
SHOTS:
Duck
Hunt
Game

Features

- Touchless control (camera based)
- Active play
- Can be adapted for mobile devices
- Faster performance with FPGA implementation
- Complete hardware implementation of game engine
- Scaled to fit any display resolution
- Play with any handheld object or your hand

Main Outline

GAME ENGINE

FLOW CHART

- The marker is captured in calibration mode
- Can return to calibration mode via reset
- Pressing the start button begins target generation
- Successful hits will increase the score
- Not hitting the target before it reaches the boundary decreases the score
- The game ends when score is less than 0 (lose) or greater than 15 (win)

MODULES

OV7670_top (game engine)

- Instantiates all other modules
- Decides the state of the game –calibration or game play

OV7670_capture (Locator)

- Capture module works in 2 modes
- Calibration-It calculates the object color(pixel RGB values) to track
- Track-It returns the location of the calibrated object by finding the average of all coordinates with the calibrated pixel values

VGA module:

- Instantiates hit-detect, position generator, scoring and vga control modules

Hit–detect module

- Gives hit signal when the marker overlaps the target. It gets the coordinates of the target and marker, compares the difference between marker's X and target's X with the dimension of the marker and calculates overlap area if the former is smaller than the latter.
  If area is greater than a threshold it gives an output hit signal.

Position Generator

- Receives a random value from LFSR and generates a square target from that position. The seed to the LFSR comes from another seed module.
- Contains following states:

1. Init- start of game play from buttons.
2. Fetch- get new target when the target goes out of boundary or is hit by the marker.
3. Ack_Wait-wait signal.
4. Move- move the target diagonally depending on the slope calculated.
5. Hit- increment hit counter and direct to fetch state for new target.

Seven- Segment display

- Receives score from VGA module, converts it to BCD format and displays in the 2
- 7- segment displays. Displays are switched at 12 ms.

VGA control

- Generates HSYNC and VSYNC for VGA module and handles timing constraints.

CHALLENGES

- Smoothening of the marker
- Clock synchronization- capture module (25 MHz) and VGA(108MHz) run at different clocks.
- Hit signals
  - Setting the right threshold value and handling the next target when hit happens
  - Count Logic
- Audio playback from memory
  - Using MicroBlaze with data in BRAM, played through AXI GPIO to PWM out
  - Using MIcroBlaze with data in BRAM, played through a custom AXI slave IP
  - Using MicroBlaze with data in DDR, played through custom AXI slave IP
  - Audio data, PCM to PWM logic, adjusted to system clock

FUTURE WORK

- Currently level up is implemented by making the targets move faster, we would like this to also include the randomness in the target's path
- Improve background, texture (marker and target)
- Debug and implement audio

Lessons Learnt

Always simulate and test prior to implementing in hardware

Video

- Smoothening
- Scaling the display
- Putting multiple objects

Gameplay

- Instantiate and synchronise

Audio

- Understand analog data represented digitally and how to recreate it
- Access synchronisation over the AXI interconnect