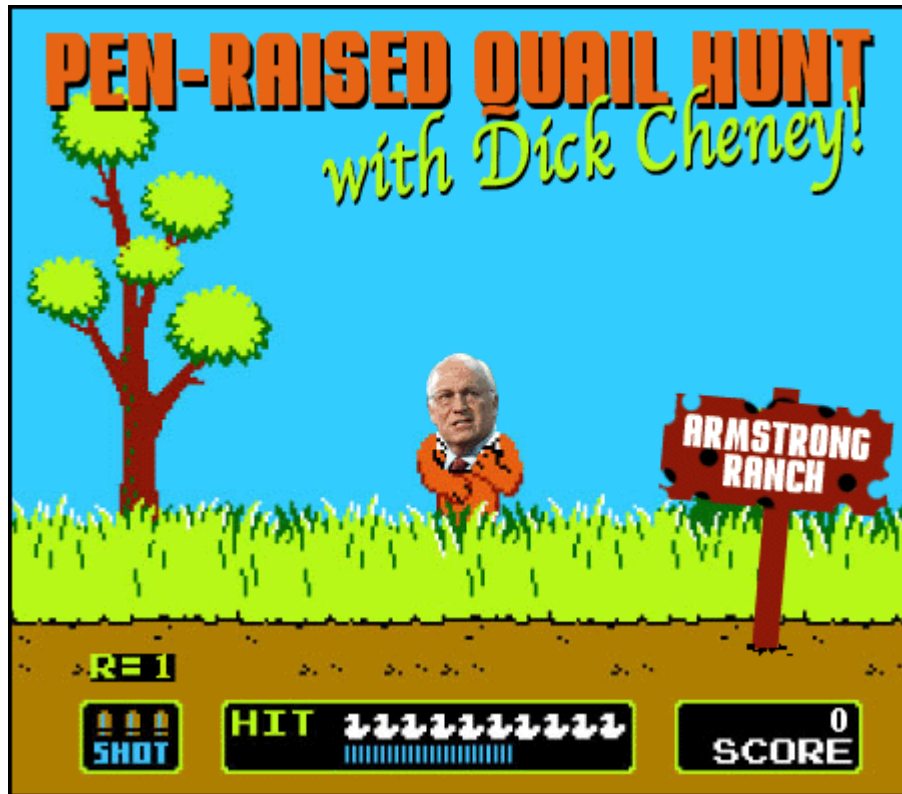




Daniel Lopuch and Zachary Remscrim

6.111
Nov 14, 2006



Daniel Lopuch and Zachary Remscrim

6.111
Nov 14, 2006



Daniel Lopuch and Zachary Remscrim

6.111
Nov 14, 2006



Daniel Lopuch and Zachary Remscrim

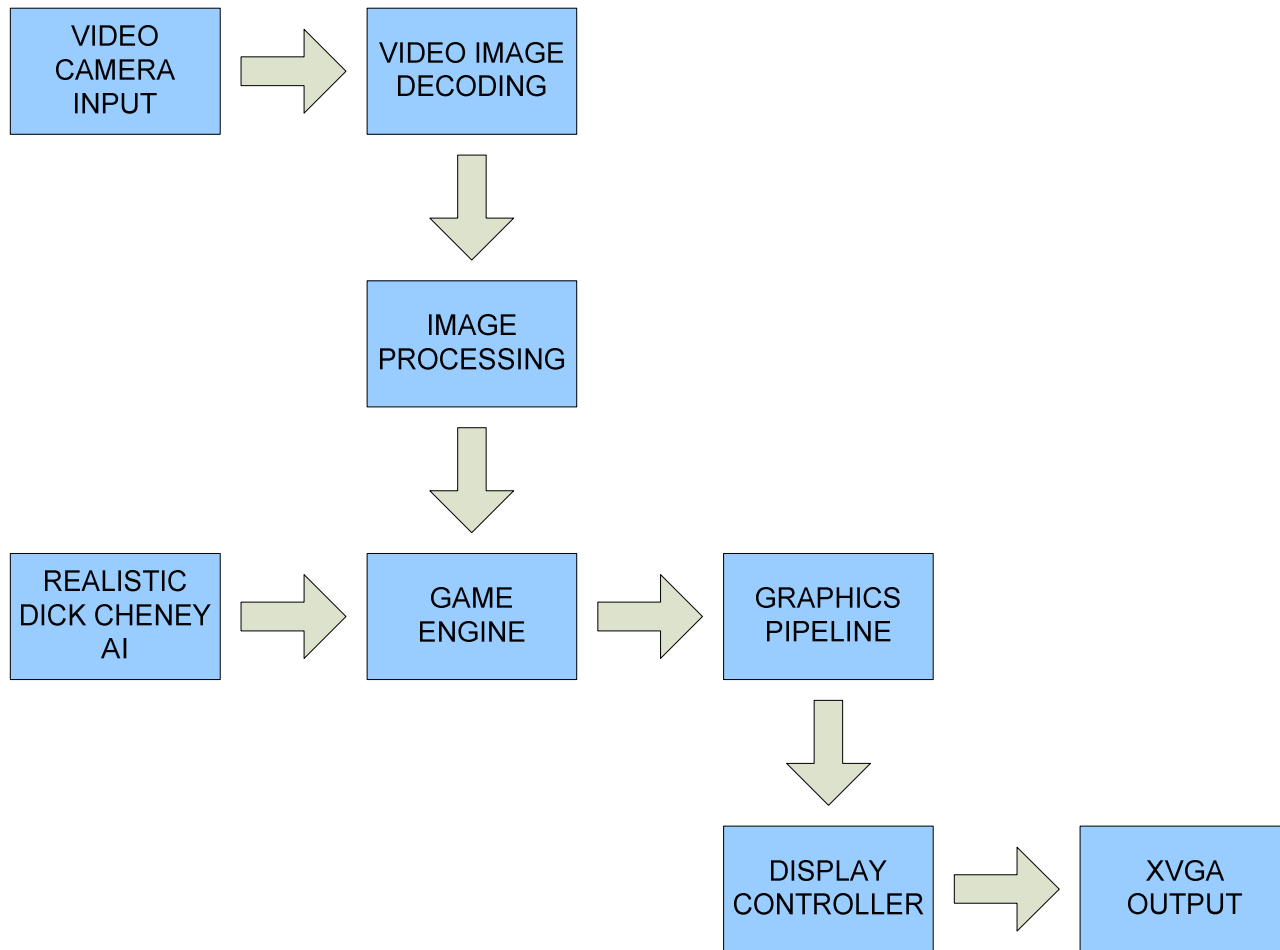
6.111
Nov 14, 2006

Pen-Raised Quail Hunt

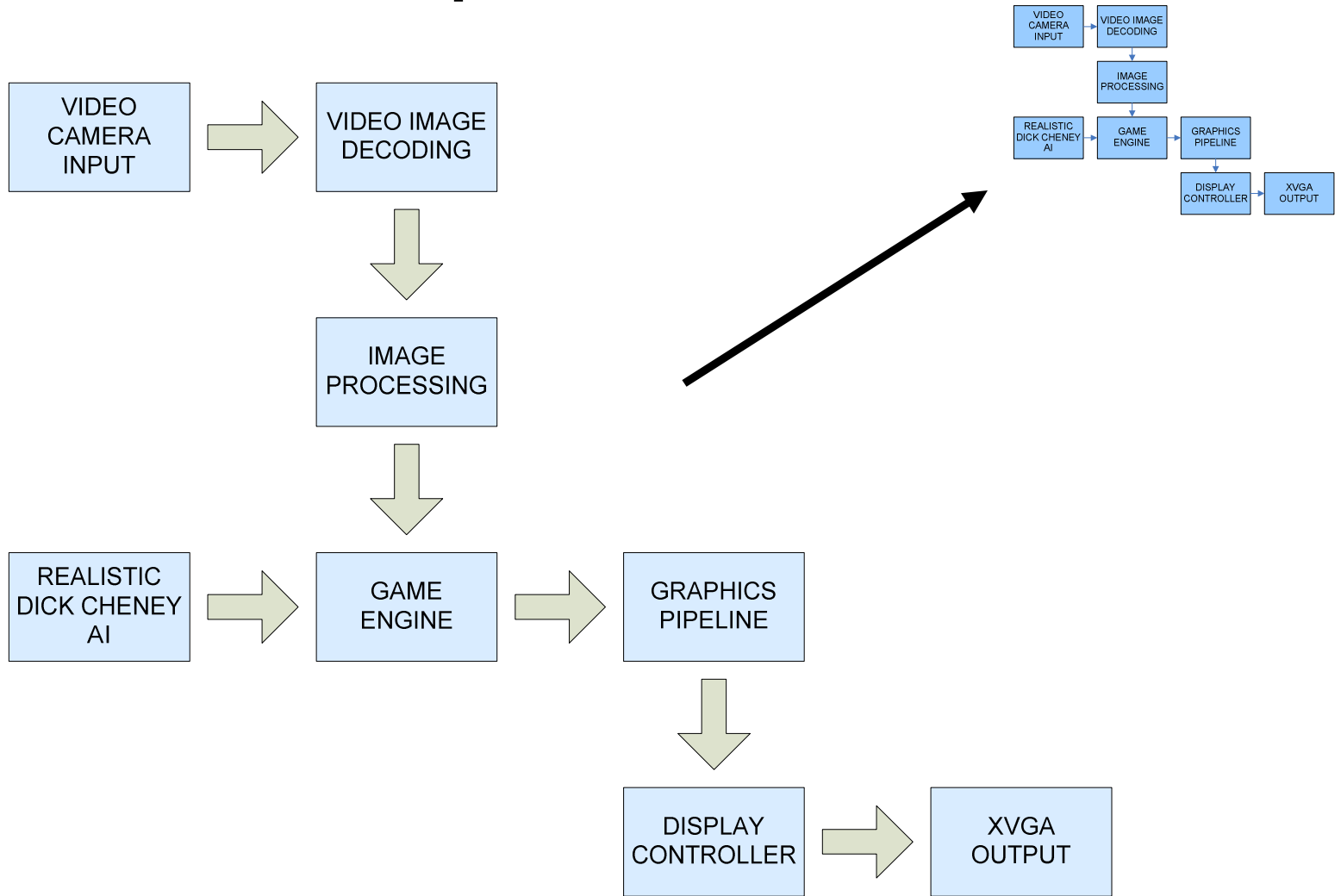
- Features:
 - Quail-hunting simulator, complete with score tracking and pop-up message displays
 - X VGA video display output
