

FPGA Hunt

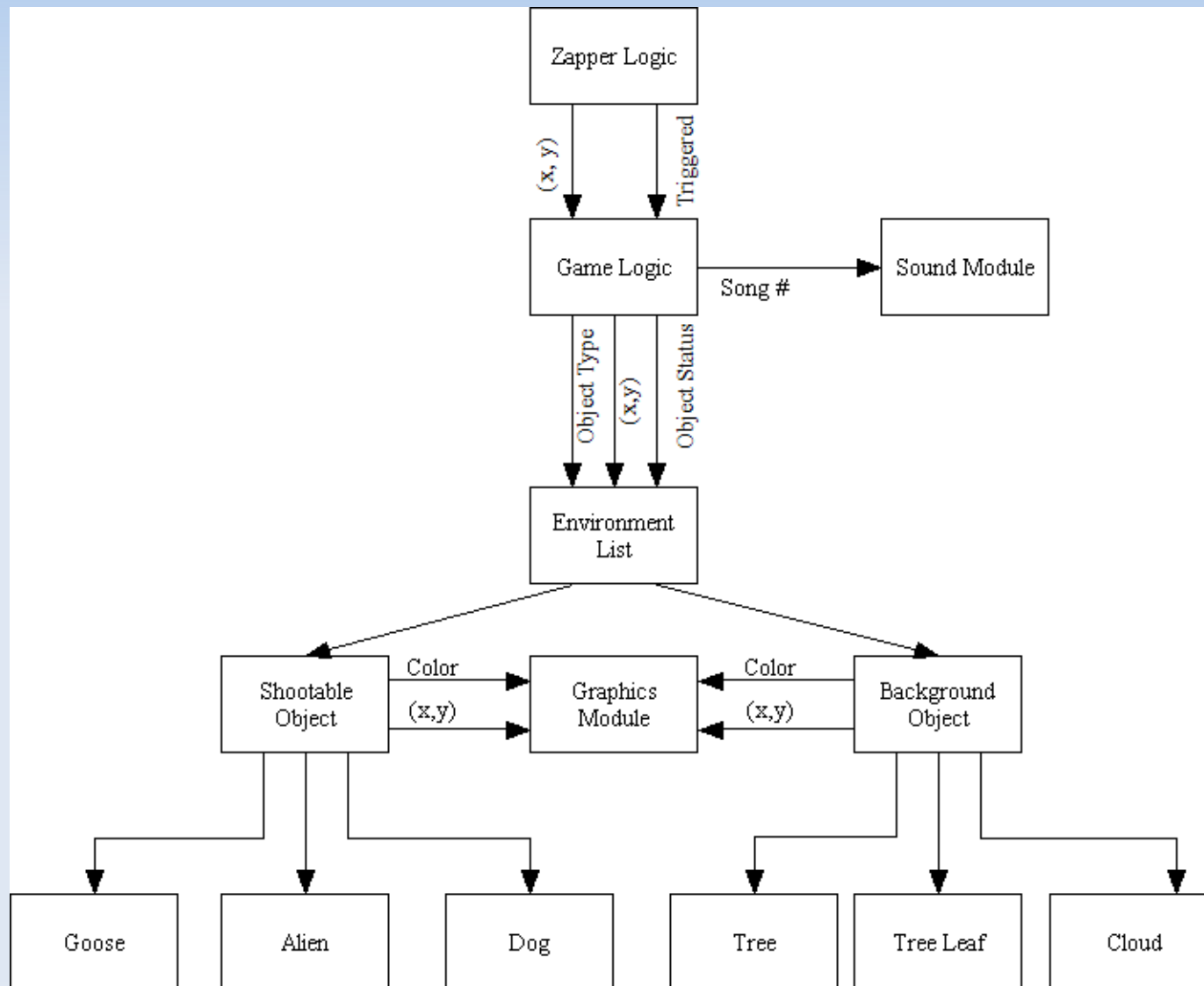


Shooting the dog... Priceless

FPGA Hunt Overview

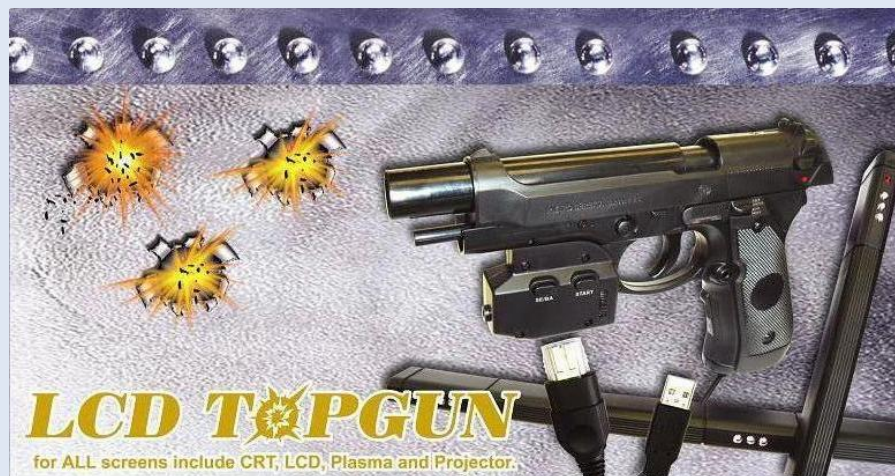
- Duck Hunt 2.0
 - More Creature Types
 - Dynamic Gaming Experience
 - Shoot the Annoying Dog
 - Improved Graphics
 - Functions on any screen (LCD/CRT...)

