

DCLab Final Duck Hunt

Teamo₄ 解正平 韓秉勳 蔡昕宇

Outline

- Introduction
- Hardware Device
- Module Architecture
 - Camera Detection
 - Game Interface
- Problem and Discussion
- Future Work
- Demo Time

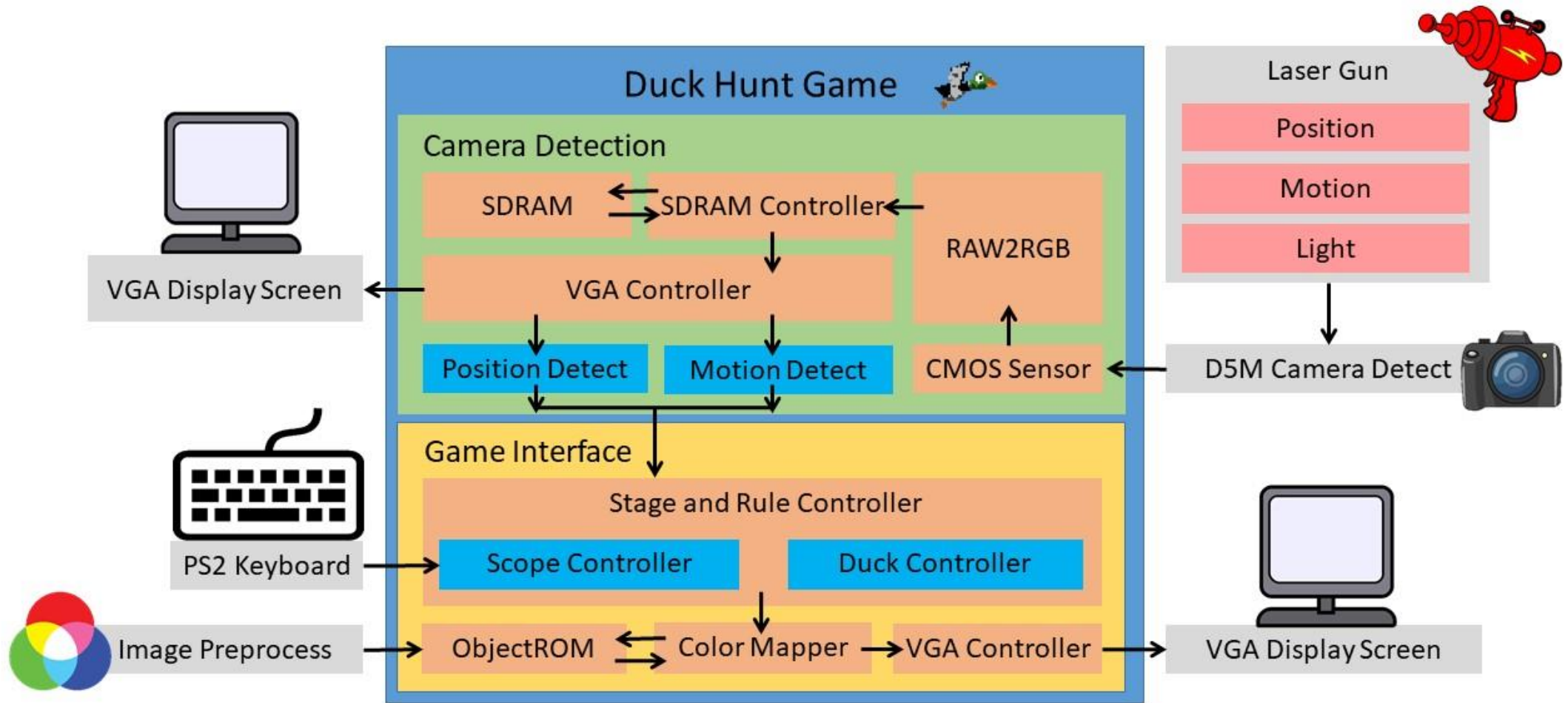


Introduction

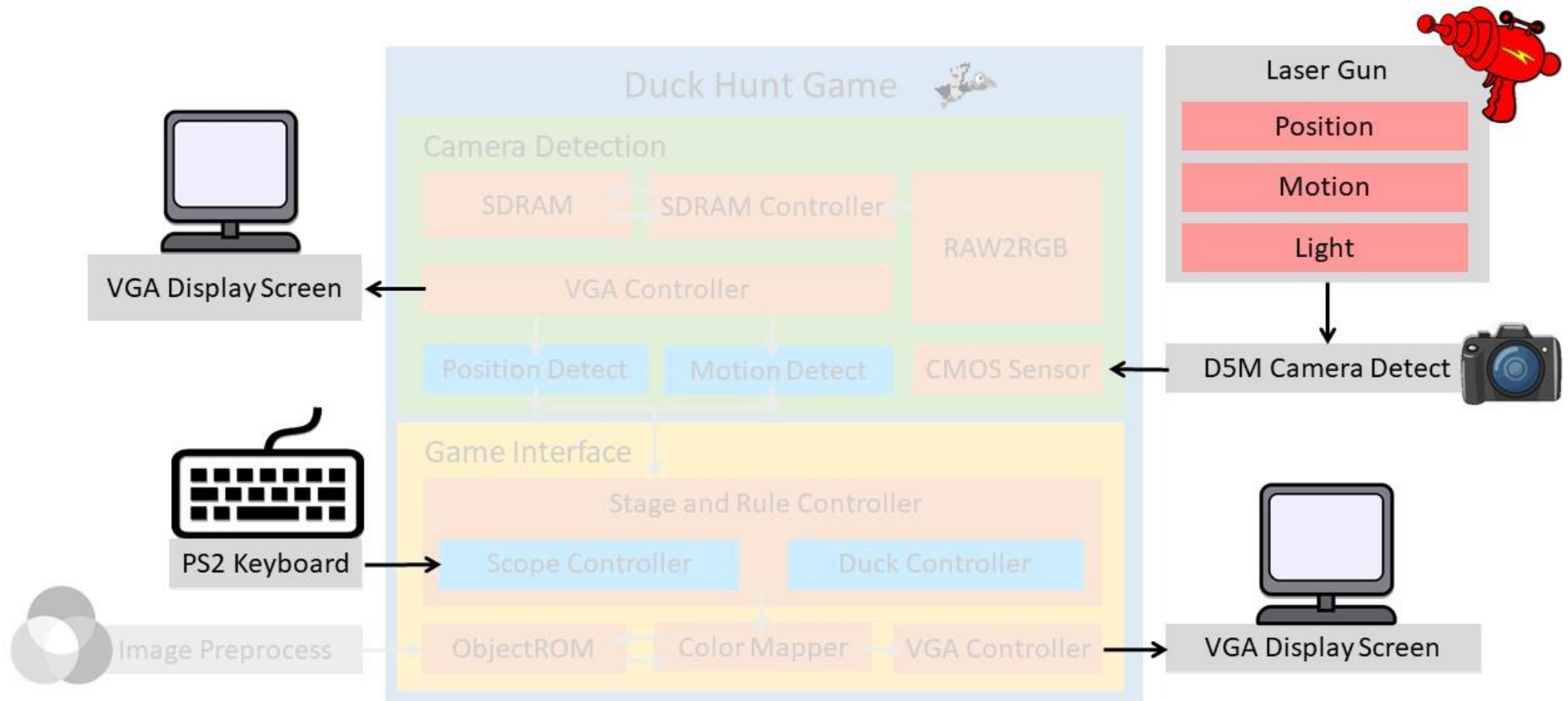
- 1984 Nintendo light gun shooter video game
- Design a shooting game in reality
 - Learn VGA display
 - Learn D5M camera
 - Think how to implement shooting gun
- PLAY GAME !!!



