

# DuckFeed

An Embedded Take on The Classic  
Video Game "Duck Hunt"

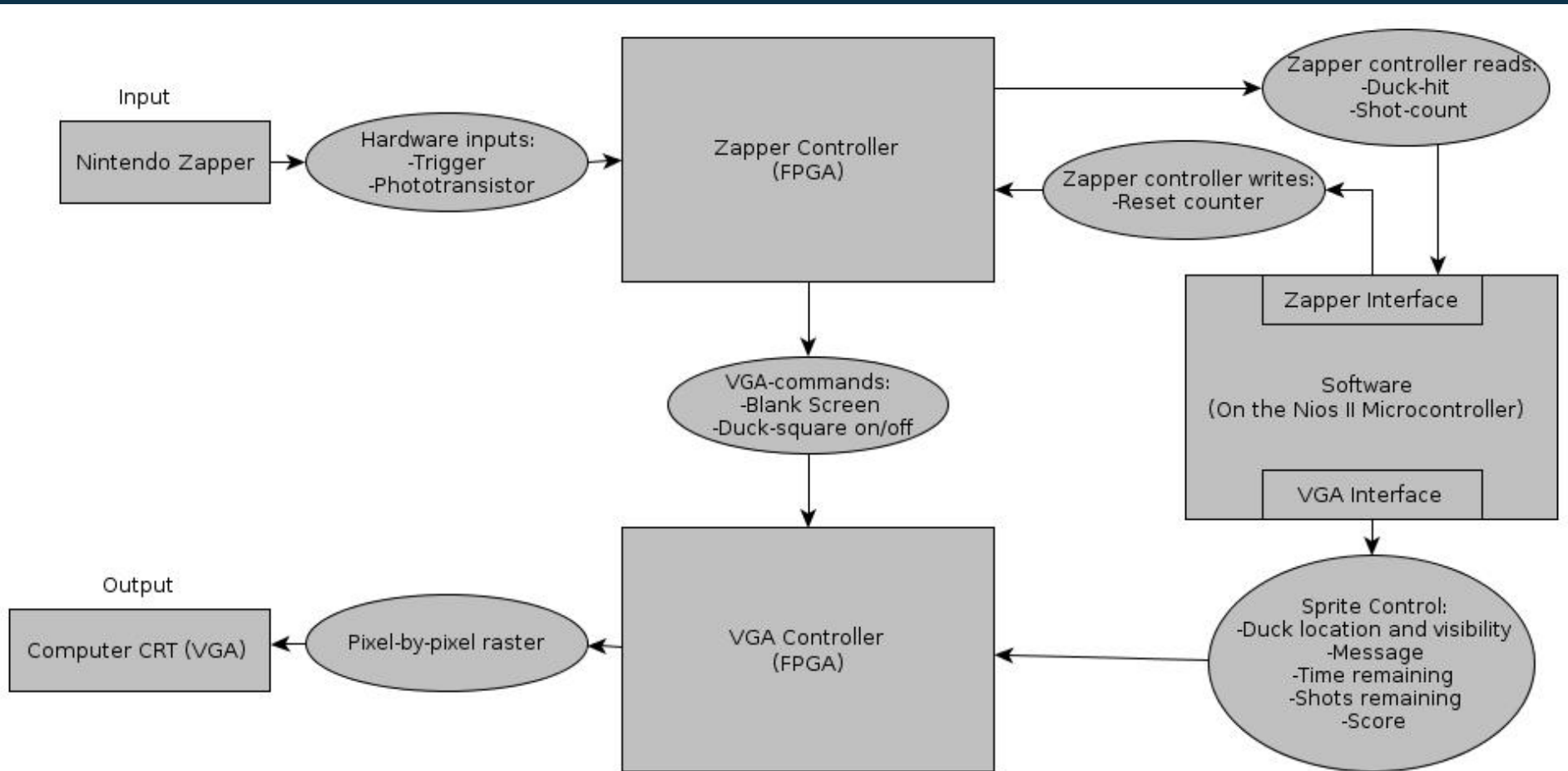
Daniel Teger (EE)

