6.111 DIGITAL DESIGN

SUPER FPGA BROS

Douglas Albert
Kevin Marengo

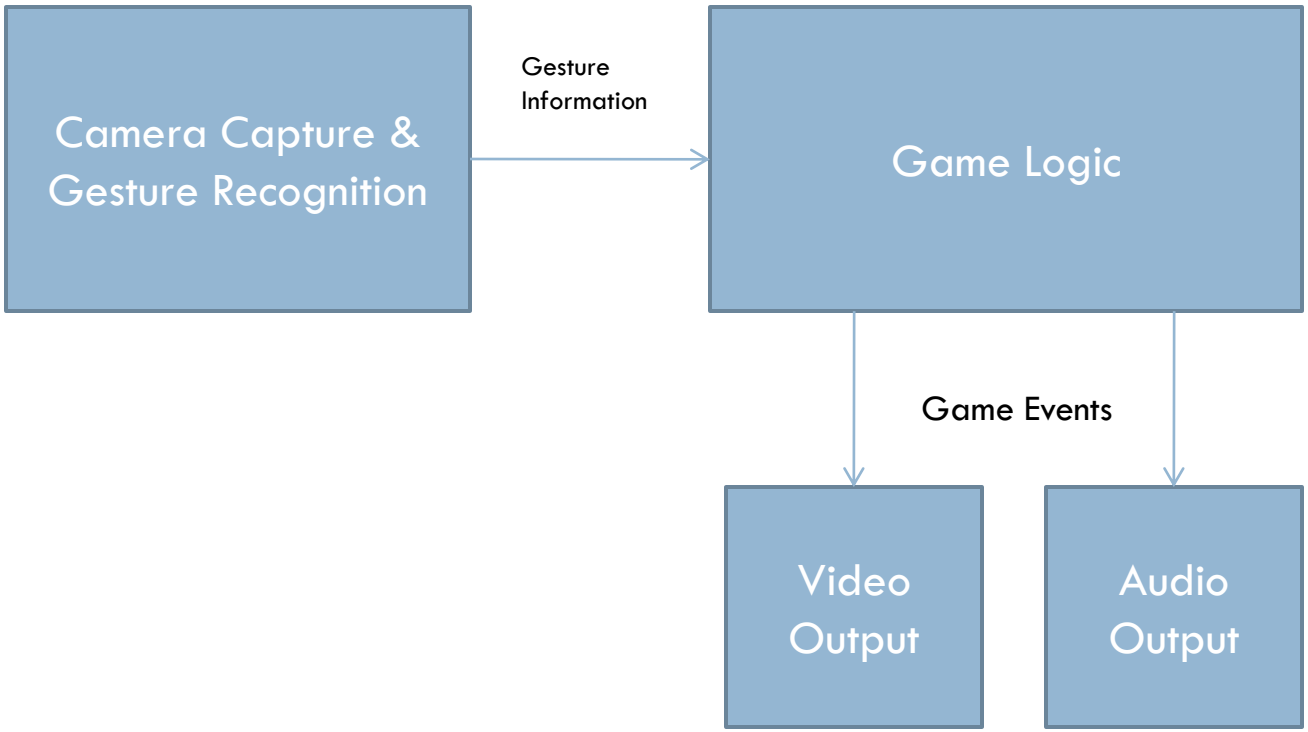
Overview

- Project Description
  - Objective
  - Descriptive Overview
- Technical Description
- Project Timeline
- Q & A