Module Architecture

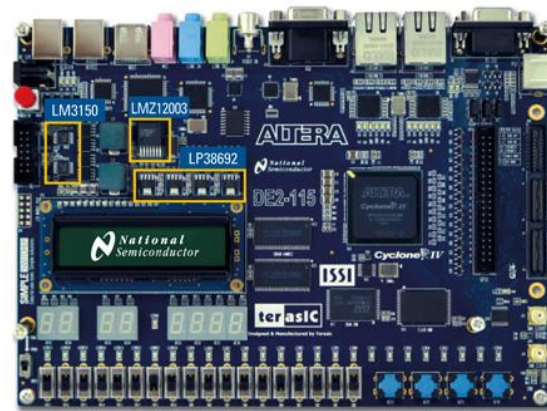


Hardware Device



Hardware Device

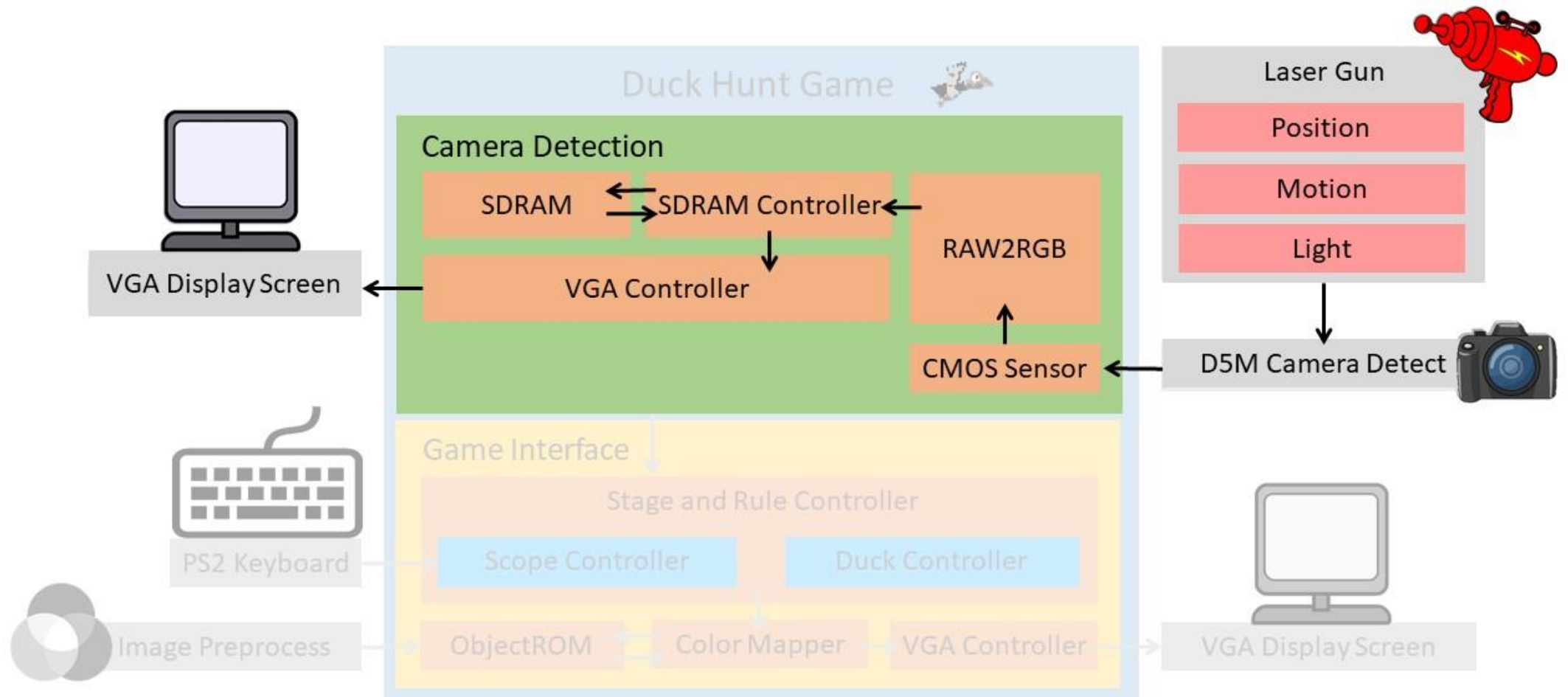
- DE2 115 FPGA Board
- D5M Camera
- VGA Screen
- Laser Gun





