



NYU

POLYTECHNIC SCHOOL
OF ENGINEERING

Duck Hunt 2014

Dhaval Palsana and Paul Phamduy
Advanced Mechatronics Arduino Project
March 25, 2014





Duck Hunt

- Popular iconic Nintendo Entertainment System (NES) shooter video game released in April 21, 1984.
- Players can shoot ducks that appear on the screen.
- Original video game hunting gun operated using light-sensing vacuum tube TV screens.





Augmented Reality

- Augmented reality is defined as the a view of the real-world environment that has been supplemented by computer-generated sensory input.
- Augmented reality is becoming increasingly popular in entertainment, fashion, and smartphone technology.





Revamp classic Duck Hunt video game with augmented reality

- Duck Hunt is a nostalgic 90's game that we grew up playing. Simple but addictive!
- Rebuilding the game in augmented reality was an interesting challenge. Fun to implement and play with!
- Apply concepts in the classroom to practical knowledge and the development of a augmented reality video game. (Arduino and OpenCV)



(Selfie!)



Logitech Webcam

OpenCV



HP Laptop

Two-way serial communication



- To laptop
 - 0-170 servo position
 - 171 hit
 - 172 end game
- To Arduino
 - 1-8 LED states
 - 9 reset



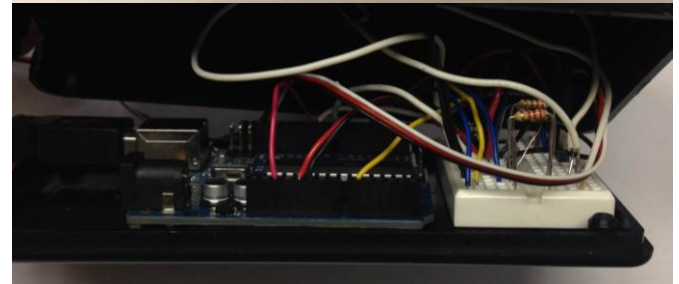
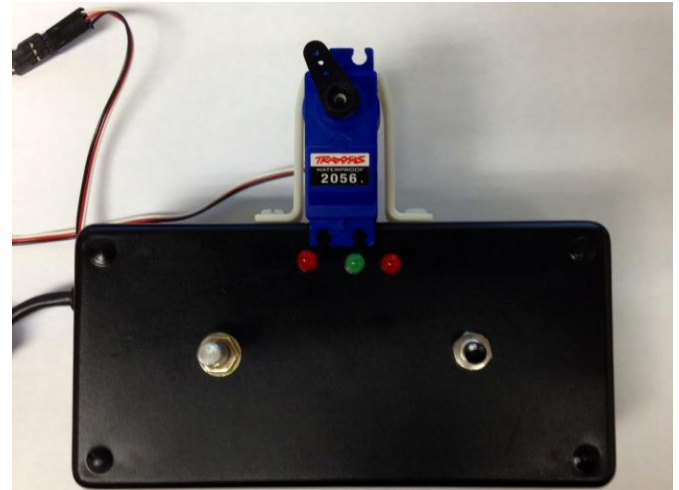
Arduino UNO

- Button
- Potentiometer
- LEDs
- Servomotor
- Reset switch



Hardware

- A potentiometer is used to control the crosshair position.
- A push button is used for the trigger to hit the ducks.
Implements an active low button configuration.
- Three LEDs are used to indicate the location of ducks. Red LEDs indicate the general position of ducks. Green LED indicate that the crosshair is on the duck.
- A servomotor is used to adjust the view of the webcam.