Julie Dinerman (EE)

Kevin Ramkishun (EE)

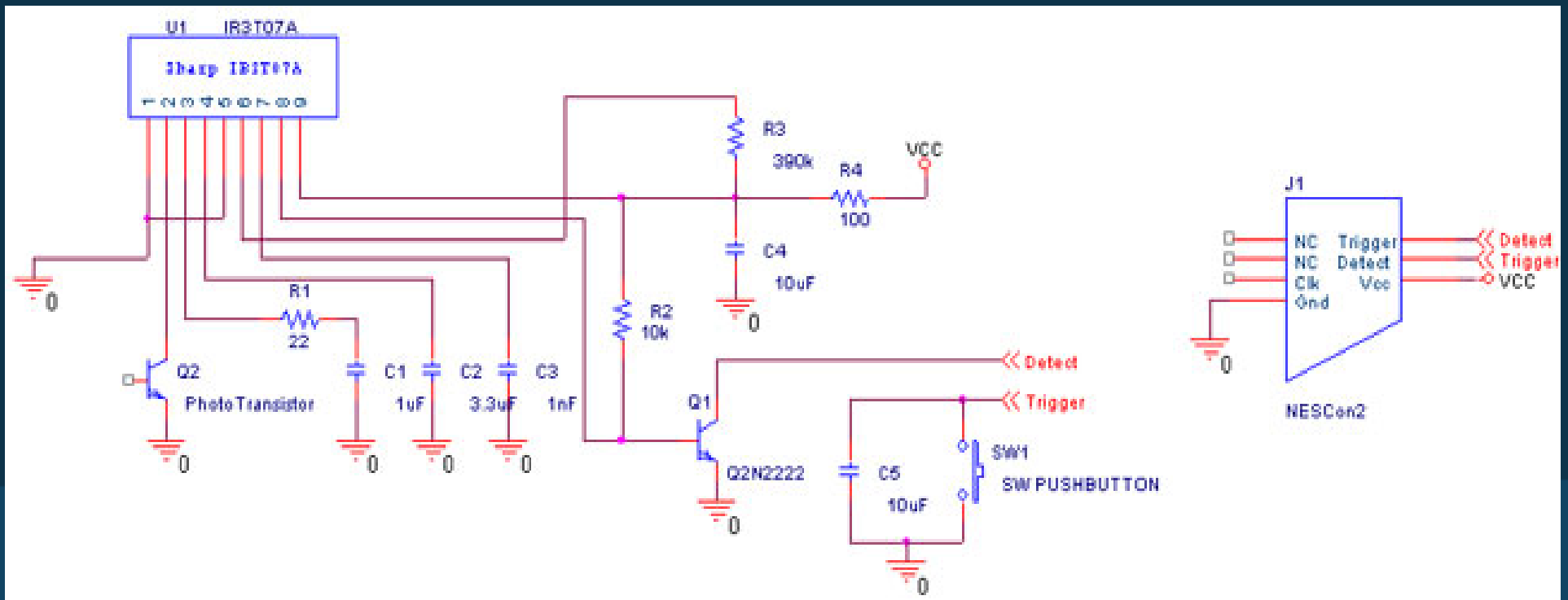
Scott Rogowski (CS)

