

Pen-Raised Quail Hunt

By Daniel Lopuch and Zachary Remscrim

6.111 Final Project Deliverables Checklist and Specifications

Nov. 22, 2006

Major modules of our project and their planned functionality are as follows:

1) **Game Engine**

- This module will store the current state of the game. It will determine the new state given the current state and inputs.
- Functionality will be demonstrated by observing the game being played and noting that game objects behave as expected while all rules are obeyed.

2) **Realistic Dick Cheney AI**

- This module will be responsible for all AI decisions. The AI will be capable of reliably (but purposefully not perfectly) moving its pointer to the duck's position and firing when appropriate.
- The AI will be capable of operating at various levels of proficiency depending on the current difficulty setting.
- Functionality will be demonstrated by examining the AI's behavior at various difficulty levels.
- If time permits, difficulty will be chosen by selecting the appropriate option from a boot-up menu. Otherwise, difficulty will be selected by setting the appropriate combination of switches.

3) **Graphics Pipeline**

- This collection of modules will be responsible for storing and providing pixel data for all game objects.
- Functionality will be demonstrated by verifying that all game objects are properly displayed on a X VGA display screen at the appropriate times. The list of game objects, a brief summary of when they need to be displayed, and their animated behavior are as follows (listed in bottom-to-top of pipeline order):
 - i. *Background*: this will always be displayed and is static
 - ii. *Dick Cheney Pop-Up*: this will take the role of the infamous Duck Hunt Dog and be animated similarly.
 - iii. *Ducks*: this sprite will fly around when appropriate and fall to the ground when hit. It will be animated such that flight looks like flight and a falling duck looks like a duck that was hit and is falling to the ground.
 - iv. *Grass and Trees Graphic*: this sprite contains the static grass and tree graphics.
 - v. *Gun Sights / Crosshairs*: this sprite contains the two crosshairs: a traditional one controlled by the player's motions, and a patriotic American flag-shaped one controlled by Dick Cheney.

- vi. *Round-Over Overlay*: this will display the appropriate message or taunt text when a round is over.
- vii. *Status Bar*: this sprite displays all of the player's game information, including score, number of ducks hit, number of ducks Dick Cheney hit, and number of shots left.
- viii. *Menu* (if time permits): this will be a splash-screen that allows the player to choose appropriate starting conditions, including difficulty and 1-player vs. 2-player.

4) **Image Capture**

- This module will capture a frame of incoming video from the camera.
- Functionality of this module will not be demonstrated directly, but indirectly through the functionality of the Image Processing module.

5) **Image Processing**

- This module will examine the position and shape of the player's hand to determine the current position of the player's crosshair and when they fire.
- *Ambitious Player Tracking*
 - i. Moving the crosshairs is hoped to be achieved by the player moving a closed fist inside the camera's field of view and tracking flesh-colored pixels.
 - ii. When the player wants to fire, it is planned that he can do so by opening his fist. Ideally, a statistical spread or variance of the flesh-colored pixels will be enough to reliably tell the difference between the two.
 - iii. However, there are many variables that may affect the reliability of this method, including variance in hand distance from camera (effective size of fist), camera field-of-view (distortion around edges of object), and camera blur of moving objects (pixel variance distortion). If these problems do indeed prove to be insurmountable in the limited time given and limited time afforded by an MIT schedule, player tracking will be accomplished through Plan B.
- *Running-Out-Of-Time Player Tracking*
 - i. If time will be running short and if it will seem that the statistical variance tracking method will be unable to be implemented reliably, the backup plan is to make a player glove with different color splotches in different areas and use simple color recognition to both track the player's crosshairs and shoot.
- The functionality will be demonstrated through playing the game, where the player's crosshair will move with the player's hand and where the player shoots when appropriate.
 - i. A "visual debugging" mode, where in real-time it displays the camera image and what pixels are recognized as player pixels, will also be demonstrated if, during testing, it was created out of necessity.