Camera Detection

Camera Module



D5M Camera

- CMOS sensor capture frames
- SDRAM store frames
- Real-time display videos
- Adjust the clock rate to change camera exposure



RGB₂YUV

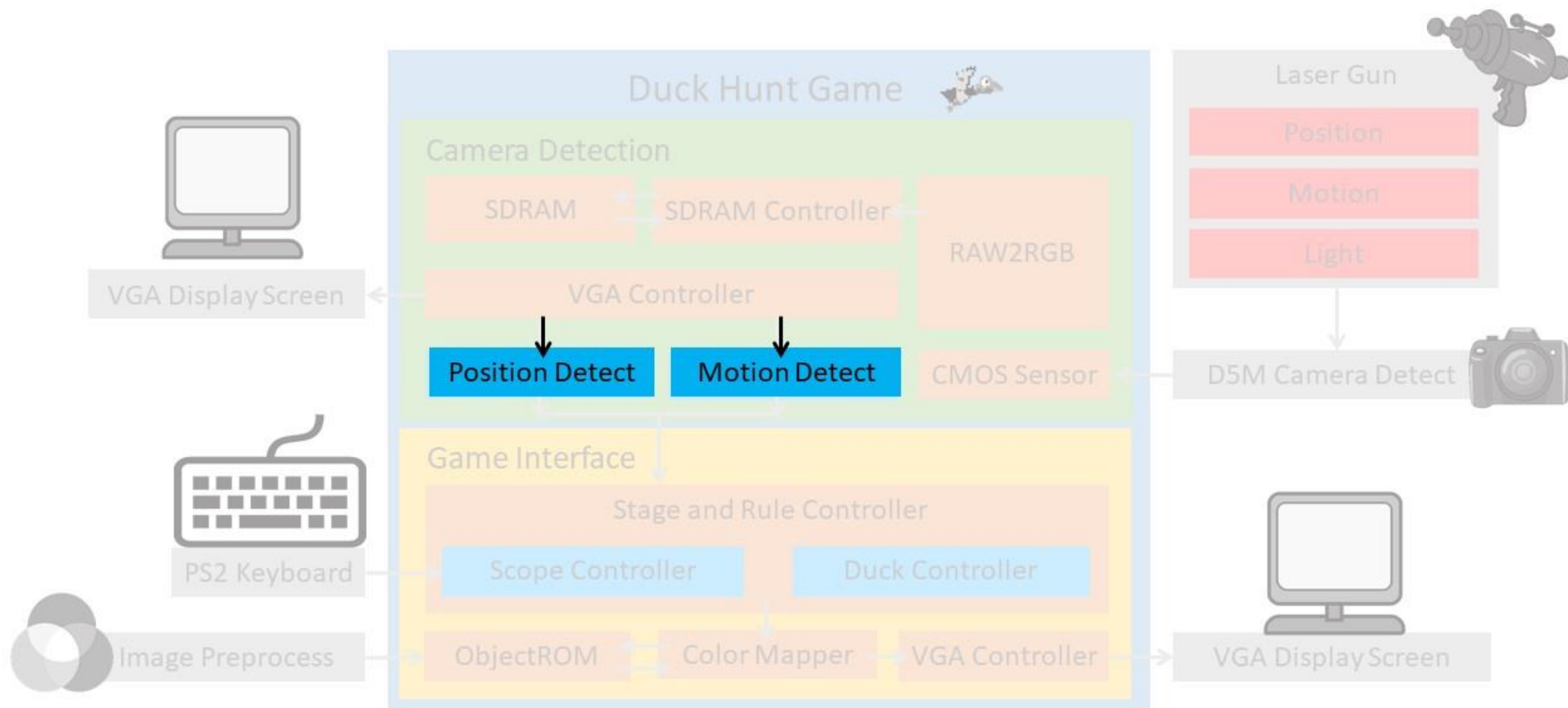
- Only capture the Laser Gun light
- RGB (color) to YUV(gray)
- Display $Y \geq 255$ pixel

$$Y = 0.299 \times R + 0.585 \times G + 0.114 \times B$$
$$Y = (77 \times R + 149 \times G + 29 \times B) \div 255$$

if $Y \geq 255$ this pixel is the gun light

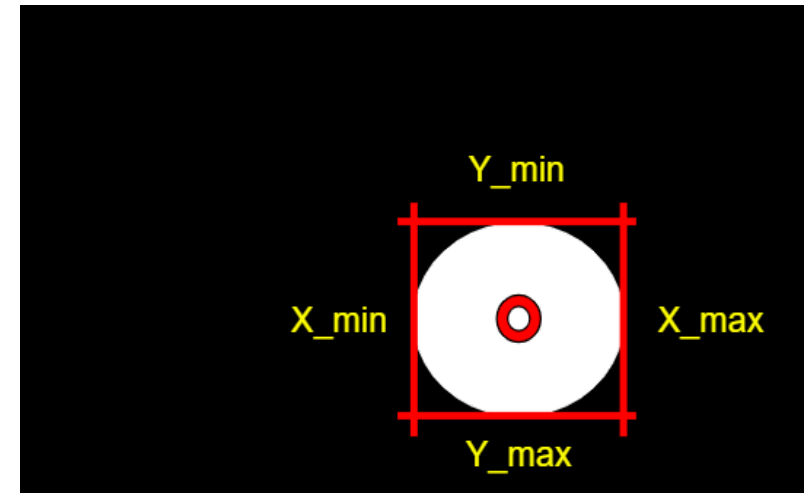
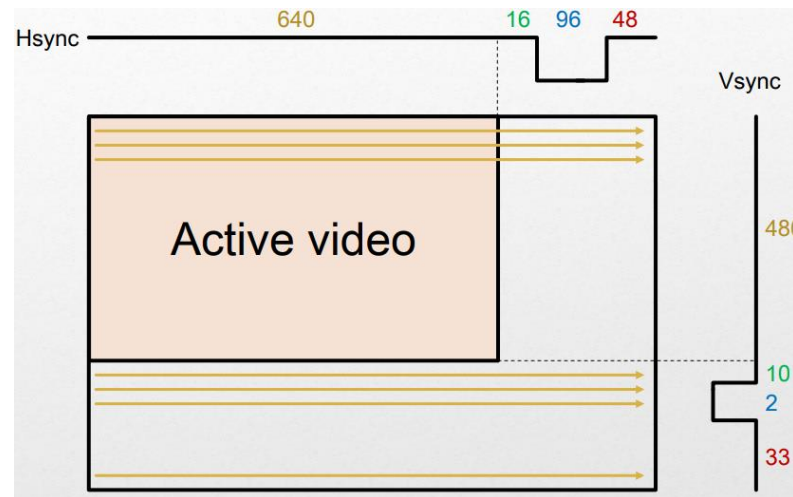