Arduino

- Potentiometer on pin a0. LEDs on pin 2-4. Push button on pin 7.
Servomotor on pin 9. Reset wire on pin 10. Setup() initializes sensors and actuators and communicates over serial on COM 7 at a baud rate of 9600.
- The LEDs change state when Arduino receives a char from '1' to '8'.
- The servomotor position is continually updated with myservo.write().
- The program ends the user controls after three minutes.
- The Arduino can be reset when a '9' char is received over serial.



```
project_adv_mech1 | Arduino 1.0.1
File Edit Sketch Tools Help

project_adv_mech1

int center=3;
int resetPin = 10; // reset pin on 10
const int buttonPin = 7;
unsigned long time =0;

void setup()
{
  digitalWrite(resetPin, HIGH);
  delay(200);
  pinMode(resetPin, OUTPUT);
  myservo.attach(9); // attaches the servo on pin 9 to the servo object
  Serial.begin(9600);
  pinMode(buttonPin, INPUT);
  pinMode(left, OUTPUT);
  pinMode(center, OUTPUT);
  pinMode(right, OUTPUT);
}

void loop()
{
  time=millis();
  if (time<180000)
  {
    sensorValue = analogRead(sensorPin);
    sensorValueforservo = map(sensorValue, 0, 1023, 0, 170);

    if (Serial.available())
    {
      char led= Serial.read();
      switch (led)
      {
        case '1':
          digitalWrite(right,HIGH);
          digitalWrite(center, HIGH);
          digitalWrite(left, HIGH);
          break;

```



OpenCV

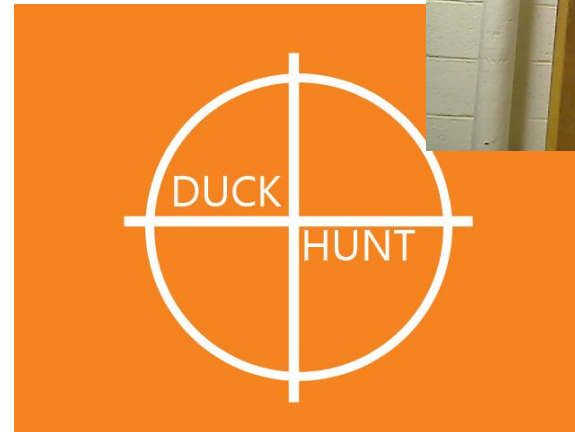
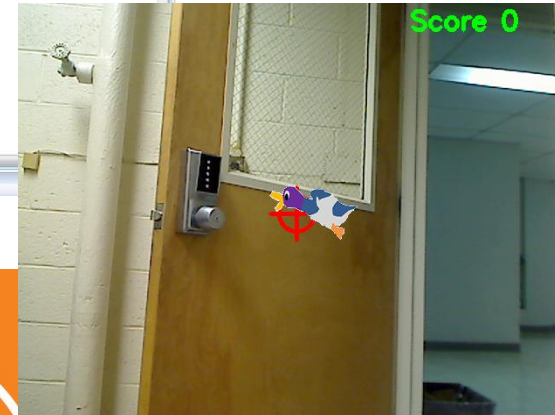
- OpenCV grabs frame from frame grabber on the web camera and overlays it with current score information, crosshair and the image of the duck.
- Logical operations are performed in order to display the duck with live feed background image.
- A mapping operation is performed based on the current crosshair position, such that a value of 50 degrees on the servo corresponds to one frame on the screen.
- Targets are randomly selected from a predetermined target array.
- Soundtrack and effects are added to enhance gameplay.

5/28/2014

```
Advance Mechatronics.cpp | Start Page
[Global Scope] | main(ikiarray-System:String *1) -> arg1
// Advance Mechatronics.cpp : main project file.
#include "stdafx.h"
#include "window.h"
#include "stdafx.h"
#include "cv.h"
#include "highgui.h"
#include "core.h"
#include <time.h>
#include <stdlib.h>

using namespace System;
using namespace System::IO::Ports;
//using namespace cv;

int main(array<System::String> ^ args)
{
    // Initialize Camera Capture
    CvCapture* capture = cvCaptureFromCAM(2);
    if ( !capture )
    {
        fprintf( stderr, "ERROR: capture is NULL \n" );
        getchar();
        return -1;
    }
    cvNamedWindow( "Duck Hunt 2014", CV_WINDOW_AUTOSIZE );
    String* cursor;
```





**GAME
OVER**

**Thank you for your attention!
We would be happy to answer any
questions.**