 - Aiming done using video camera image processing of player's hand
 - Player moves hand in front of camera to control player gun
 - Test your skills against Realistic* Dick Cheney AI opponent!

*: Realistic Dick Cheney AI never lets a pen-raised quail fly away. If you dally for too long, however, Realistic Dick Cheney AI has tendency to pepper you in the face pretty good.

PRQH Component Overview



PRQH Component Overview



Video Image Decoding

- Video decoding done using 6.111 labkit's ADV7187 chip
- Can be programmed for either NTSC S-Video or Composite input
- ADV7187 initialized using I2C serial bus
- Image Decoder Module writes to a dual-port video buffer RAM

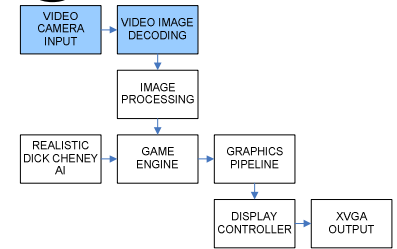


Image Processing

- Controlling player gun description
- Image processing algorithm:
 - Check each pixel location
 - Determine whether it is flesh-colored
 - If it is, store the X and Y coordinates
 - Calculate overall X & Y average and X & Y spread
 - Use centroid to determine position of player gun sight
 - Use spread to determine whether player is shooting or not

