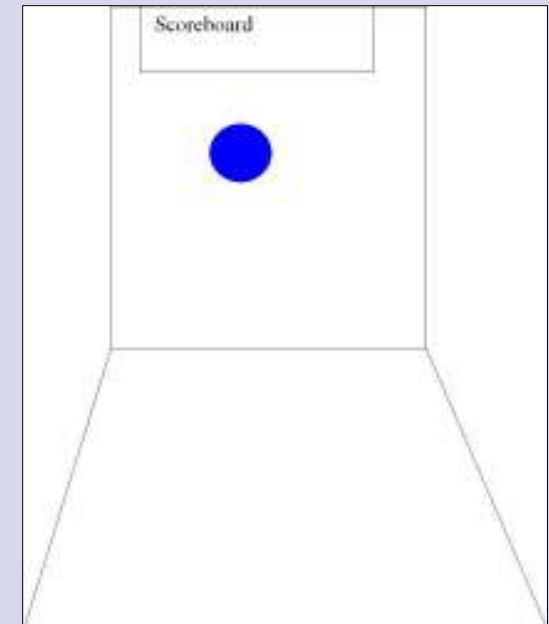
# Outputs

# Display

- On The Screen
  - Court
  - Ball with appropriate size according to the Game Physics
  - Paddle(s) with appropriate size and coordinates according to the Game Physics
  - Scoreboard
    - Player 1 Score, Player 2 Score
    - Rally Score (Hits per point)
    - Ball Heat
  - Frame Buffer will store the display on ZBT Memory

# Audio

- Audio clips will be added to make the game more fun and interactive
- Audio clips for
  - Server Change (Player 1 or Player 2)
  - Hit
  - End of a Point
  - End of a Game
  - Power Hits
- Audio clips will be stored into BRAM
  - The samples will be stored using the microphone and selecting the correct BRAM through 3 switches on the FPGA. A button will be used to indicate record mode.
  - BRAM will be split up into 6 different sound bytes on 32K x 8 BRAMs
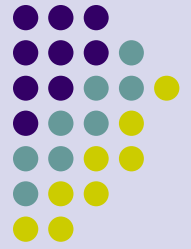
# Project Roadmap

- Phase 1 – Thanksgiving break
  - Basic ball and paddle movements with court graphics
- Phase 2 – December 2$^{nd}$
  - Full one player mode
  - Major enhancements to the graphics (scoreboard)
  - Ball heat and acceleration calculations
- Phase 3 – December 9$^{th}$
  - Two player game
  - Sound effects
  - Added graphic detail