Light Detection



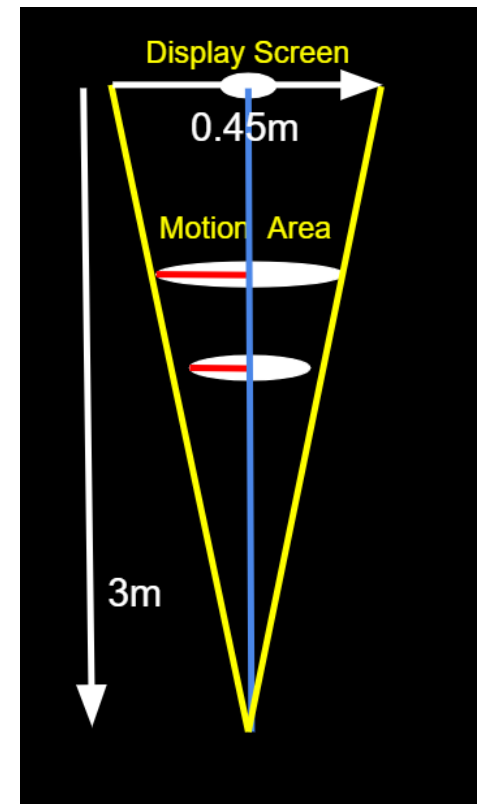
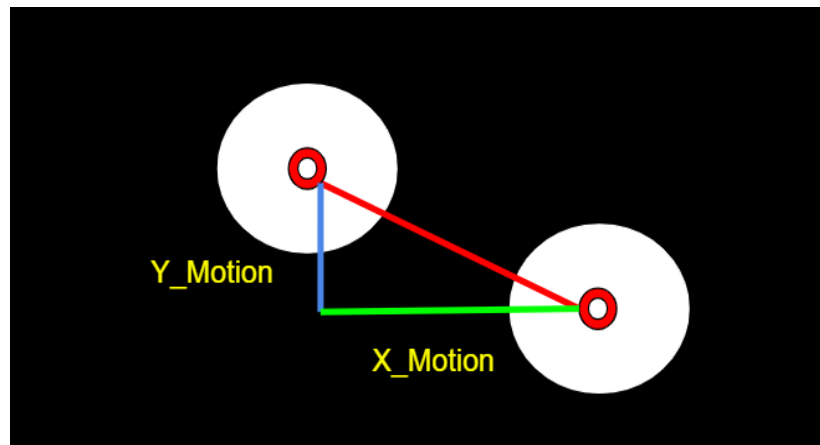
Position Detection

- Catch X_min, X_max, Y_min, Y_max and get **Center Position**
- H_count and V_count in 640x480
- Use ready signal to show read one frame
- Noise interference

