

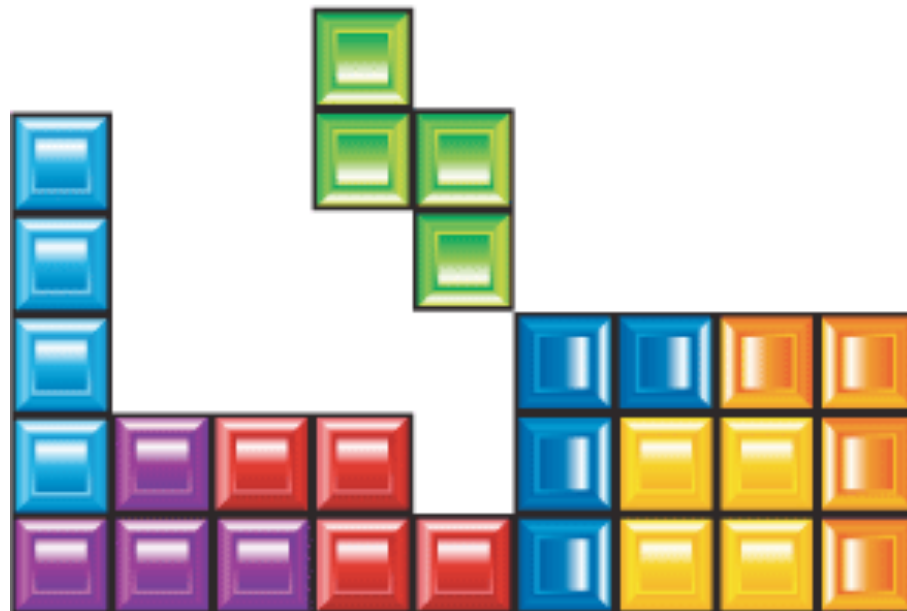
# ImprovTetris

Scott Bezek  
Ray Li

11/15/11

# Classic Tetris

- Piece together blocks
- 7 possible blocks, picked randomly



# Classic Tetris

- Clear complete rows

