

Piano Dance Revolution

- CHARLIE'S ANGELS

...and Charlie

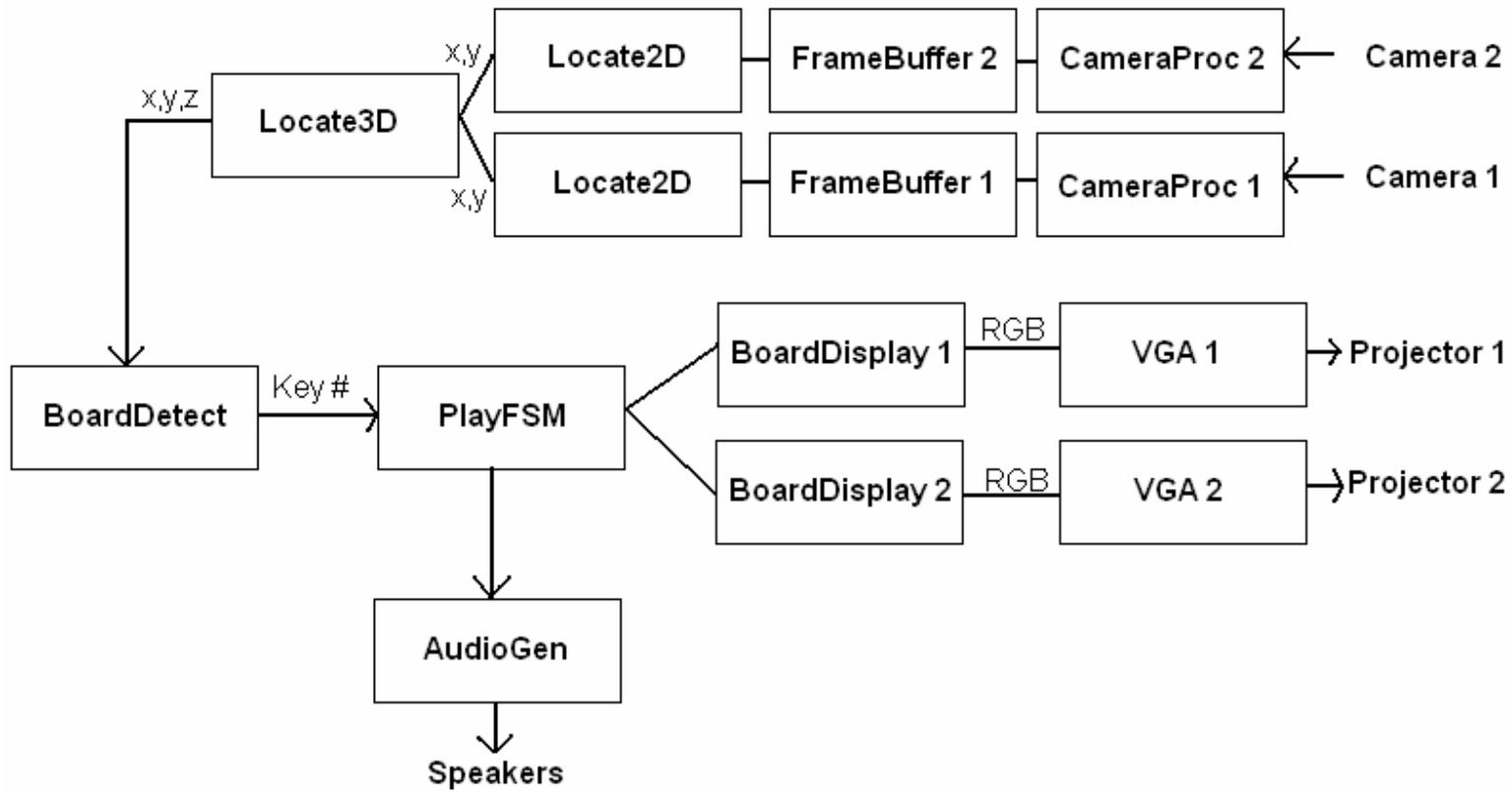
I. Inspiration



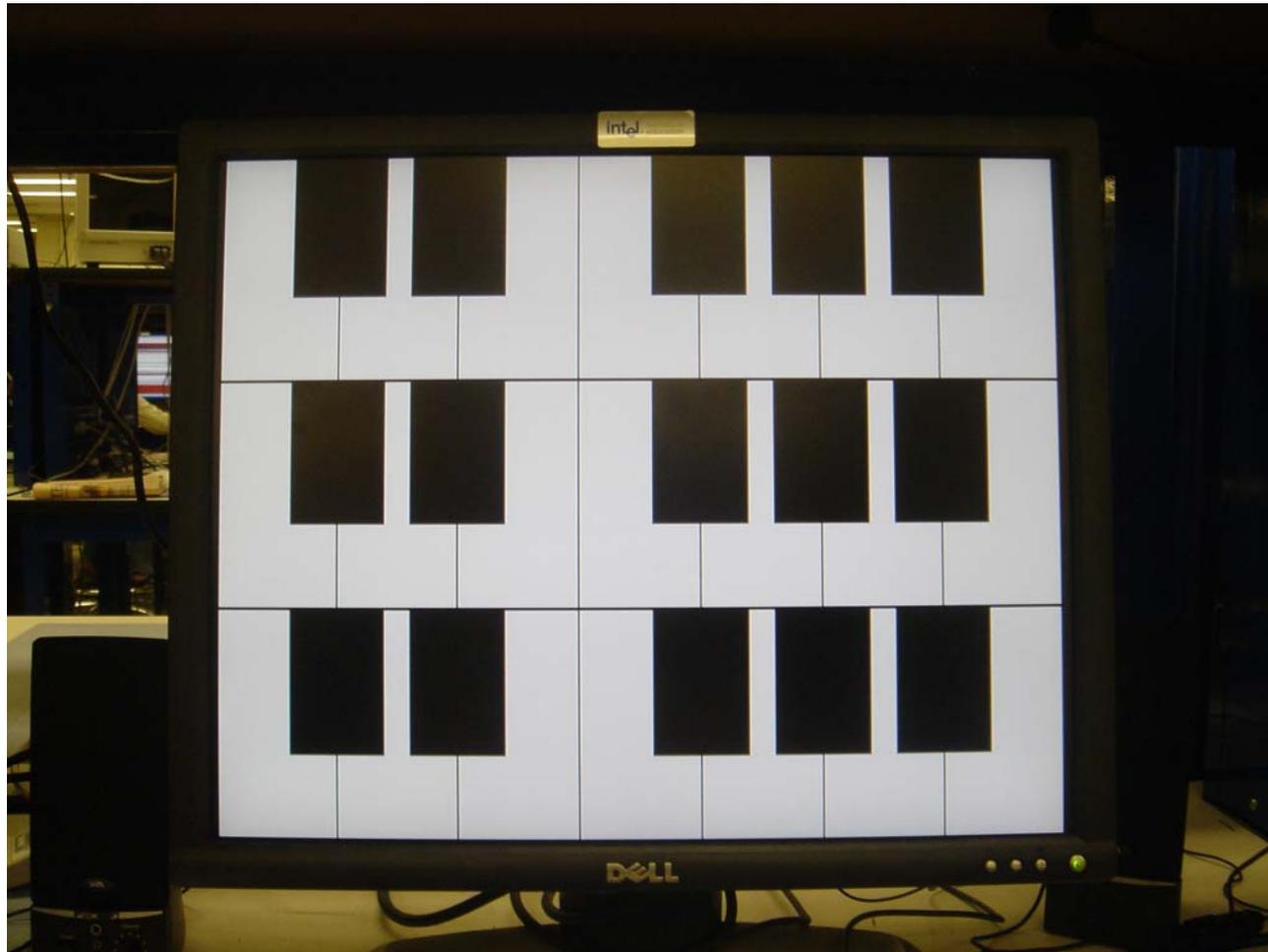
Courtesy flickr user charles.hope. Used with permission.

- Inspired by FAO Schwarz piano
- Consists of a projection, position detection, and audio output system
- Piano keyboard projection on the floor
- User interacts with piano by stepping on the keys
- When activated, the keys light up and play appropriate note