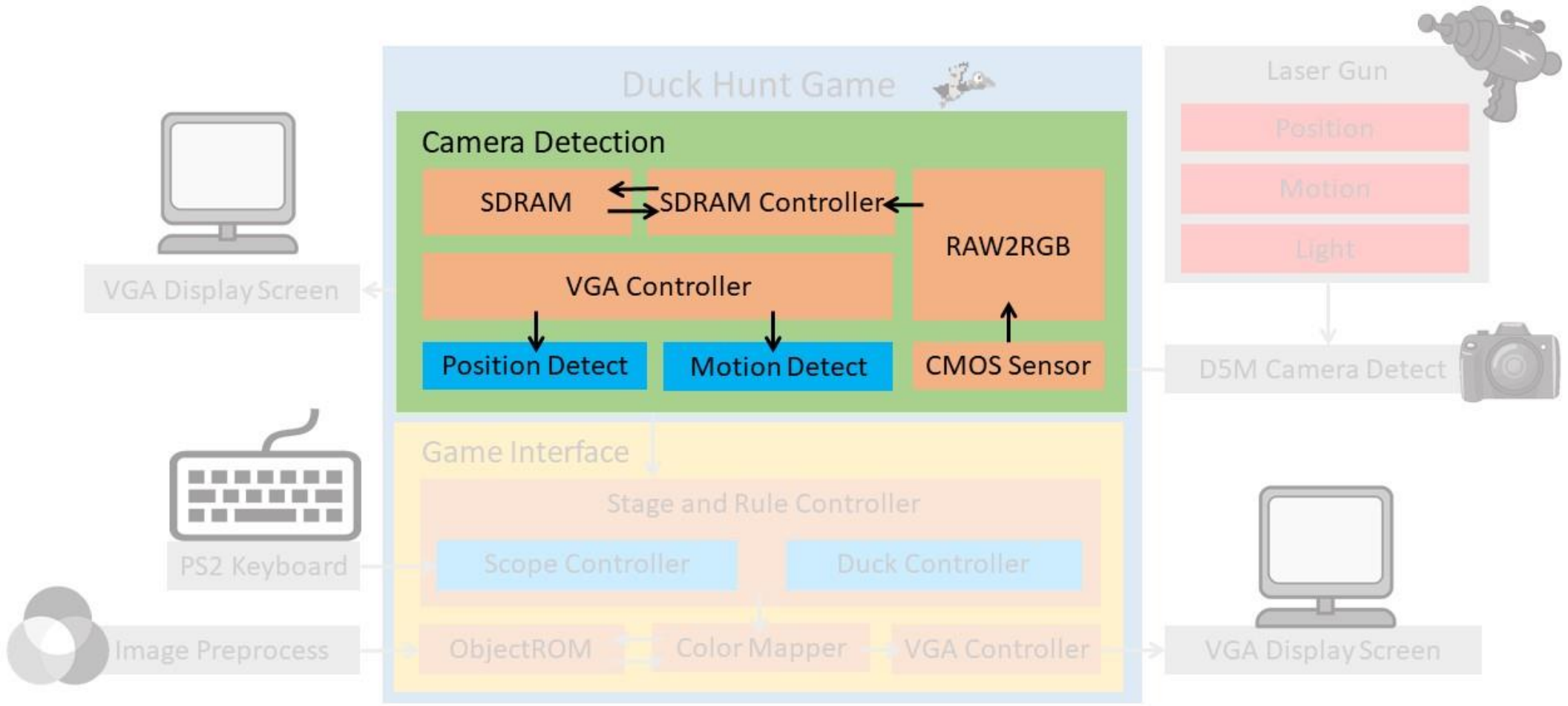


Motion Detection

- Frames position **difference** to detect motion
- **Zoom coefficient** for motion with small angle
- Distance Invariant and motion constraint
 - Get coefficient with light circle size
 - Motion have to ≥ 2 pixels (motion threshold)



Camera Detection





Game Interface

Color Display

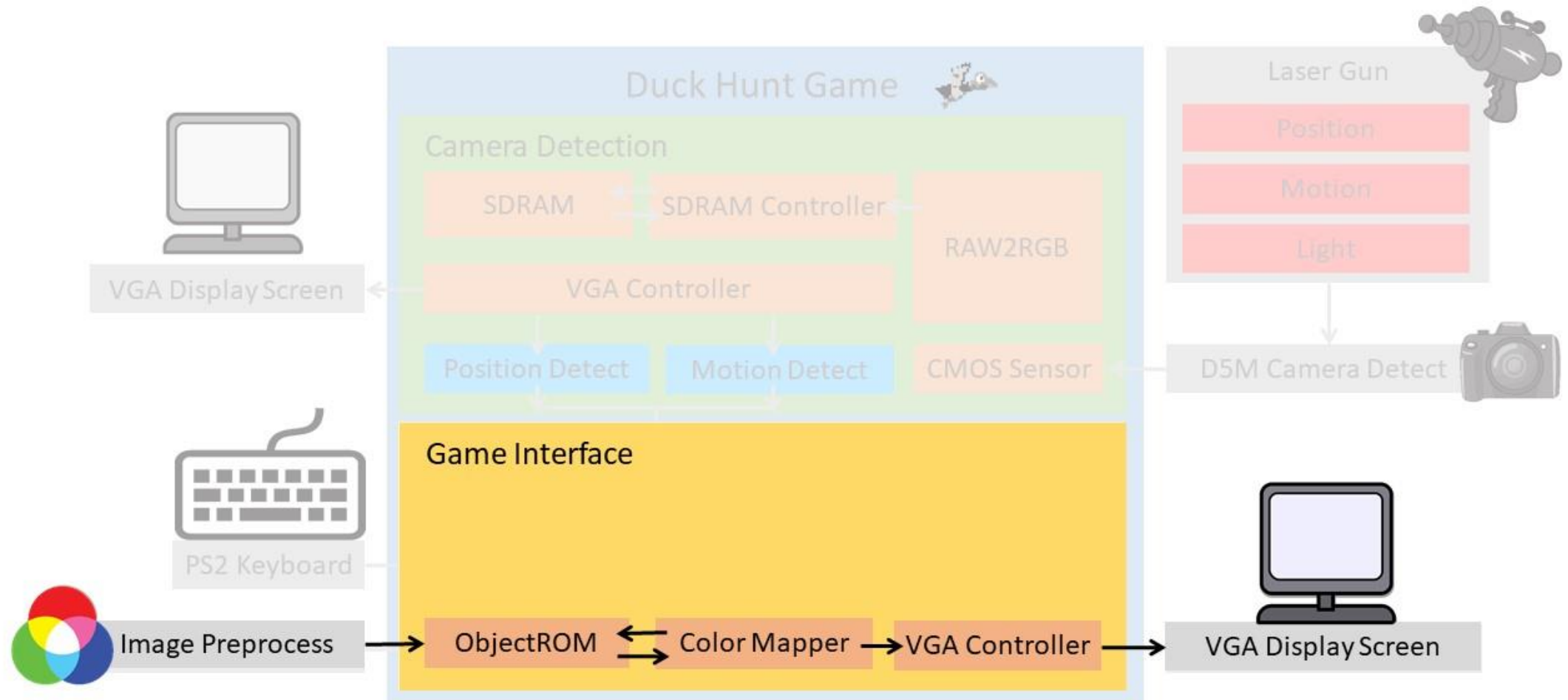
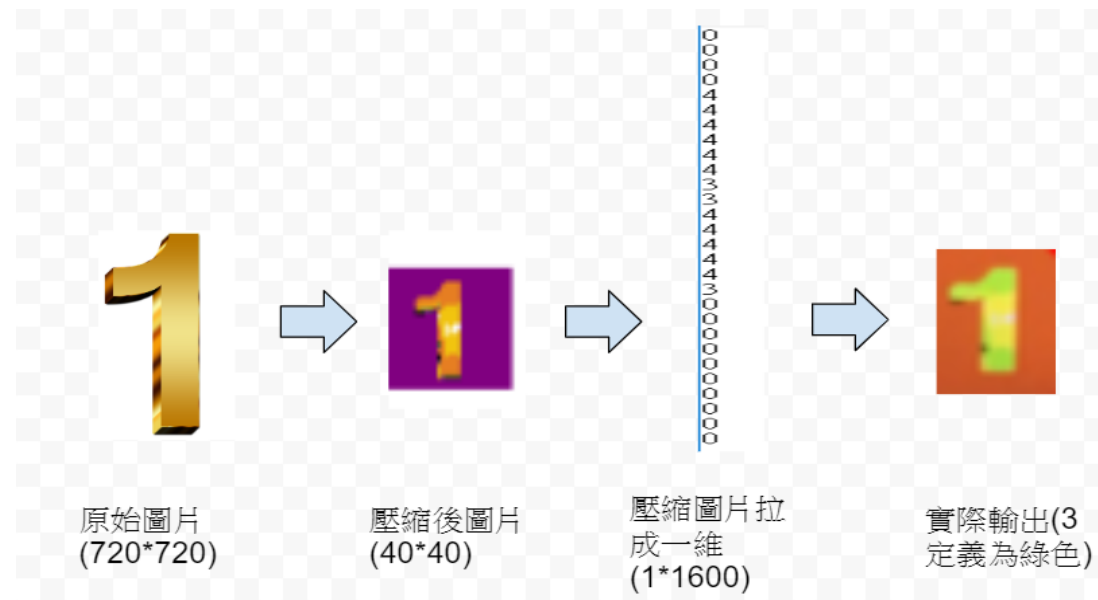