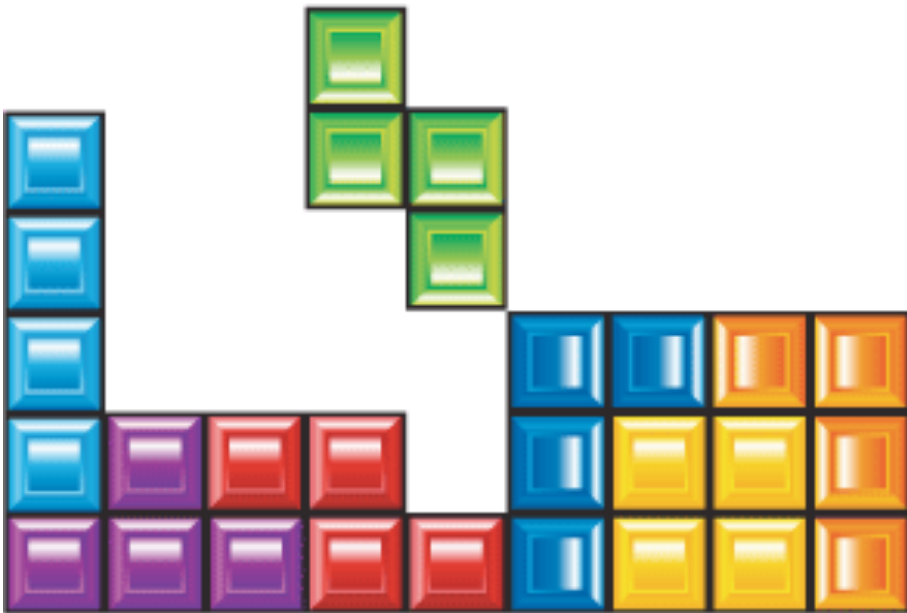


# Classic Tetris

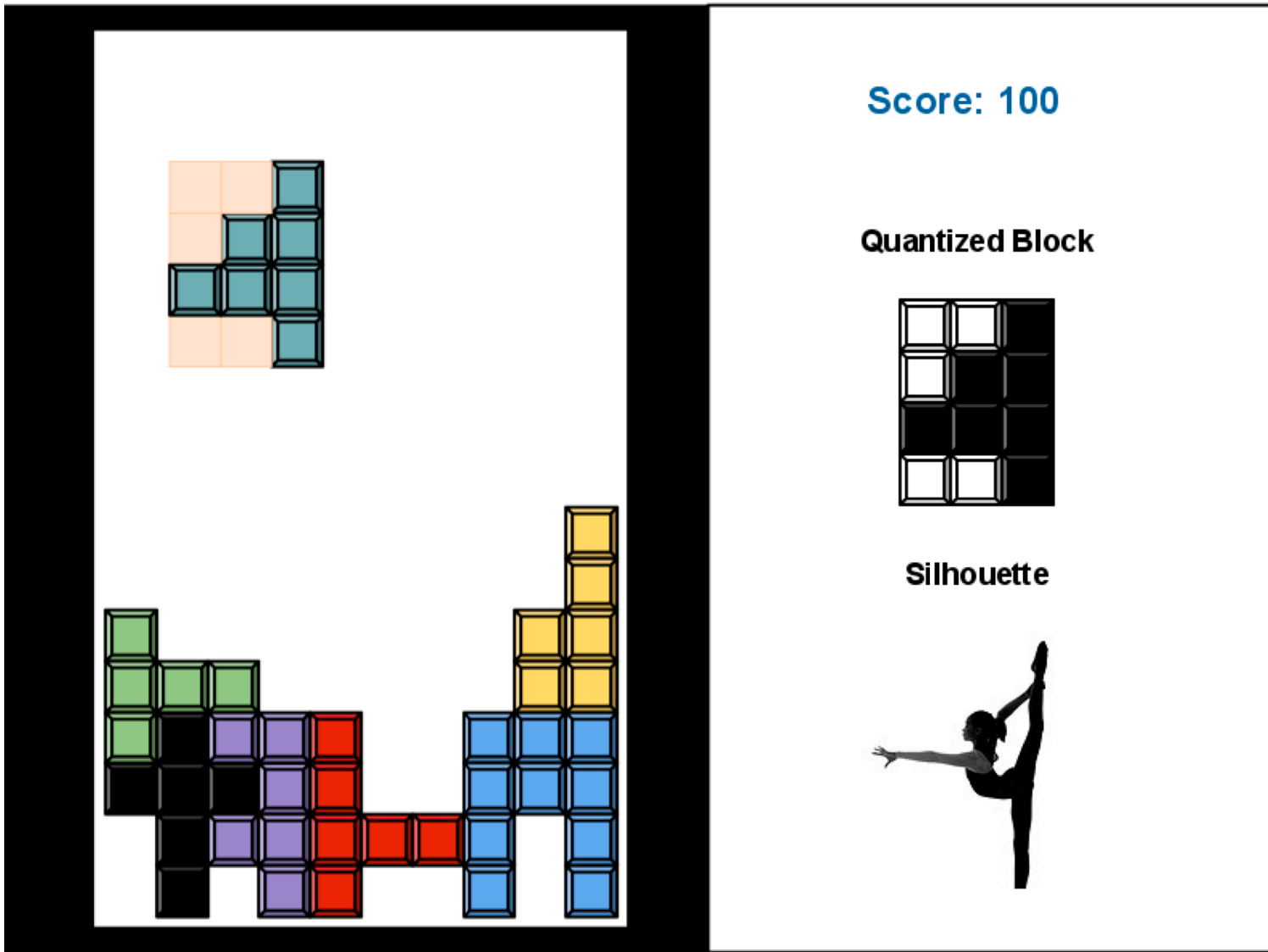
- Clear complete rows



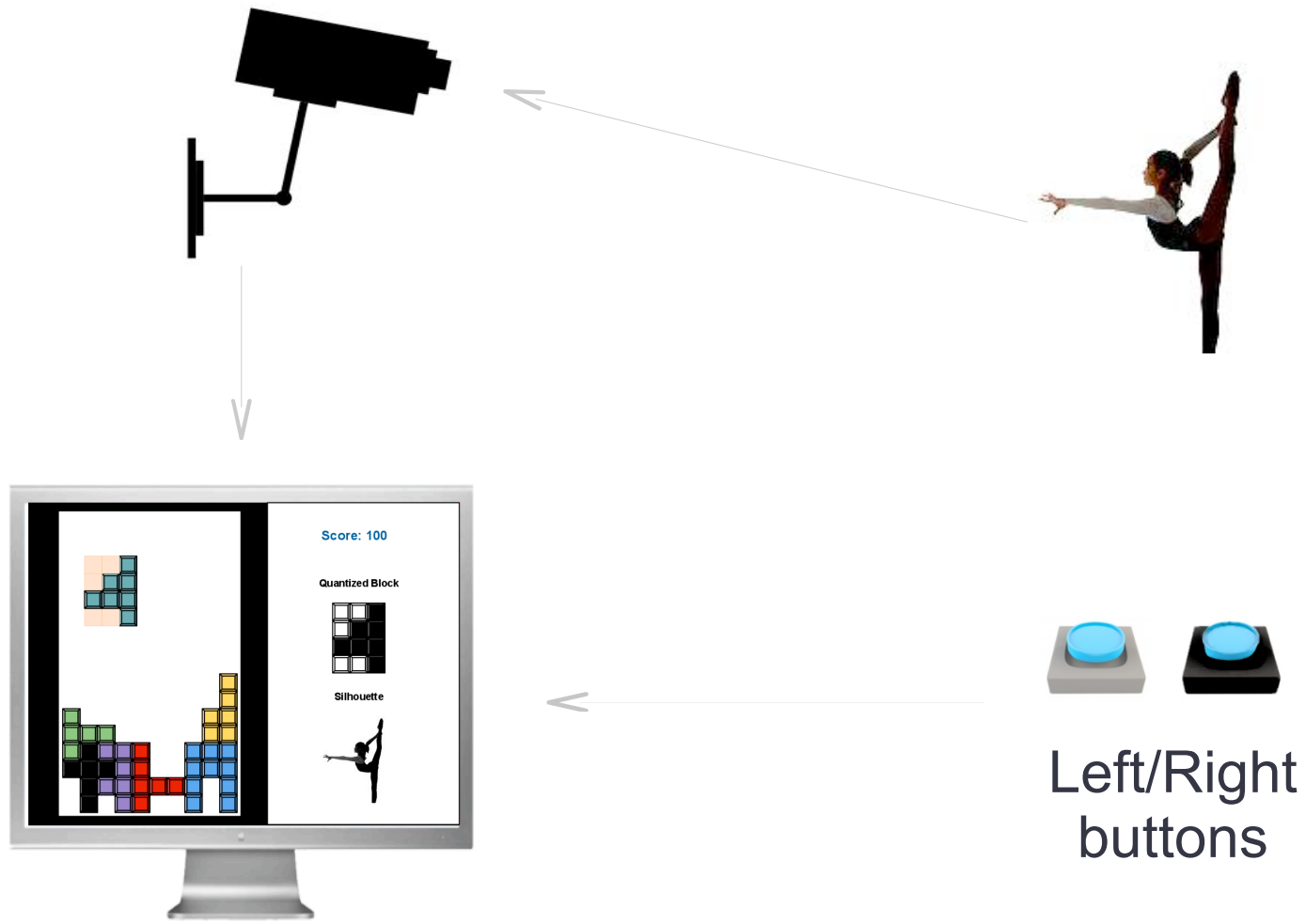