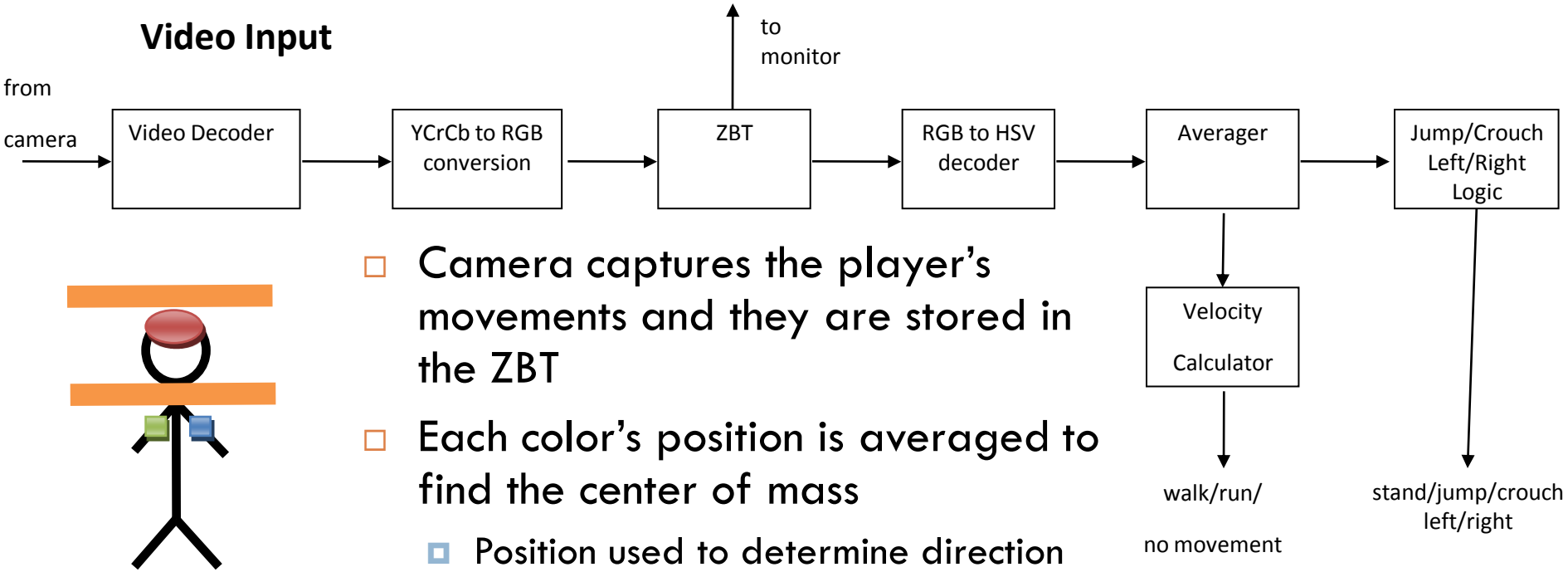
Project Description

- Overview
  - Input is provided by user gestures, initially upper body to be later expanded to legs as well
  - Game world is provided by predefined levels and include obstacles and environmental hazards
- Potential Additions
  - Scoring & Time Tracking
  - 2 player competitive “ghost” mode
  - Dynamic representation of player character