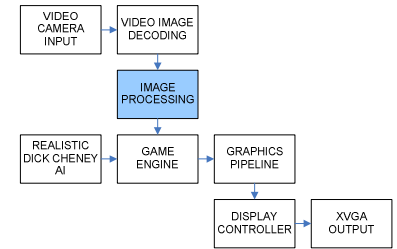
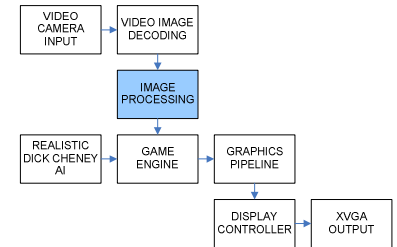


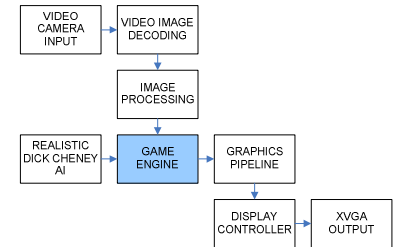
Image Processing

- Include “shot debouncing”
- Need some kind of training session to determine what spread corresponds to a fist (aiming) and what spread corresponds to an open hand (shot)
- Fall-back plan: glove with colored regions

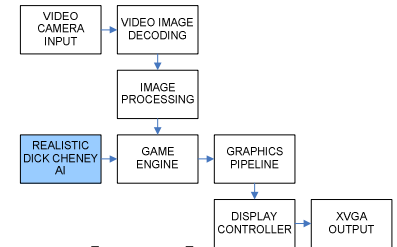


Game Engine

- Stores state of all game objects
- Updates state of all game objects whenever frameUpdate is asserted
- Enforces game rules
- Provides state information to elements of the graphics pipeline



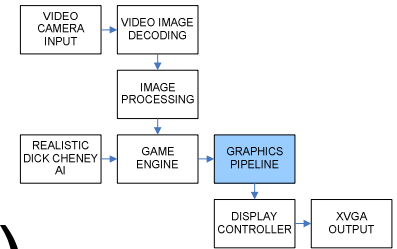
Realistic Dick Cheney AI



- The AI will attempt to move its crosshair to the duck's position, and fire when appropriate
- Varying difficulty levels- speed with which crosshair is moved to the duck and chance of hitting will vary by chosen AI difficulty

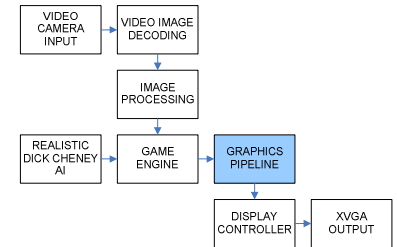
Graphics Pipeline

- Consists of all displayable game objects (duck, tree, crosshair, etc.)
- Given an hCount and vCount, all elements of the pipeline determine the appropriate pixel color, if any, in parallel
- Layers created by having the pixel color for an element higher in the pipeline take precedence