# Classic Tetris



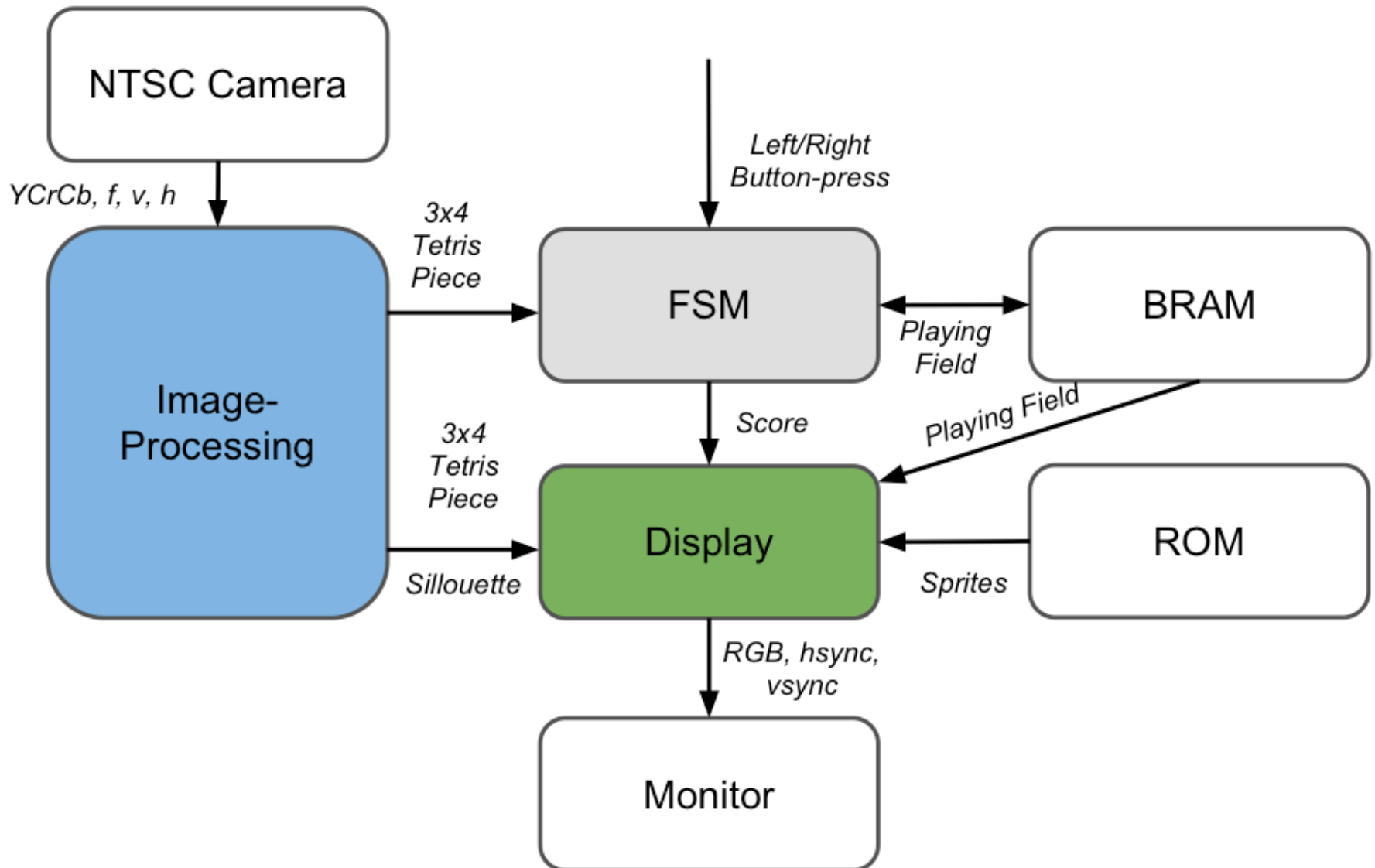
# ImprovTetris



# I/O



# Block Diagram





# Image Processing

Current Frame



Reference Frame



VS

# Image Processing



Subtract