# Intro Slide - Kevin

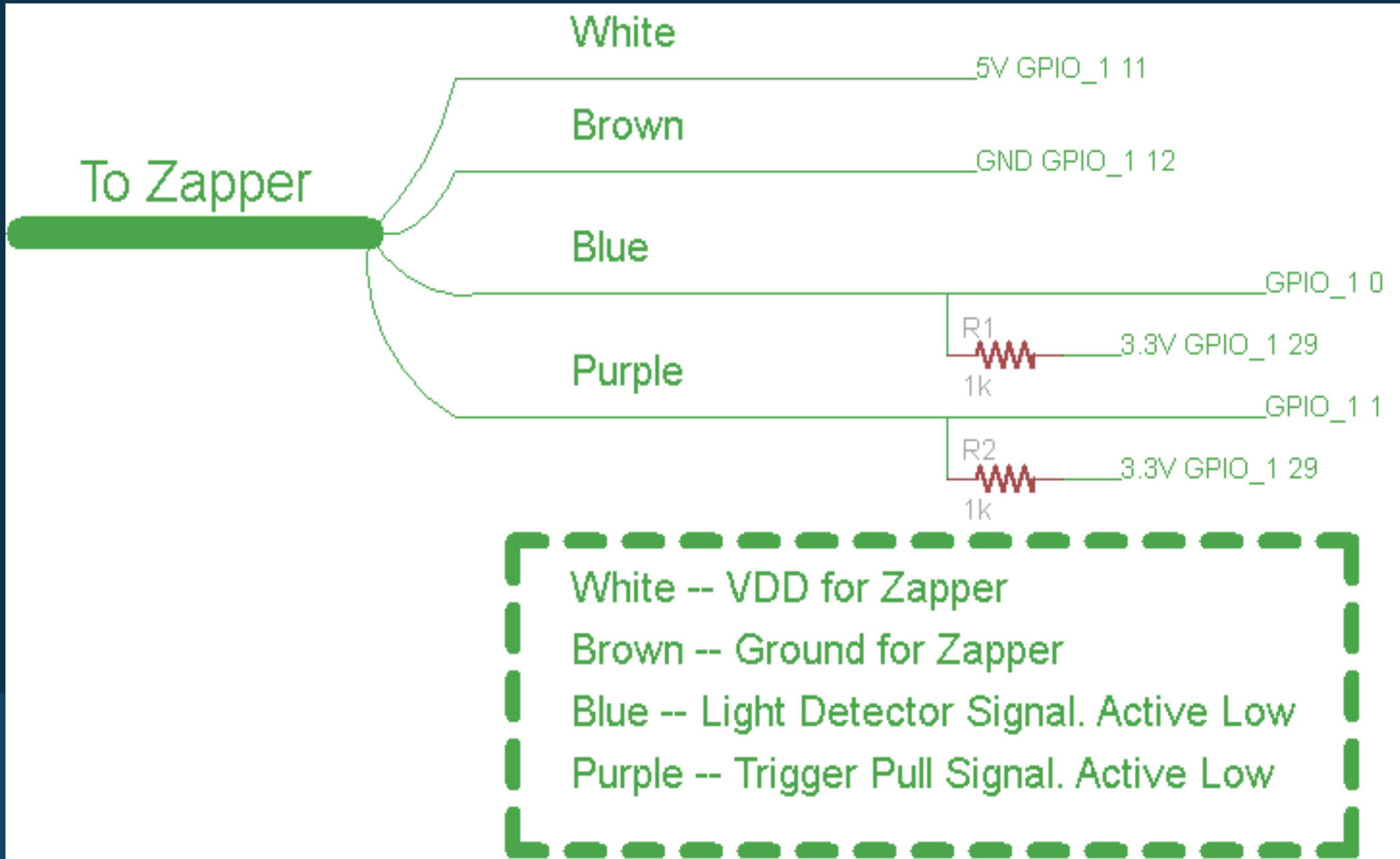


# Zapper Schematic

- Open Collector Outputs
  - Easy to pull up to 3.3V (board voltage)
- Band pass filter
  - Place 390k resistor in parallel with existing one