II. System Block Diagram



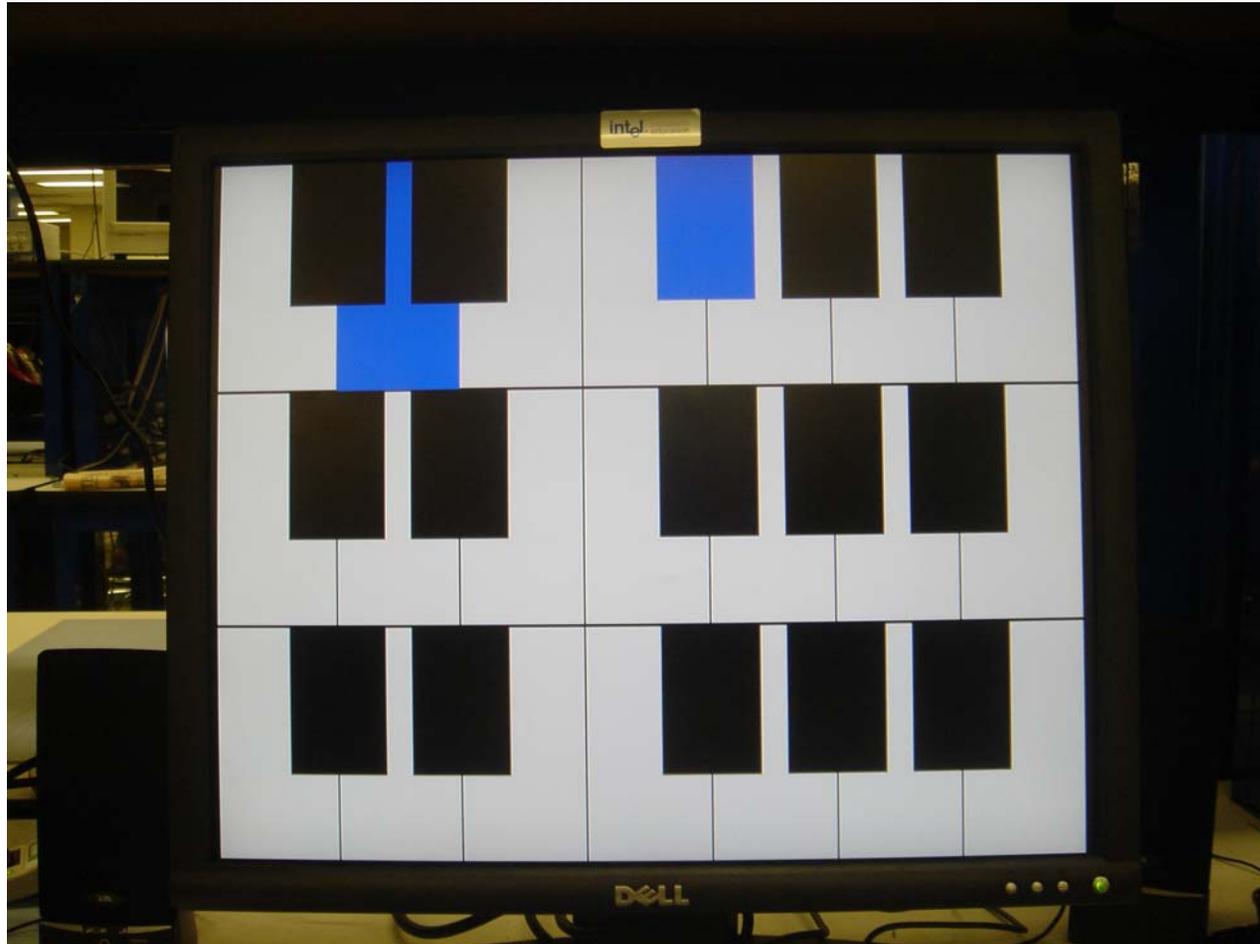
III. Projection



III. Projection

- Two projectors, two FPGA's
- Rectangle module to generate rectangle with border
- Modules black_key and white_key to generate black and white keys
- Module row to generate a row of keys
- Module piano_board to stack up three rows of keys
- 35 bit number into piano_board, each bit encode for a key
- 1 – key is stepped on, light up