High Level Block Diagram

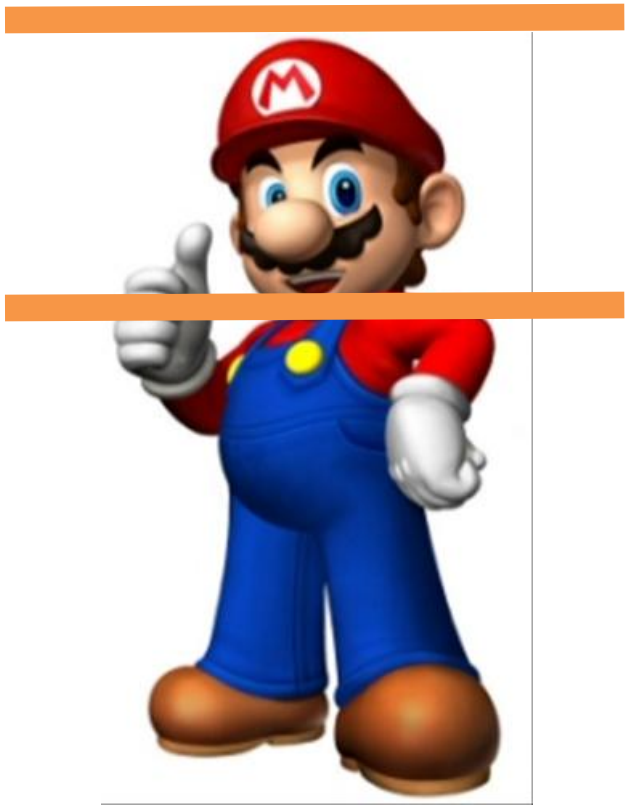


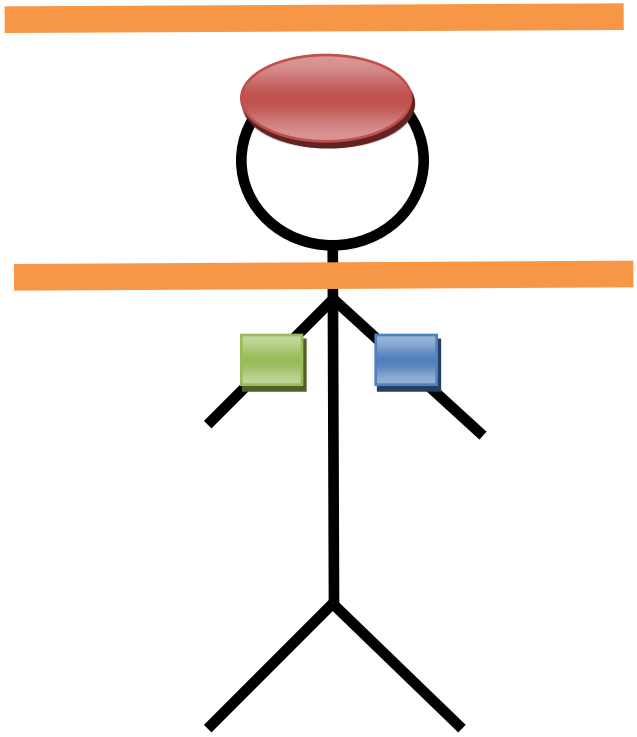
Video Capture & Gesture Recognition

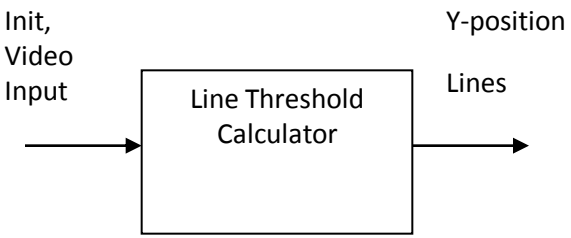


- Camera captures the player's movements and they are stored in the ZBT
- Each color's position is averaged to find the center of mass
  - Position used to determine direction and action
  - Velocity of arm patch used to determine walk vs. run
- The actions are output to the Game Logic