# Hardware Interfacing

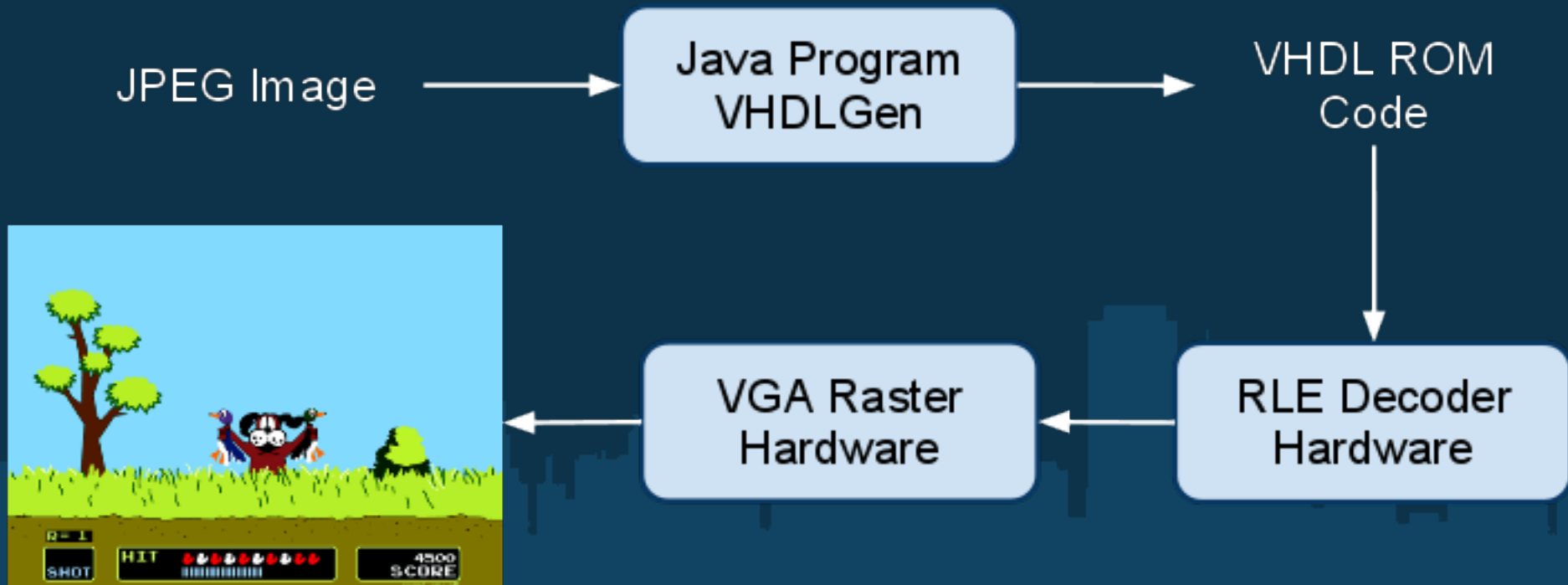


# Timing (screen blanking) Slide

State	Task	Time
1	Wait for trigger pull	Indefinite
2	Wait for trigger release	Indefinite
3	Set timer to debounce trigger release	200,000 cycles
4	Wait for raster scan to reach end of field	< 416,000 cycles
5	Turn screen black for one frame. Ensure there is no light signal from the gun.	332,500 cycles
6	Keep screen black but place white targets in the place of ducks in binary search pattern. Look for light signals.	450,000 cycles
7	Return screen to background and flying ducks	After 5 flashes

# Background Encoding Saves Memory!

- Background is 640 x 480 pixels, each pixel is 24 bits (rgb)
  - Almost 1MB! Definitely won't fit on chip
- Solution: Run Length Encoding with 16 Colors
  - Multiplex colors into 4 bits, length encoded in 8 bits
  - Fits in less than 10% of the memory
  - Pain to implement



# Sprites



- Sprites are made out of pixel arrays stored in a rom
  - ducks need 5 colors:  
type duck\_type is array (0 to 1599) of unsigned(3 downto 0);
  - score numbers and duck food need only 1 color:  
type score\_type is array (0 to 1599) of std\_logic;
  - type fish\_type is array (0 to 399) of std\_logic;
  - each sprite can be read simultaneously up to 2 times-  
need multiple constants
- Pixels are read from the rom and colors are decoded in processes in the VGA raster- 1 transparent color
- Then RGB pixel values are called in the VideoOut method which draws the sprites

# Sprites



## Array Indexing:

- Use horizontal and vertical counters from the raster and duck position from software to calculate the index in the sprite arrays
- Ducks display differently depending on how the sprites are indexed, for example right and left facing ducks.

Flapping: alternate between up-winged duck image and down-winged duck image at every duck flap.