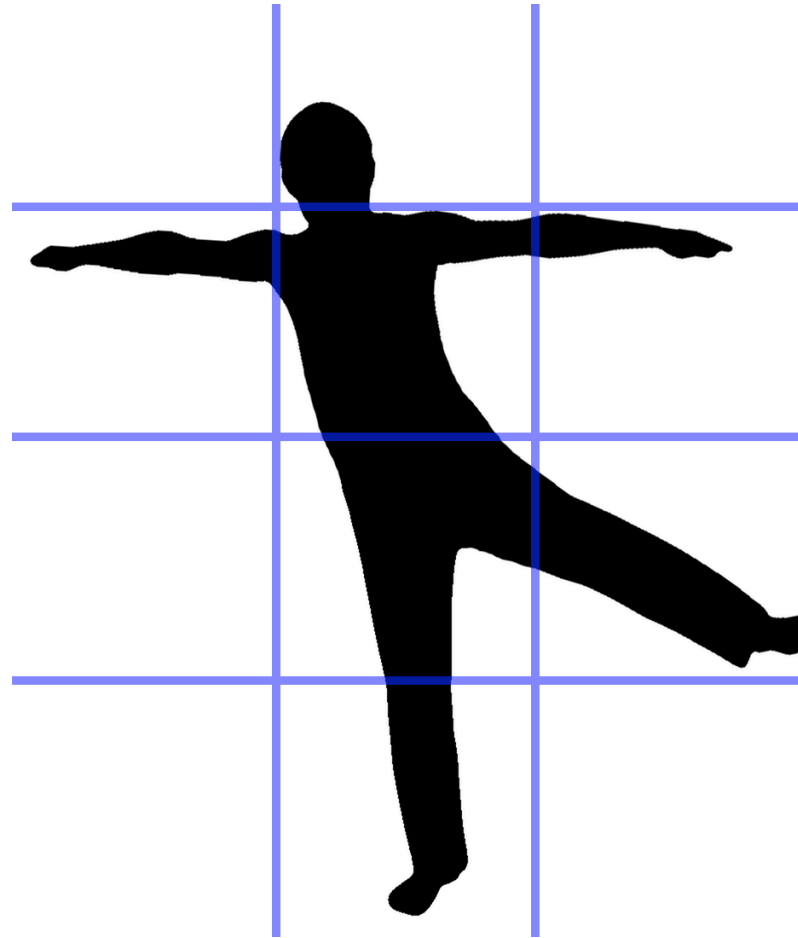
Silhouette

# Image Processing



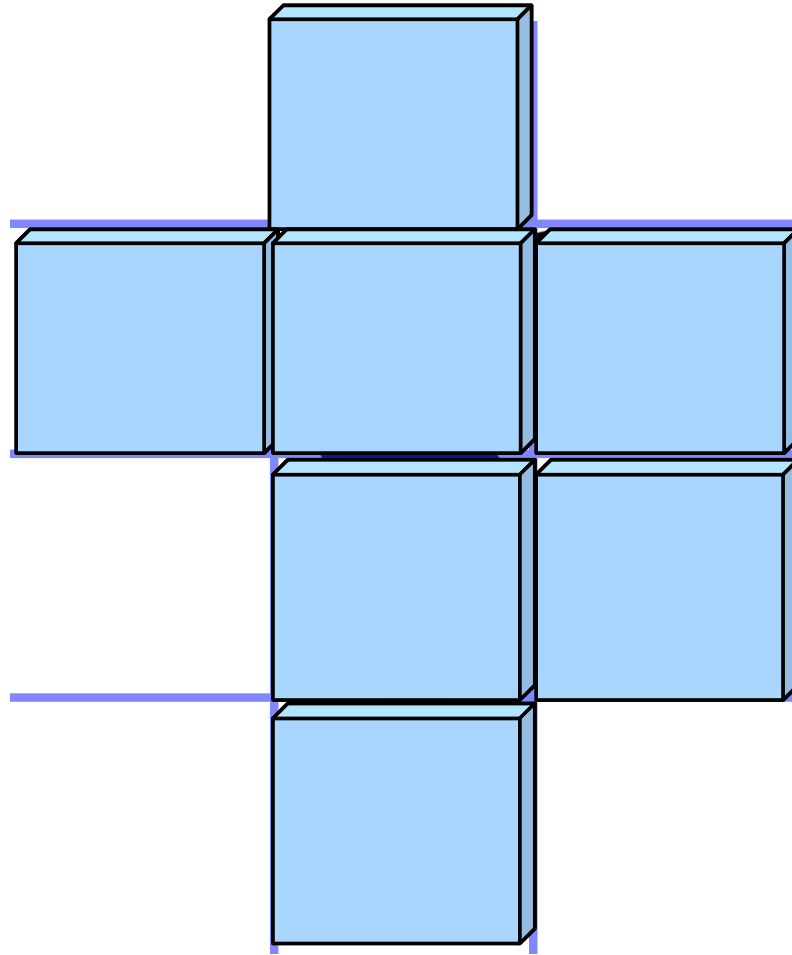
Silhouette

# Image Processing



Silhouette

# Image Processing

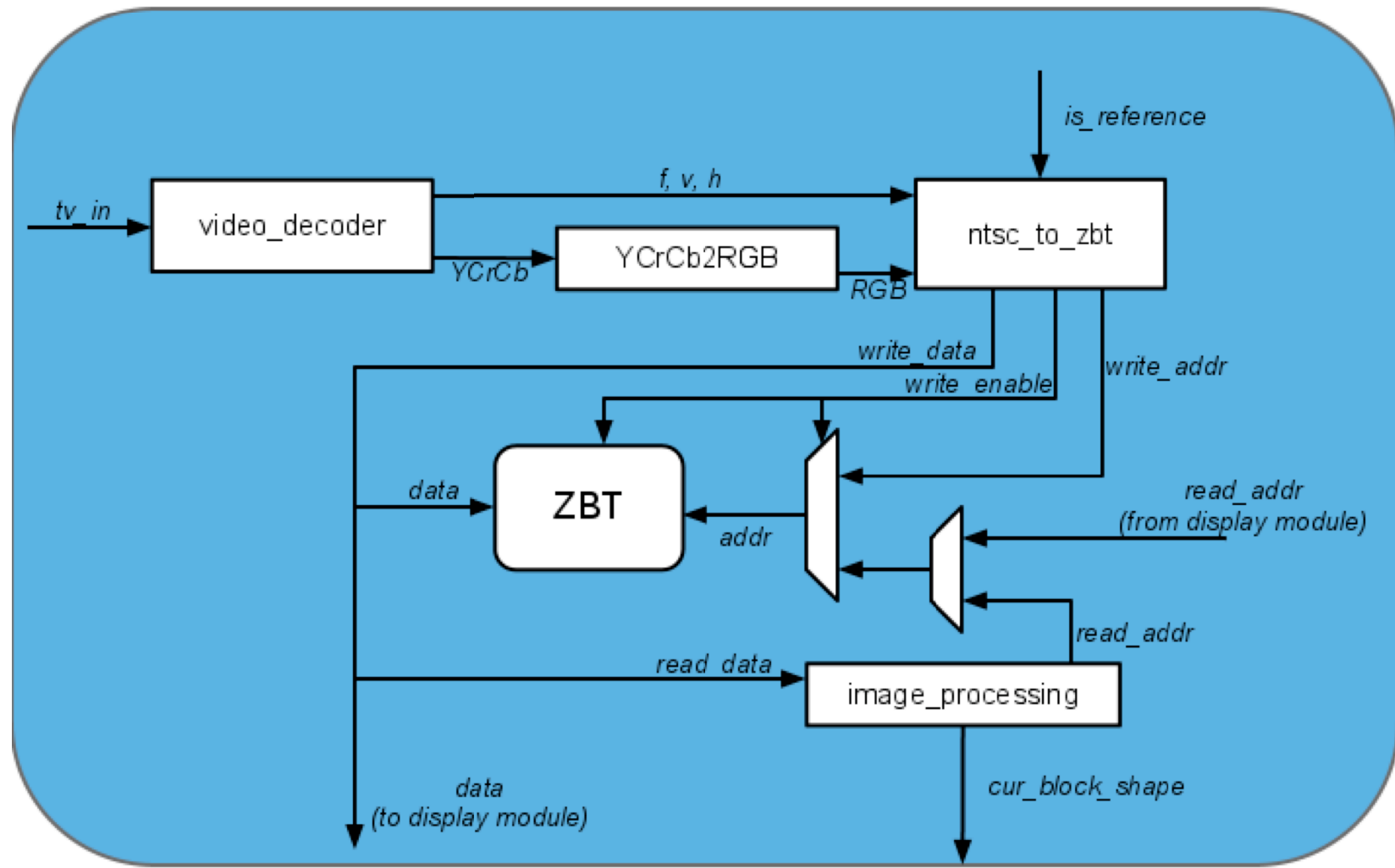