III. Projection



IV. Position Detection

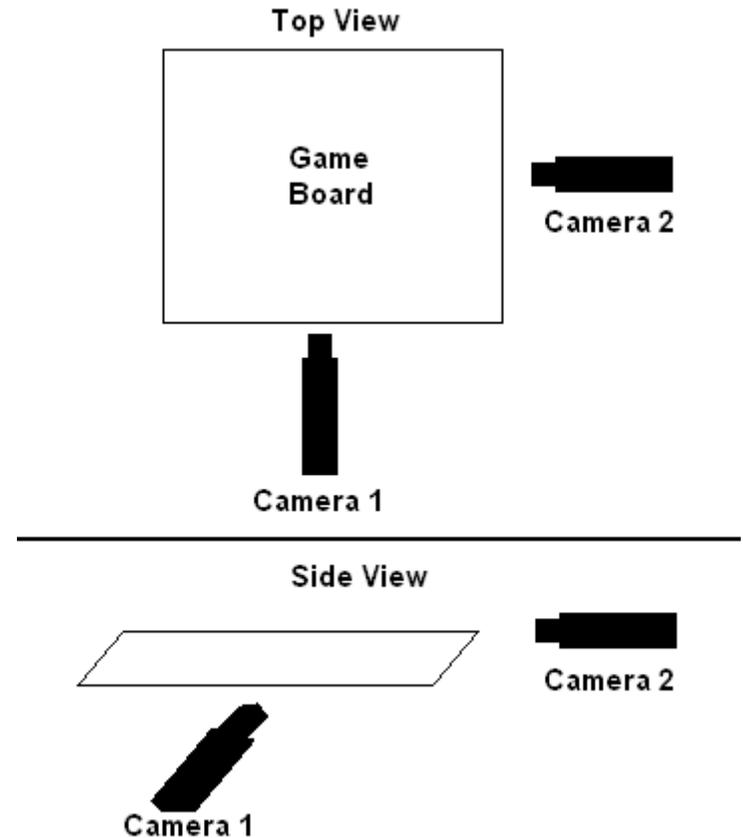
- Bright red band is worn on the player's ankle.
- Two cameras connected to two FPGAs.
- **CameraProc:** Processes the camera input and stores the RGB data into the frame buffer.
- **FrameBuffer:** A module which stores the input signal of CameraProc into the on-board ZBT (Zero-Bus-Turnaround) memory for later processing.

IV. Position Detection

- **Locate2D:** Examines the frame buffer of a single camera and determines xy coordinates of the user's ankle from the viewpoint of the camera. Output will be an xy coordinate, or a signal indicating no visible player.
- **Locate3D:** Receives both xy coordinates from both instances of Locate2D. Uses simple triangulation to determine the height and depth of the user's feet. Output is an xyz coordinate.
- **BoardDetect:** Based on the Locate3D xyz coordinate, determines which key is being pressed, if any.
- **PlayFSM:** State machine for game play. Receives BoardDetect output. Sends signals to AudioGen module and BoardDisplay modules to play the appropriate sound and light up the appropriate keys.

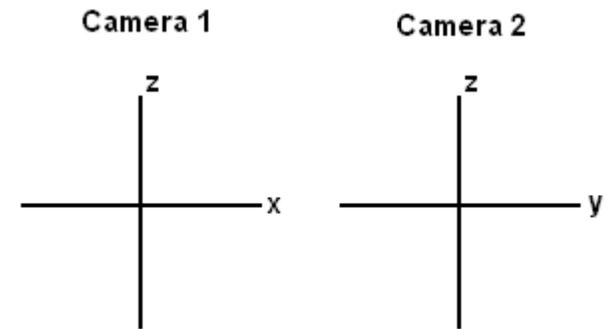
IV. Position Detection

- Cameras are at a 90 degree angle to each other.
- Cameras are at ground level.



IV. Position Detection

- Both cameras measure the height of player's ankle.
- One camera receives x position, other camera receives y position.
- Combination is a single (x, y, z) coordinate.



V. Audio Output

- Pitch input from PlayFSM Module
- Create sine wave using an instantiation of a sine/cosine table from IP CoreGen
- Sample sine wave at 48kHz
- Output PCM data to left and right channels of AC'97 Codec

V. Audio Output

- AC'97:
 - 12.288MHz bit rate
 - 48 kHz sample rate
 - 256 bit frames
- Mixed audio project – will use multiple clock domains
- Synchronize inputs to ac97_bit_clk

VI. Game Mode

- Prerecorded songs in “storage,” played upon selection
- Corresponding keys light up with note
- User must step on appropriate key to accumulate points
- Score is calculated and displayed

VII. Lab Kit Communication

- At least two Lab Kits need to be connected and communicate information, since we have two VGA projectors and two NTSC cameras.
- To minimize wiring, lab kit communication will be through a high speed serial protocol.
- One lab kit will be primary, and the other a slave. This minimizes communication requirements.

Thank You!