Block Diagram



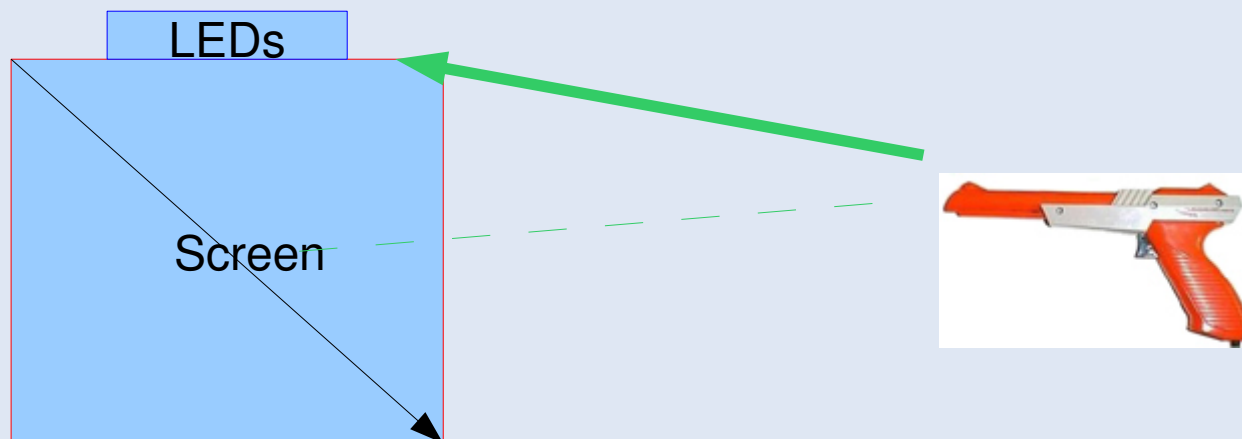
Zapper Functionality

- Uses LED's around LCD to determine direction
- Connects to PC via USB
- Will Utilize Logic Analyzer to Help Integrate



Zapper Logic

- Informs Game Logic Unit when Trigger Pressed
- Translates coordinates from Zapper to GLU
- Calibrates Zapper for Any Screen Size



Game Logic Module

- Inputs
 - -Triggered
 - -(x,y)
- Outputs:
 - Object type
 - Object Position
 - Object Status
 - Game Time
 - Player Score
 - Player Health / Ammo (If Time Permits)

Game Logic Internal Workings

- Finite State Machine
 - Updates every 1/60th of a second
 - Each state correlates to a scene/level
 - At beginning of each level, spawn enemies
 - Upon trigger, test position for enemy collision
 - Vary fuzziness related to current difficulty level
 - Upon certain events (enemy death)
 - Update Score, Change State



Start of State



Current State

BAM!

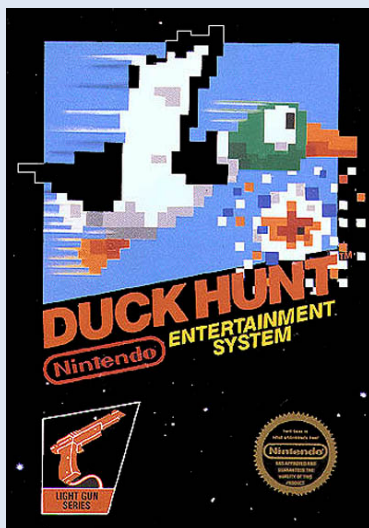


Next State



Graphics Overview

- Objects read from Environment Lists
- Graphics farmed out to submodules
- Pixels rendered in order of depth



?