Quantized Block

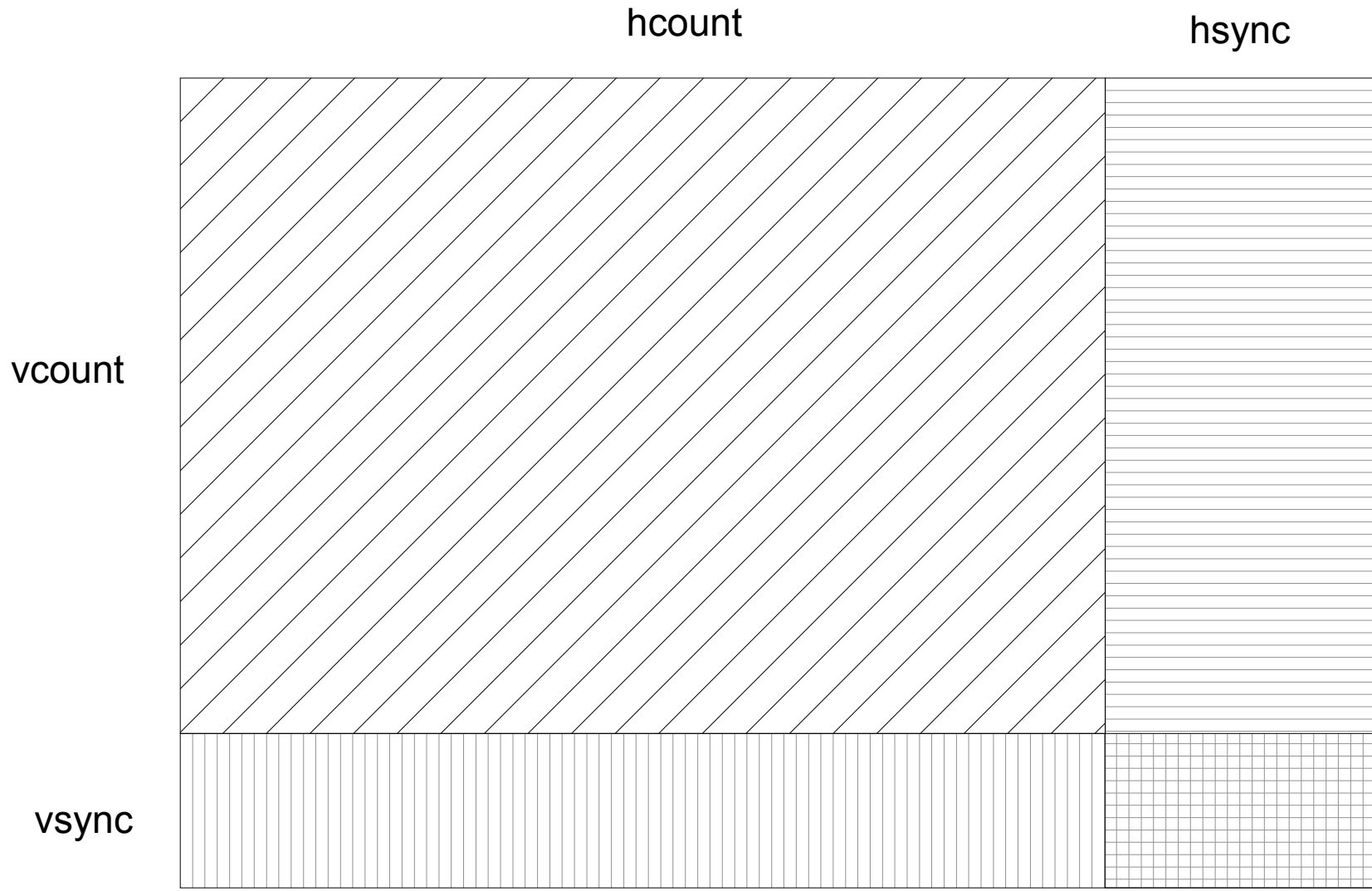
# Image Processing

- Pixel-by-pixel subtraction (2 px per clock cycle)
- 4x4 pixel “blocks” - increase SNR