Gesture Recognition



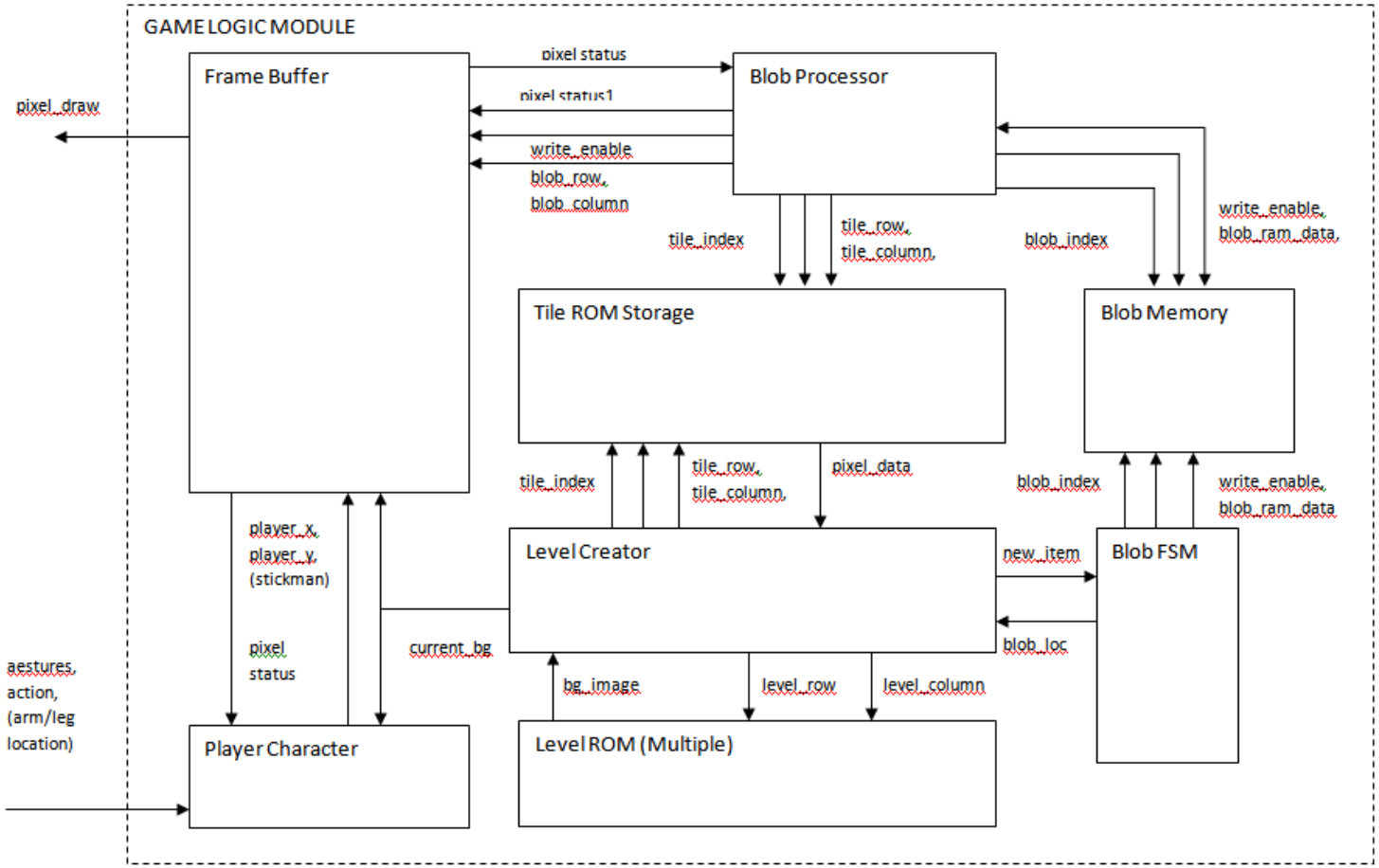




Game Cartridge/ Main Logic

- Implement a 2D Game “engine” in hardware
- Abstract away controls and audio output
  - Camera capture and gesture module passes input signals here
  - Event signals triggered in the engine can trigger audio outputs