Image Preporcess & ObjectROM

- Change Image size
- Transfer to 1D txt file (one value to represent RGB)
- Use **register** to store every pixel information



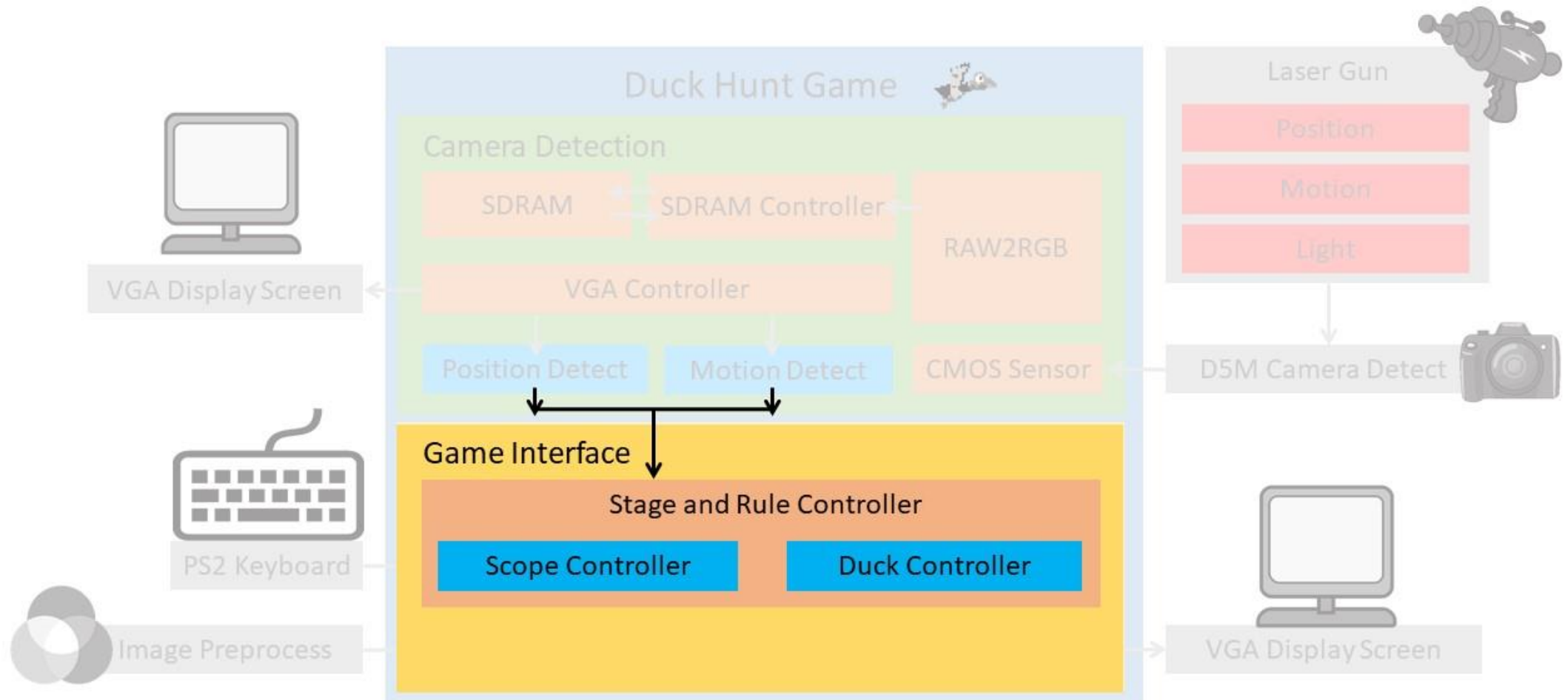
Color Mapper

- Use **flag** to show specific images color in each pixel
- Overlapping image
 - **Ducks** avoid appearing suddenly
 - Scope need to be prior to duck