# Image Processing

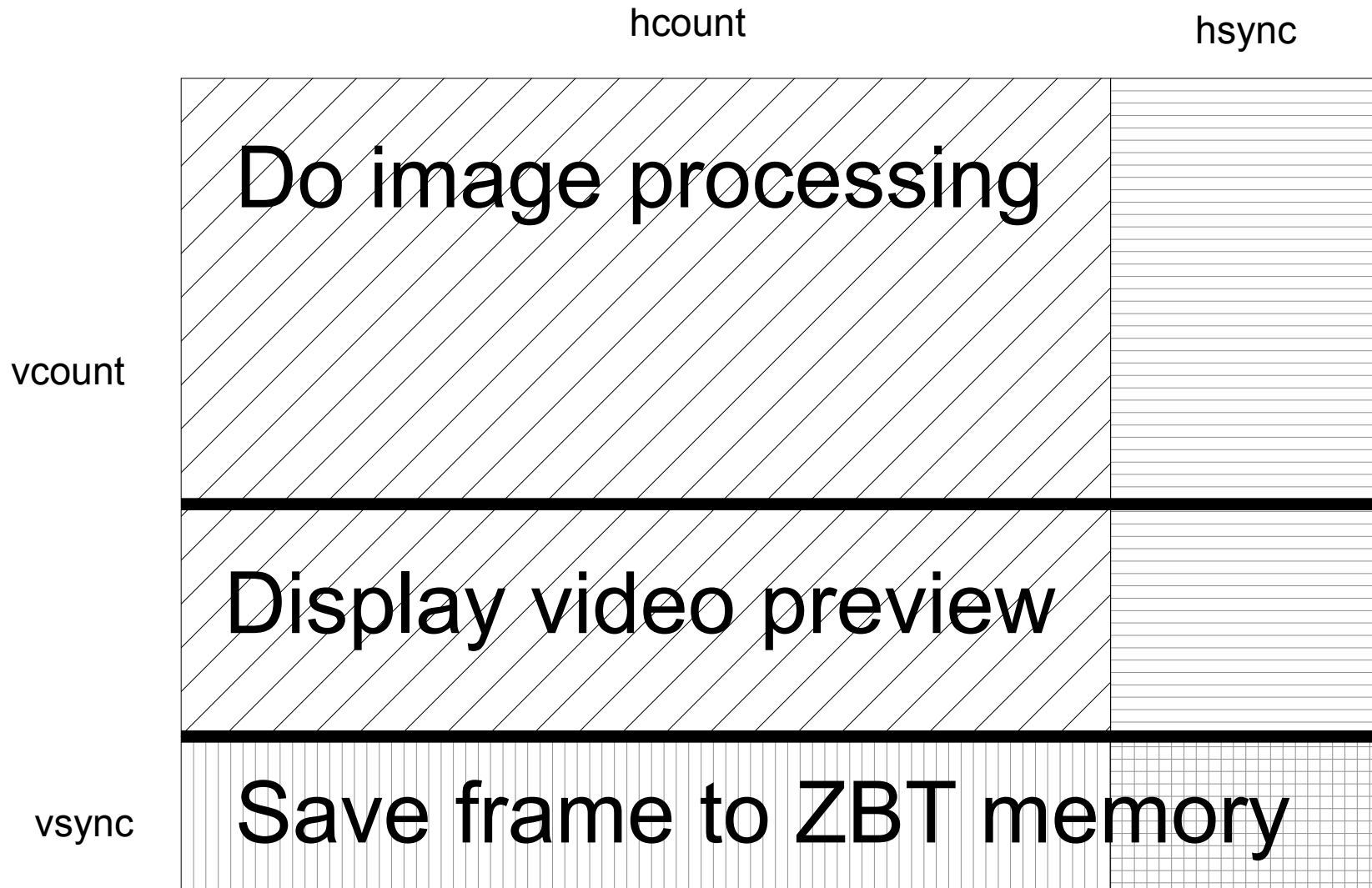


# Image Processing

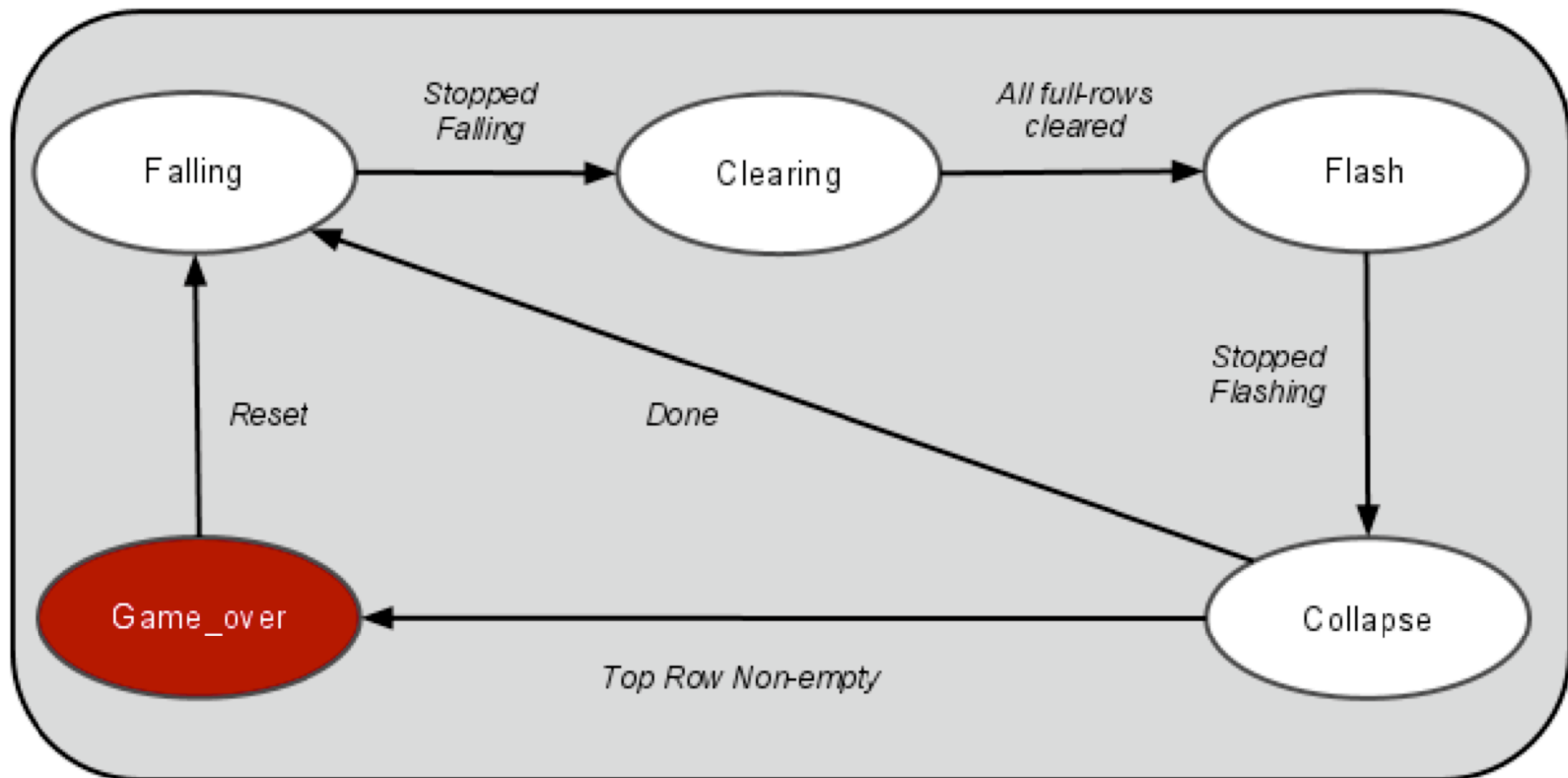




