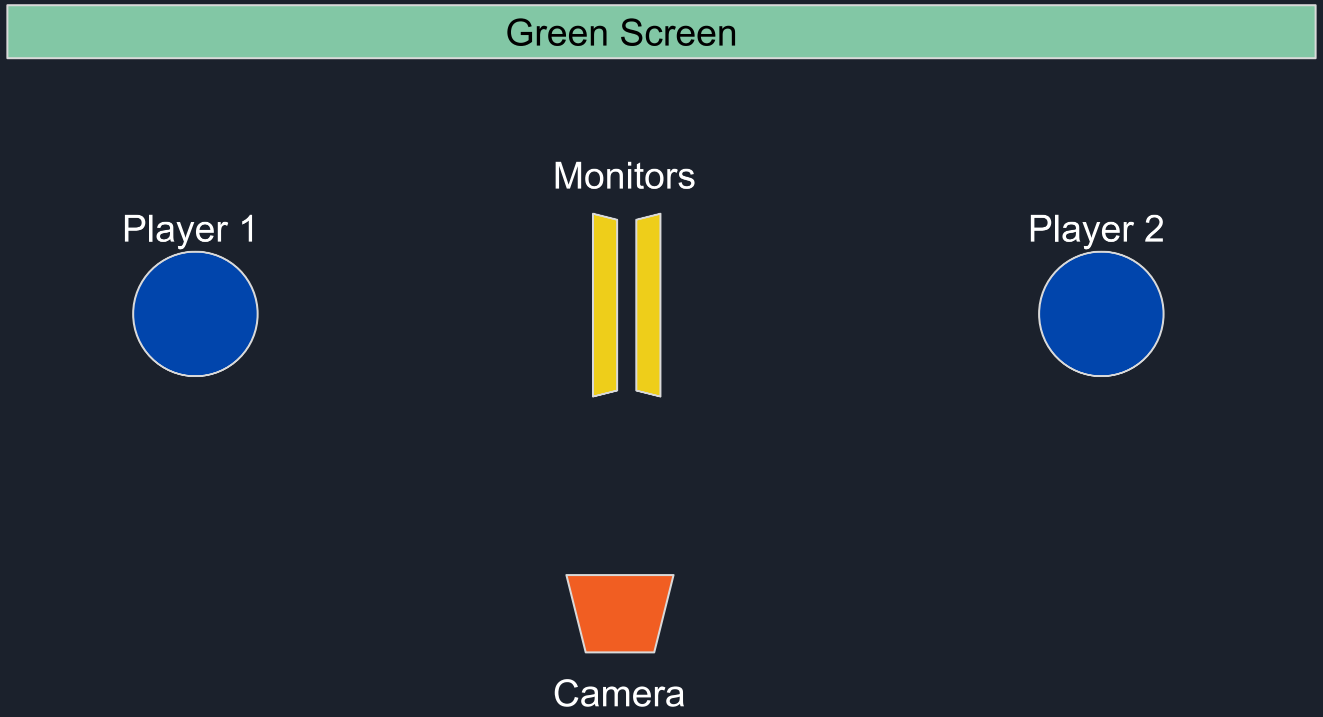
Live Action Pong

Nicholas Waltman and Mike Wang

What is it?

- 2 players hit virtual ball back and forth
- Point scored when ball hits wall behind opponent
- Camera tracks physical movement to control virtual paddle