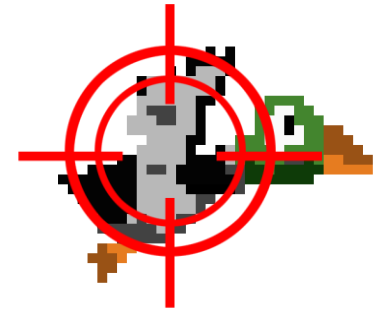
Lives, Score, Home
Scope, Gameover
Start
Grass
Duck
Background

Game Controller



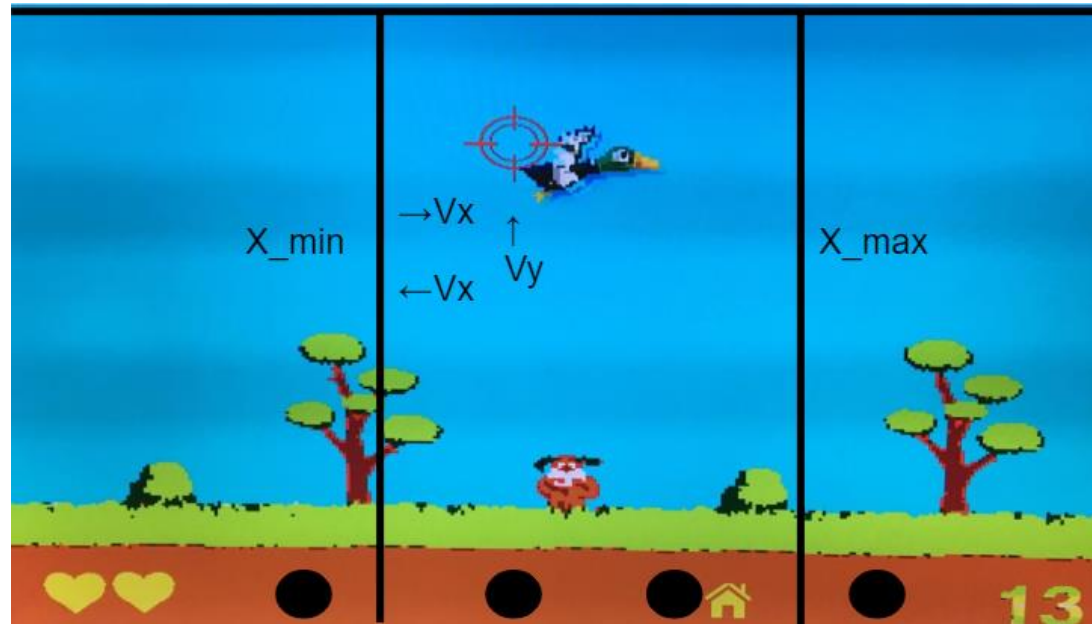
Scope Controller

- **Boundary** Constraint to avoid scope disappearing
 - Initialize center scope in motion mode
- Position mode
 - Horizontal **mirror**
 - Update when find center in one frame
- Motion mode
 - Motion **threshold**
 - Zoom coefficient
- Shoot decision
 - **Dark -> Light** shows one shot
 - Dark will fix the position



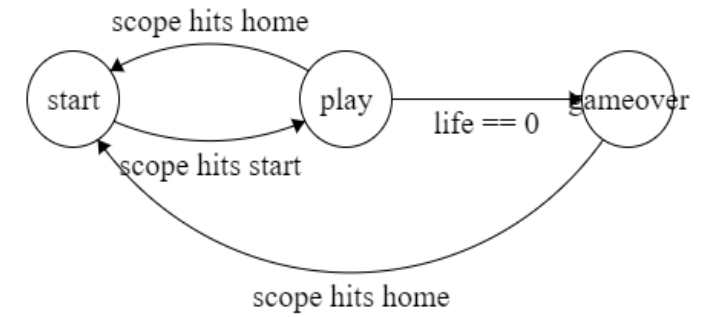
Duck Controller

- Random Move
 - X-axis and Y-axis Speed
 - X boundary for collision
 - Different Initialize position
- Counter and Score
 - Speed
 - Direction
 - Number of Ducks