Game Cartridge/ Main Logic

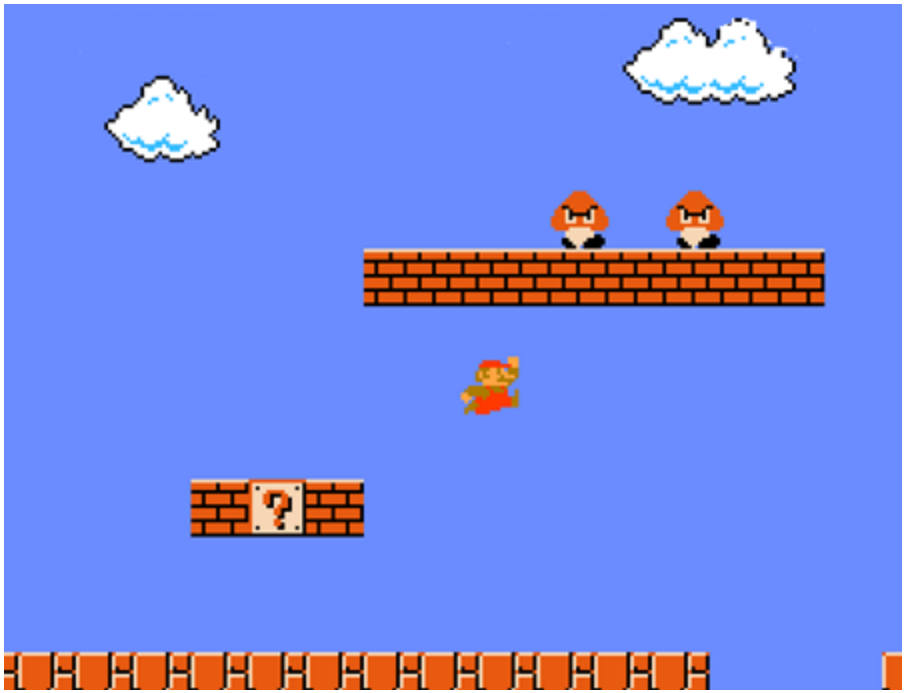


Game Engine

- Level Creator
  - Writes current level layout to Frame Buffer
- Tile ROM
  - 16x16 tiles to create graphics with
- Level ROM
  - 15 x 256 x
  - Levels are made up of tiles
- Blob RAM
  - Holds information about actors on screen: enemies, items, etc.
- FSM & Processor
  - Collision detection & enemy movement behavior

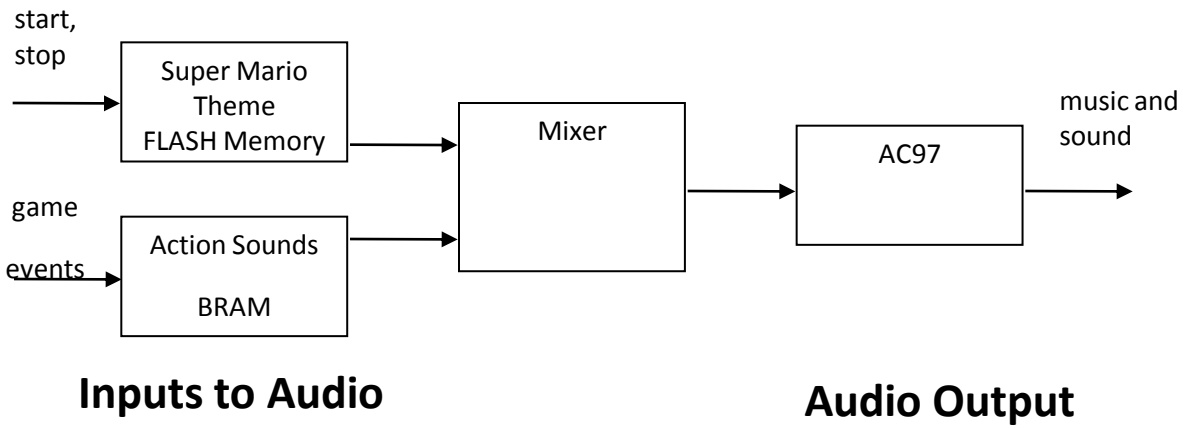
In-Game UI Mockup

Graphics



- Base entity is 16x16
  - Mario is 16x16
  - Big Mario 16x32
- Store sprites in 16x16 chunks
- Use a framebuffer for glitchless output

Audio Output



- The theme music is loaded into the FPGA FLASH memory
  - Song loops, starting when the game starts and ending when the player dies or completes a level
- Action sounds like jumping are stored in a BRAM
  - Game events from the Video Output and Game Logic Output trigger these action sounds
- Theme music and action sounds are combined in the mixer and output as sound via the AC97

Project Timeline & Milestones

- Planning is complete, now to implement
- Major Milestones
  - Rudimentary Game Logic & Functionality
  - Graphical Overhaul and Gesture Control
  - Audio Overhaul and Scoring Functionality
- If we have time
  - 2 player competitive race
  - Additional levels & Items

November 2009

SUNDAY	MONDAY	TUESDAY	WEDNESDAY	THURSDAY	FRIDAY	SATURDAY
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

Design Presentations

Video Output

Implement Basic Game Logic

Camera Capture

Gesture Recognition

Sprite Generation

Audio Output

Initial Debugging

December 2009

SUNDAY	MONDAY	TUESDAY	WEDNESDAY	THURSDAY	FRIDAY	SATURDAY
		1	2	3	4	5
	Initial Debugging		Implement Additional Functionality			
6	7	8	9	10	11	12
			Project Checkoff			
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

Questions and Discussion