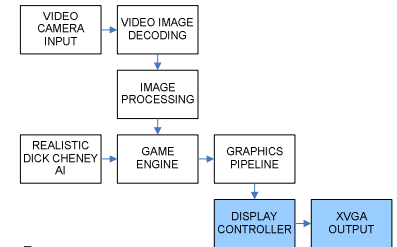
Graphics Pipeline Progress

- Graphics generation through Matlab script
- Example “pong-duck” flying around

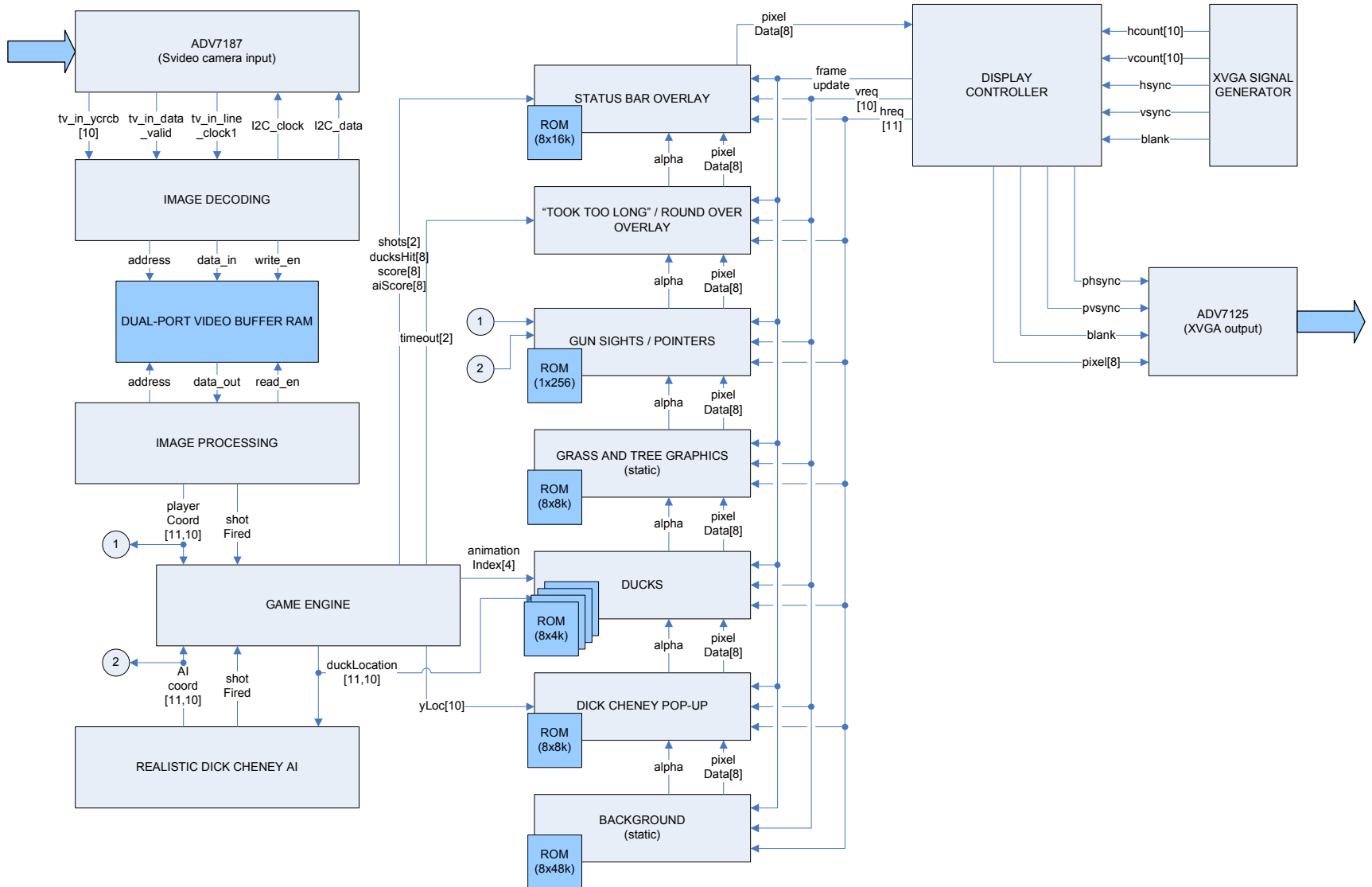


Display Controller

- Generates XVGA timings
- Requests appropriate pixels from the graphics pipeline
- Introduces a pipeline delay if needed
 - Frame buffer expected to be unnecessary
- Outputs to 6.111 labkit's onboard ADV7125 video encoder chip



PRQH Full Block Diagram



Projected Schedule

- Dan's Deadlines
 - Nov 18: Video decoding done
 - Nov 22: Image processing on live video stream able to recognize regions
 - Dec 3: Image recognition working well
 - Dec 3: Text overlay module in graphics pipeline
 - Dec 6: Project integrated
 - Dec 6 onwards: the final 10% / buffer-time

Projected Schedule

- Zack's Deadlines
 - Nov 18: Bottom five graphics pipeline blocks
 - Background, Ducks, Cheney Pop-Up, Grass and Trees, and Gun Sights
 - Nov 22: Game engine done
 - Dec 3: AI Done
 - Dec 3: Status Display graphics pipeline block
 - Dec 6: Project integrated
 - Dec 6 onwards: the final 10% / buffer-time



Daniel Lopuch and Zachary Remscrim

Questions? Comments?