# Image Processing



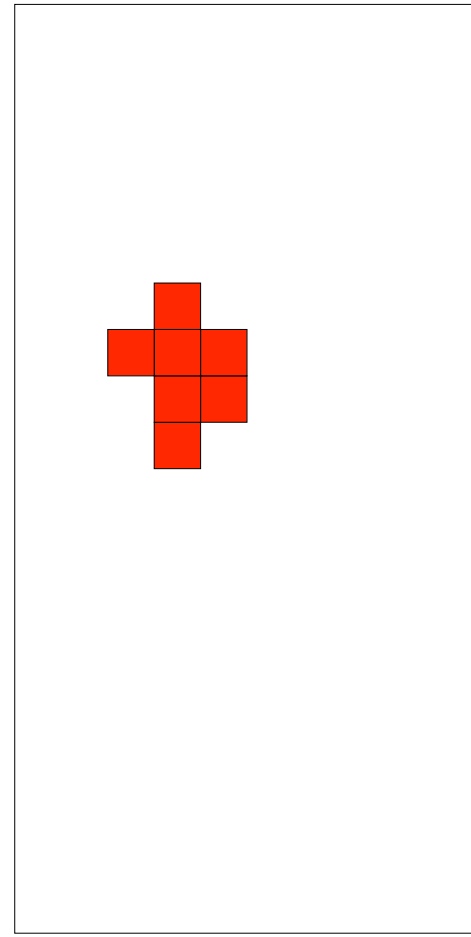
# FSM (Game Logic)