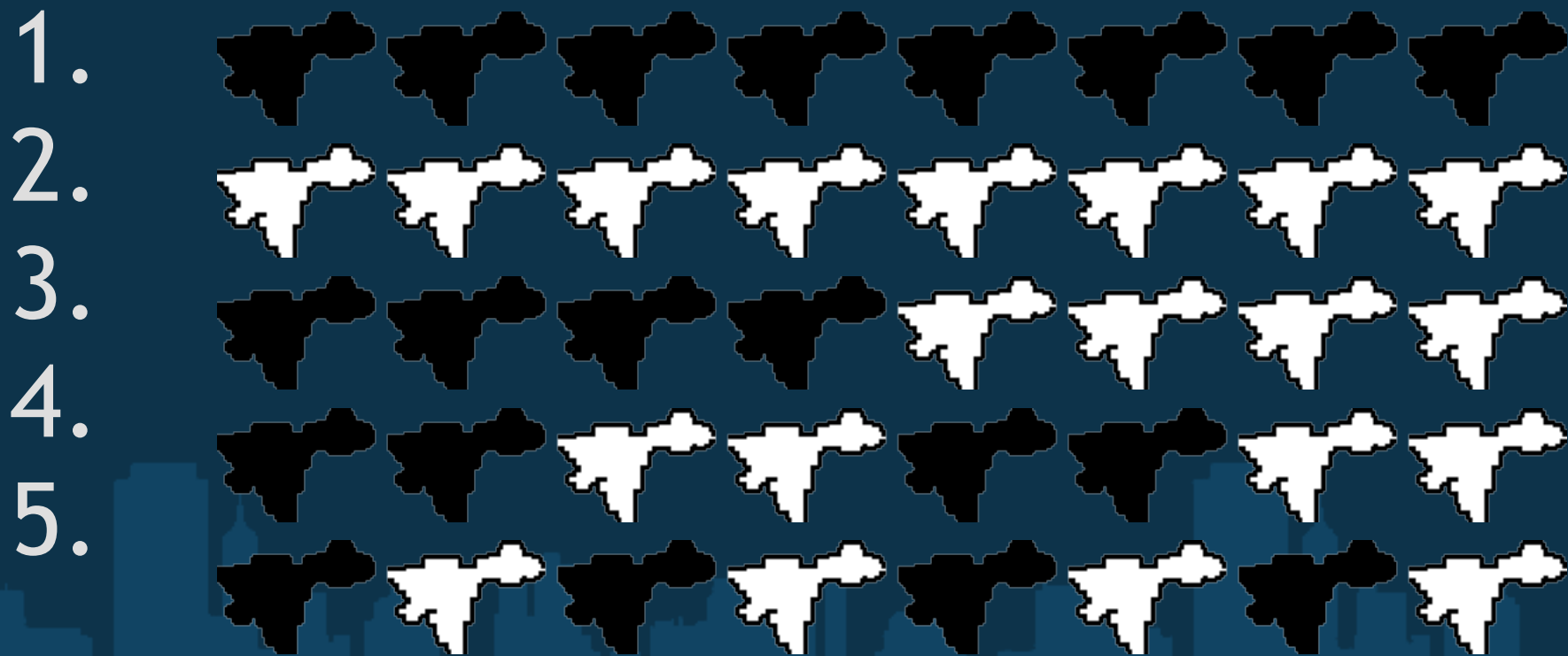
# Software

- The cerebral cortex of the operation
  - Defines level variables (number of ducks, speed, time limit, etcetera)
- Consistently loops and:
  - Controls duck position, duck visibility, etc.
  - Keeps track of time and shots remaining
  - Decides when a level has been won or lost

# Finding the Right Duck - Scott

Actually using Data Structures knowledge!

Implemented a binary search algorithm which will determine which duck was hit, checking for cheaters, in five cycles.



# Lessons Learned

- The best advice we got this semester:  
Fix the small problems before tackling the big problems!
  - Fixing a bug in the VGA raster helped us get the background working
- Understand the hardware
  - Dual-Ported ROMs made our life easier

Questions?