Physical Setup



Green Screen

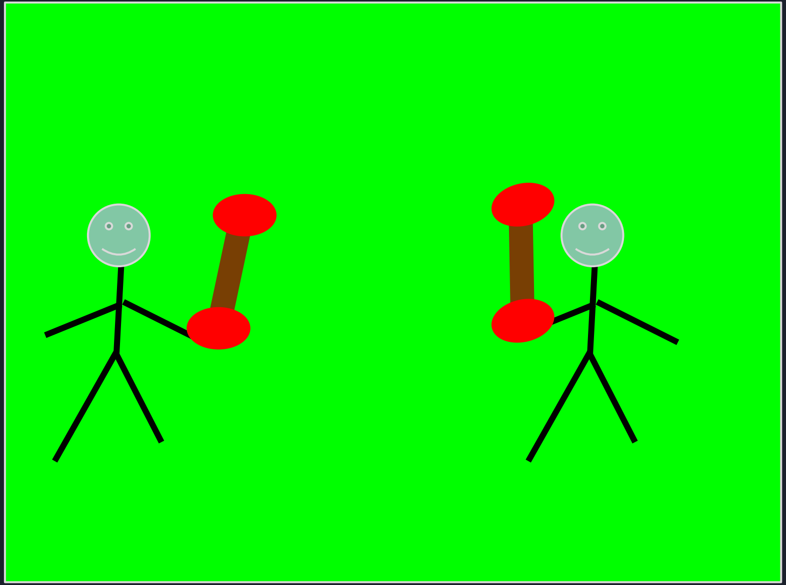
Monitors

Player 1

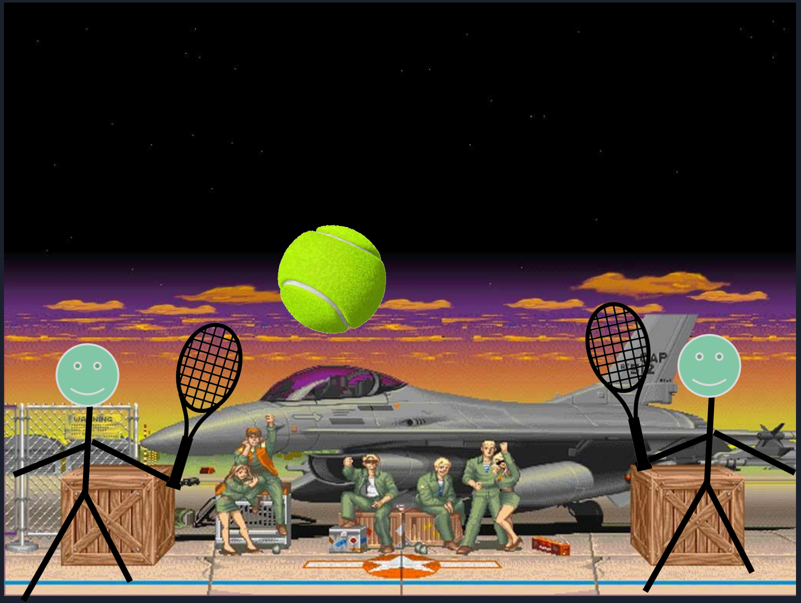
Player 2

Camera

What the camera sees:



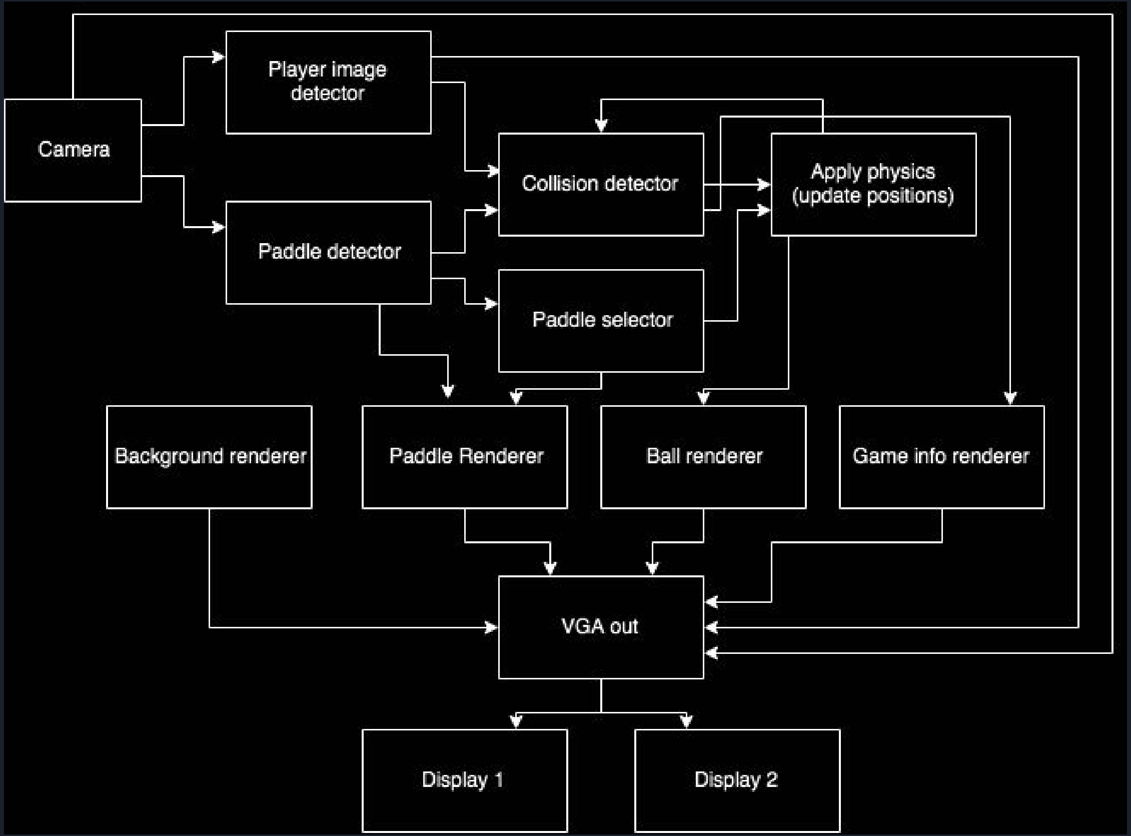
What the monitor shows:



Hardware

- Nexys 4 Board
- NTSC Camera
- 2 Monitors (use a VGA splitter)
  - Both monitors show the same thing
- Green screen
- Physical paddle with LEDs
  - Minimum 2 LEDs lit so FPGA can get paddle angle
  - Number of LEDs lit determines paddle mode
  - Paddle requires LEDs and battery

Block Diagram



Player Image Detector

- Convert RGB output of camera to HSV (module on website)
  - 480x620x24bit buffer for camera input, overwritten by HSV module
  - Threshold comparison saved to 480x620x1bit buffer

- Apply erosion and dilation kernels
  - Eliminate noise from image and smoothen player outline
  - Kernel can be applied to each pixel in parallel
  - Need another 480x620x1bit buffer to hold output

Paddle Detector

- Find x and y coordinates of paddle markers
  - Easy to extract angle of paddle
- Determine number of leds each paddle has turned on
  - This allows for wireless communication!
  - Using different colors was considered, but the brightness tends to make everything look white
- Find velocity of paddle
  - Decompose velocity into translational and rotational for more accurate physics simulation

Collision Detector

- Calculates paddle area using paddle position and angle
  - Use this and ball position and velocity to calculate ball-paddle collision
- Uses ball position and velocity to find wall collision
- Time permitting: attack collision
  - Use output of player detection module to find where the player is and see if attack hits player

Physics Simulation

Collision Detector module will let us know which objects collide (including walls)

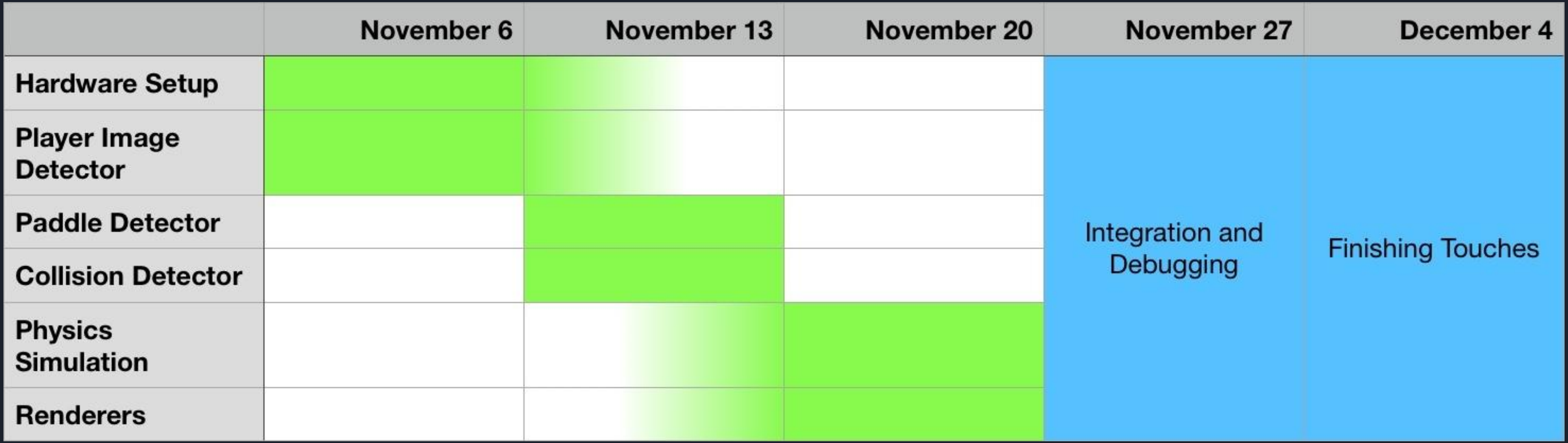
- Stores the positions and velocities of moving objects
- If there’s a collision:
    - Take relative velocity and angle between the two objects
    - Calculates new velocity
- Otherwise:
    - Just update positions using current velocity
- Will update scores when the ball goes out

Outputs the new positions of moving objects to the collision detector and graphics renderers

Renderers

- Ball: simply use pipelining techniques from lpset 8
    - Potentially make more interesting with a shape changing ball
- Paddle: use paddle position and angle to orient shape
    - Either use preloaded sprite or simple equations of rectangles and circles to create graphic
- Background: displays static image
- Player: looks at the output of the player detection module, if the pixel is a player, display the output of the camera

Timeline



	November 6	November 13	November 20	November 27	December 4
Hardware Setup	🟩	▨		Integration and Debugging	Finishing Touches
Player Image Detector	🟩	▨			
Paddle Detector		🟩			
Collision Detector		🟩			
Physics Simulation		▨	🟩		
Renderers		▨	🟩		