# Game State

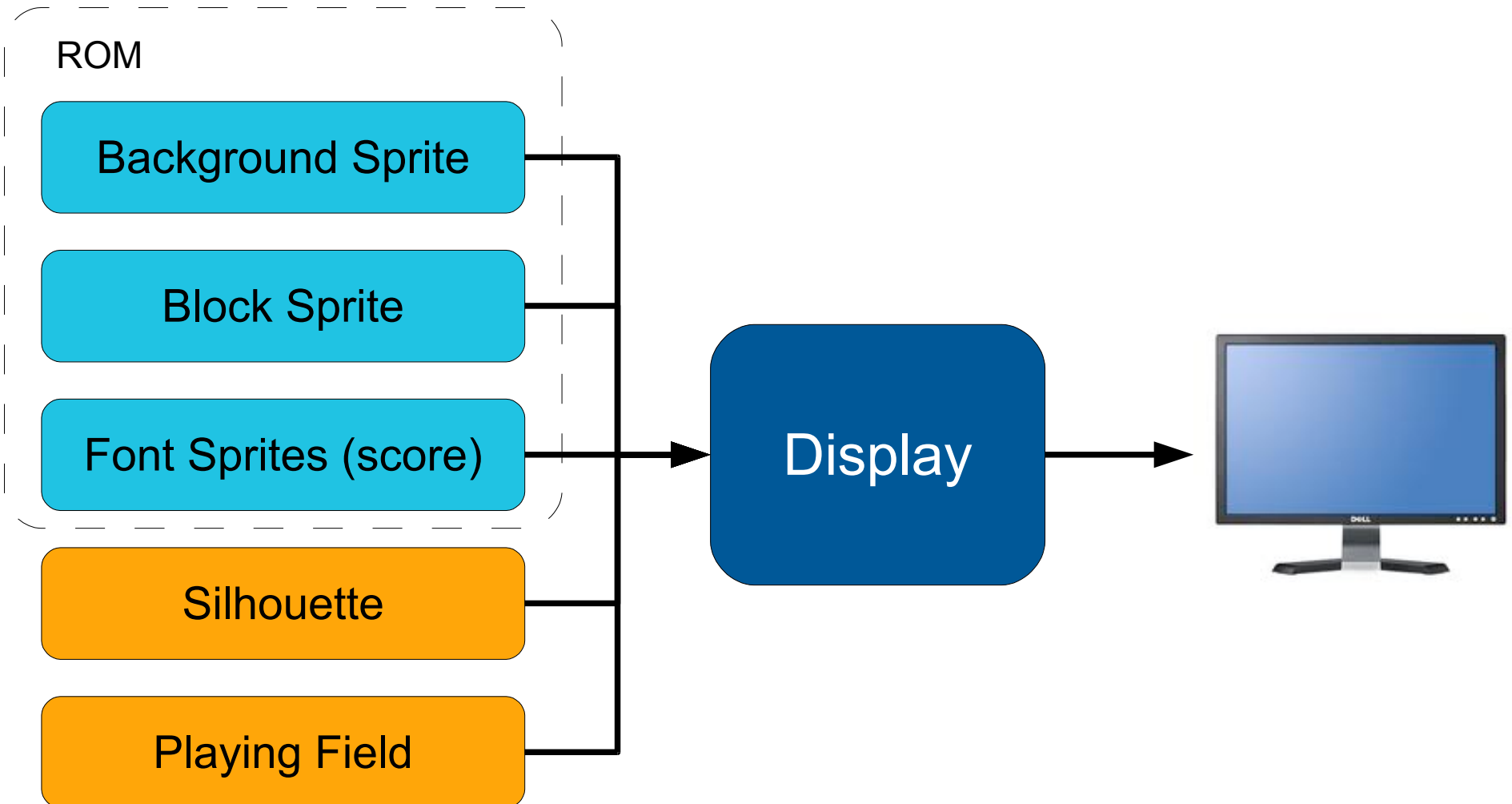


Settled



Falling

# Display Module



# Major Challenges

- Timing constraints
  - Pixel clock speed – ROM sprite lookup
  - Image processing takes many cycles
  - Memory address forecasting
- Noise from NTSC camera
- Shared memory access
  - ZBT – camera write vs. processing read
  - BRAM – FSM write vs. display read
  - Only one module can read/write at a time
  - Must arbitrate access

# Timeline

---

<b>Week</b>	<b>Ray's goals</b>	<b>Scott's goals</b>
November 14	Implement FSM	Implement Silhouette Image Processing
November 21	Finish FSM. Test FSM and begin Display Module	Implement Block Quantization
November 28	Finish Display	Test and fine-tune Image-Processing Algorithms
December 5	Debug [Add audio effects] [Improve graphics]	Debug [Add Gyroscope] [Add two-player support]

---