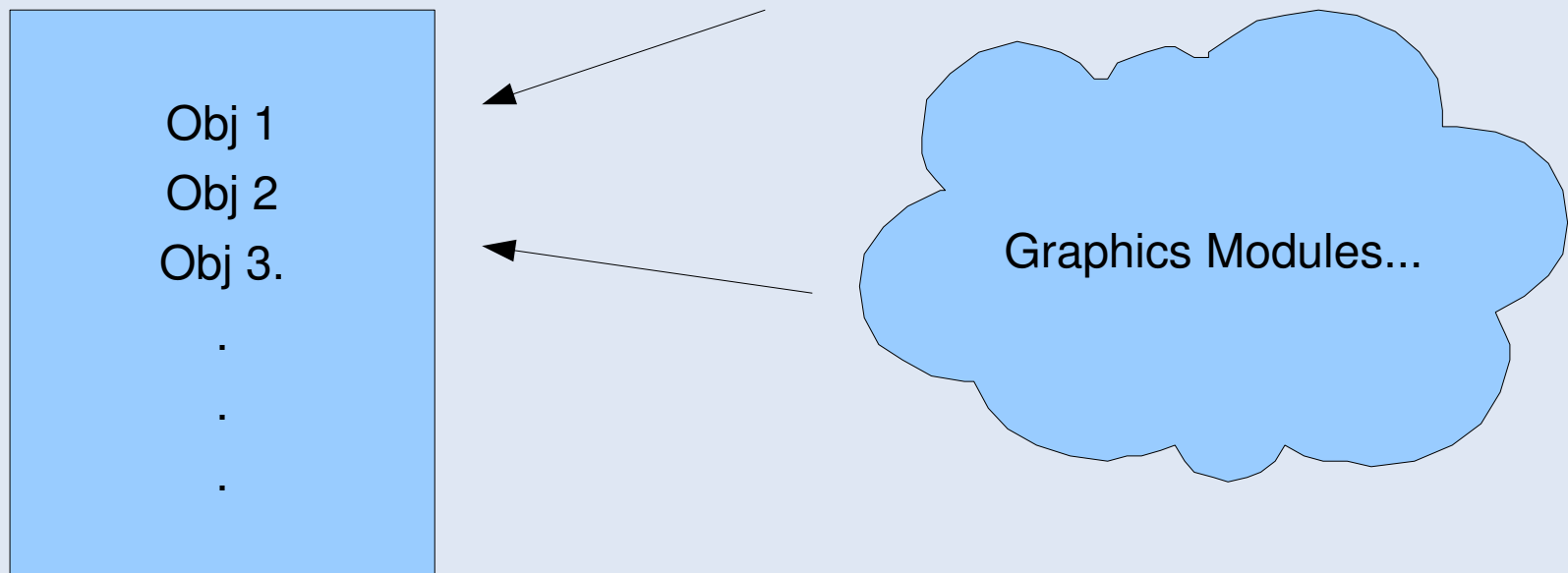


Graphics Main Module

- Driven by pixel clock
- Asks each object for a pixel and its depth
- Picks pixel based on depth comparison

Environment Lists

- Decouples Dependency on GLU
- Encodes Obj Information in Registers
- Generalizes The Objects on Screen



What is a an Object?

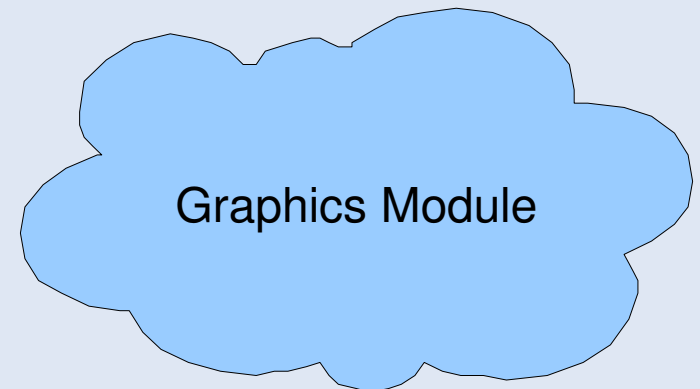
- Animated Objects – Goose, Dog
- Background Objs – Trees, Clouds
- Each is a module
- Will maintain state (unrelated to GLU)



rgb →

depth →

← (x,y)



Graphic Modularity



- Decoupling Leads to Freedom
 - Once Basic Infrastructure in place..
 - Expansions easy
- Keeping extra state in objects
 - More complex animations

Memory Usage

- My One True Fear

8 Shootable Objs + 8 Background Objs = 16 Objects

n Sprites per Object = 16n

Assume a max 200x200 pixels per obj:

40,000 pixels * 16n = 640,000*n pixels

640,000 * 3 bits = 234*n kilobytes

We have 4 MB ram...

4 MB / 234 kB = ~17 sprites / obj !!!

Development Time Line

- Four Milestones

Module	Game Logic <i>Peak</i>	Graphics <i>Pete</i>	Zapper/Sound <i>Fareed</i>
Milestone			
Monday, Nov 19th	Interface to Zapper Make one Goose Move	1 Sprite fully functional (Including fly/die)	Obtain Zapper from Hong Kong
Monday, Nov 26th	Write multiple objects to environment list	Read object from env list Finish Background	Determine pin out Output to GLU on trigger
Monday, Dec 3rd	Test integration with Graphics and Zapper	Full integration w/ Game Logic	Zapper calibration for LCD Position output to GL
Monday, Dec 10th	Testing/debug completed Implement extra user state	Testing complete More advanced graphics	Work on sound (if time allows)

Acknowledgements

- Owlo.com – Green Feathers Image
- Nintendo – DuckHunt Images