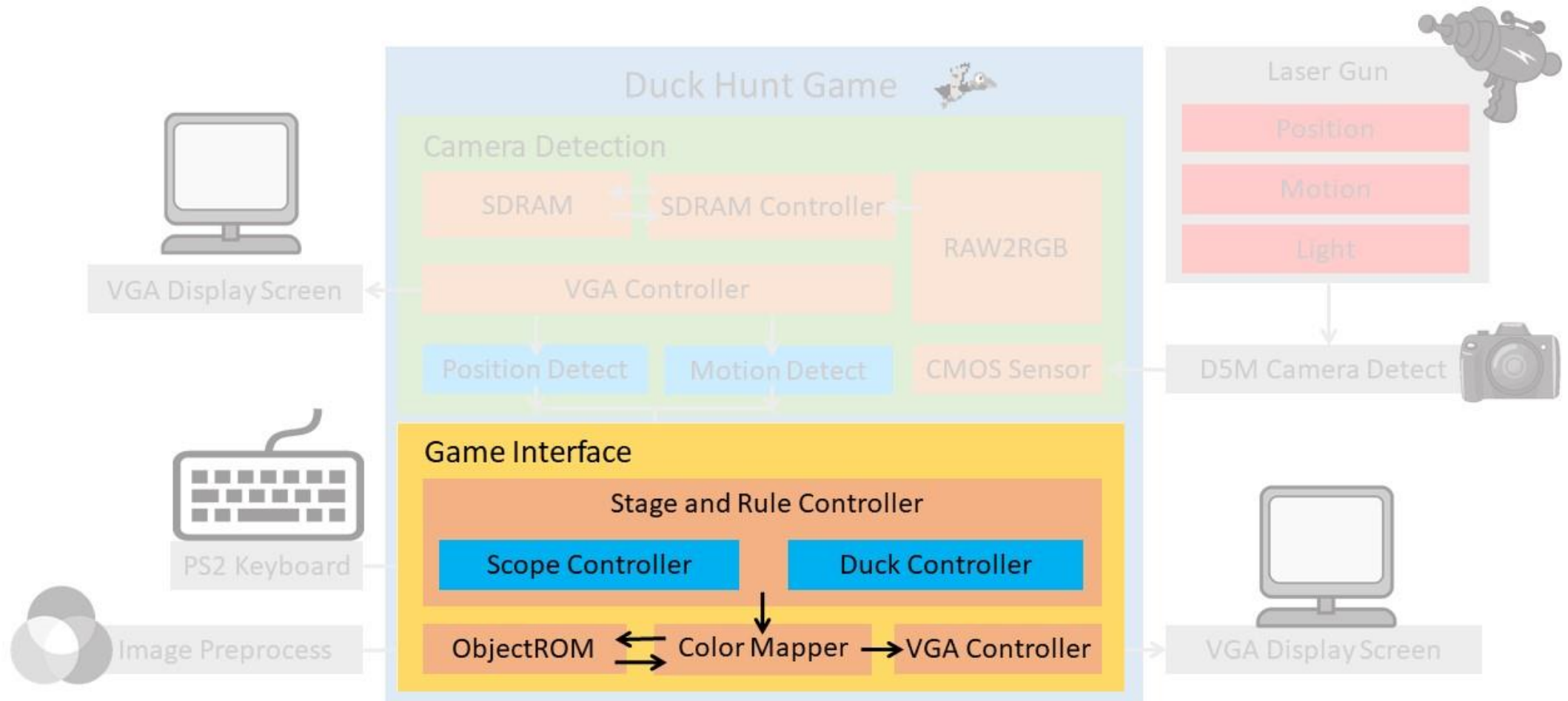


Stage and Rule Controller

- Stage : Start, Play, Game Over
- Lose Lives : Duck position $Y = 0$
- Get Scores :
 - Duck center in the scope square
 - Duck goes down straightly
- Bullet Count : +1 when shoot

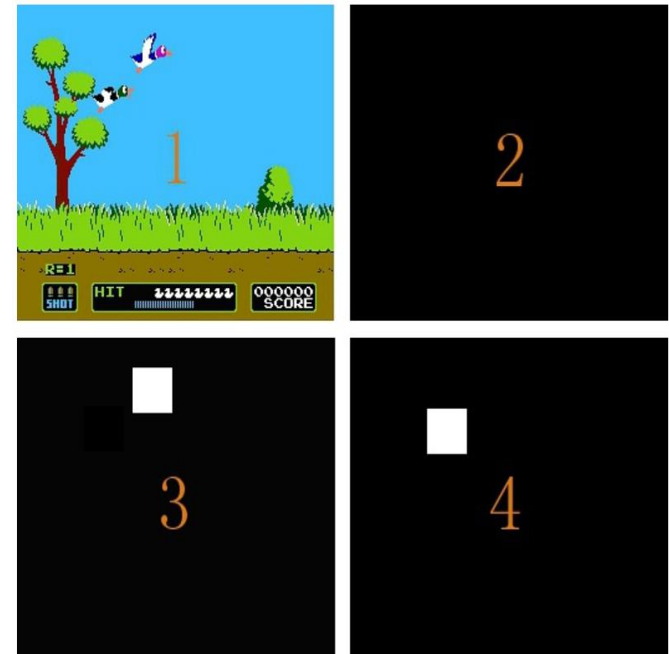
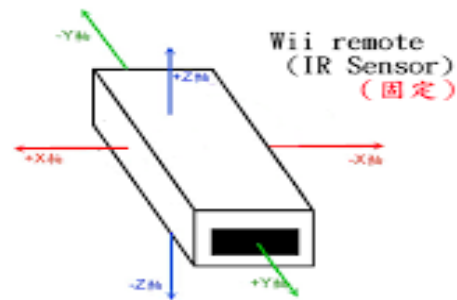


Game Interface



Problem & Discussion-1

- Problem1 : Scope Detection (white light)
 - Position : user **inconvenience**
 - Motion : display **discontinuance** and noise **amplification**
 - Shoot : **jump** position
- Solution1 :
 - Traditional NES Gun
 - Modern IR remote control



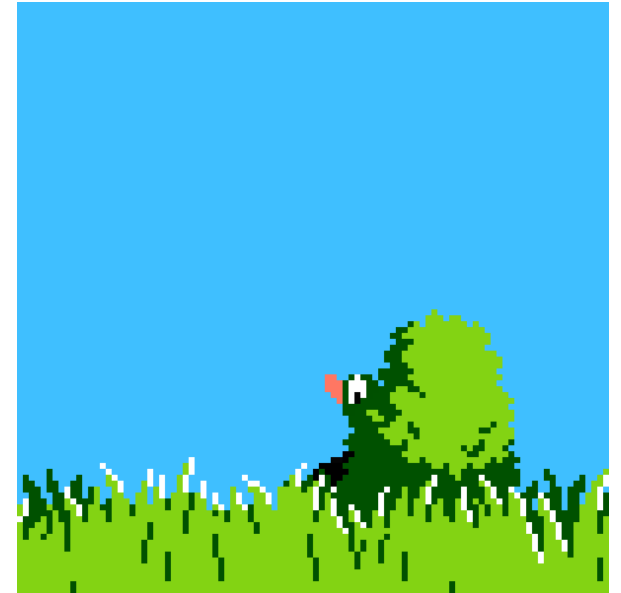
Problem & Discussion-2

- Problem2 : Module Combination
- Solution2 : Different experiment process
 - VGA display for scope detection
 - PS2 keyboard for scope control
- Problem3 : Image position adjustment
- Solution3 : Try and Error (guess roughly)
- Problem4 : Game is too difficult to play
- Solution4 : Practice makes perfect



Future Work

- Scope : Detection enhancement with **IR**
- Hardware : Use **real laser gun**
- Duck
 - **Animation** for wings flapping
 - More random motion path
- Game
 - Add **more players** for competition
 - Implement other shooting game





Demo Time