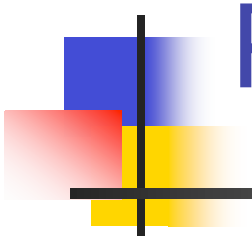
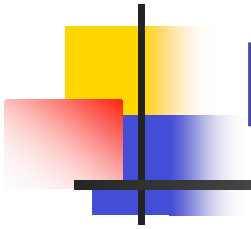


Programming FRC Robots

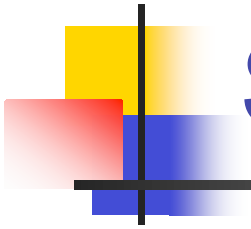
CMU2 Camera



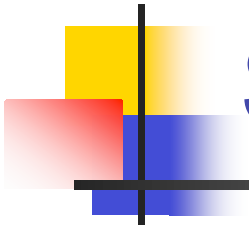
Presenters

- Michael DiRamio:
  - Programming and Controls mentor on 1114 for the past 2 years.
  - High School teacher for District School Board of Niagara
  - E-mail: Michael.DiRamio@dsbn.edu.on.ca
- Tyler Holtzman:
  - Mechanical and Programming mentor on 1114 and 2056.
  - Started in 2003 as a student on 1114.
  - Software Engineering Student at McMaster University

Programming Concepts: Structures



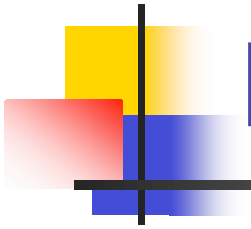
- Hold a collection of values
- Define them with typedef
- Treat like any variable, access the different values by <struct variable>.<variable>



Structure Example

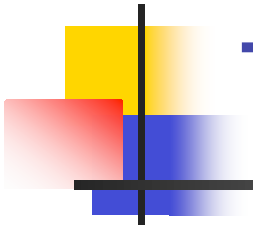
```
typedef struct {
    int a;
    int b;
} myStruct;

myStruct ms;
ms.a = 5;
```

Programming Concepts: Define Statements



- Pre-processor operative.
- Replacing one “word” with something else before compilation

```
#define TEST 5
x = x + TEST    // TEST will be replaced by
                // “5” before compiling
```



The Camera

- The CMU2 Cam has been used for the past three years in FRC competitions.
- The first year it was difficult to use because the target was a painted object, that looked different under different lighting.
- The past two years the target has been a green light, made of cathode tubes (think suped-up computers).

Game Objectives with Camera

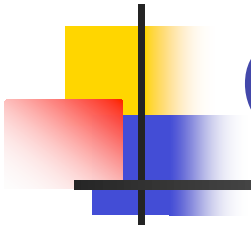
- 2005 - score the tetra on top of the green triangle for extra points
- 2006 - shoot balls into the stationary goal beneath the green light
- 2007 - put rings on posts located below the light. The posts were moved before the competition started.
- Changing Colours - rumoured at the championship event in 2006.

Setting up the Camera

- TTL Chip - attached to the RC
- Information: Connect to PWM connector near the serial port with the black pin facing out.
- Power: Connect another PWM cable between the other PWM connector and a spare analog input on the RC.
- Secure Wires: All cables should be secured into the camera and RC using either silicon or hot glue.

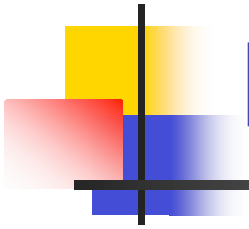
Lights on Camera

- Green Light - Has power
- Red Light - can see the target (if it has loaded properly from the RC).
- Put your hand over the camera. If the red light goes off when covered and on when it can see the light, the camera is functioning properly.



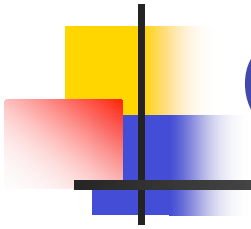
Camera Mount

- Provided in the kit of parts.
- Uses two servos to move the camera.
- Assembly instructions provided on www.ifirobotics.com
- Hook up the servo motors to PWM outputs on the RC.



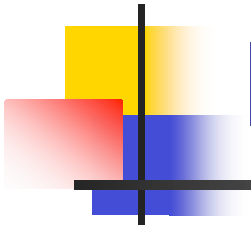
Kevin Watson Code

- Designed to be integrated into the provided base code.
- Available at www.kevin.org
- Rules about publishing code that includes Kevin Watson code.
- Code for
  - Encoders
  - Gyros
  - CMU2 Cam
  - Serial Ports
  - EEPROM



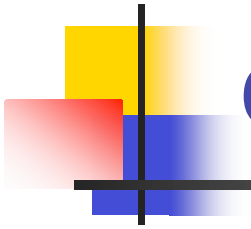
Camera Code

- Two versions: “Bells and Whistles” and “Streamline”.
- Bells and Whistles includes text-based utilities to change camera values.
- Streamline version has camera functionality. All variables are modified in “.h” files.
- Installation instructions are provided in camera_readme.txt



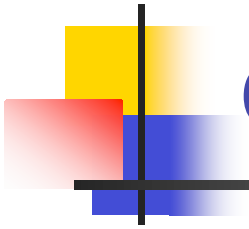
Important Files

- Breaks down into two main files:
  - camera.h and camera.c: the code that communicates with the camera
  - tracking.h and tracking.c: the code that controls the values of the pan and tilt servos on the camera mount.
- The “.h” files contain all of the definitions and variables for the camera and mount.



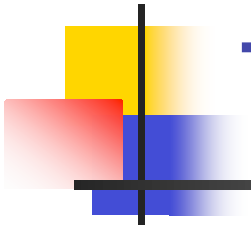
camera.h definitions

- CAMERA_SERIAL_PORT_1/2
  - Uncomment the proper define for which serial port you are using. Programming port is 1, TTL port is 2
- R/G/B_MIN/MAX_DEFAULT
  - Set colour ranges for the target
  - Actually set as YCrCb, so names are confusing



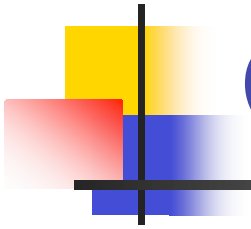
camera.h definitions (2)

- NF_DEFAULT
  - The noise filter value. Raise this if the lighting is confusing the camera.
- Contains the definition of a struct, T_Packet_Data_Type, and one instance of the struct, T_Packet_Data



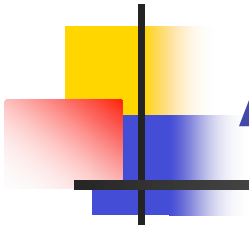
T_Packet_Data

- Contains information about where the target is in the camera's field of view:
  - mx: x-value of the middle of the target
  - my: y-value of middle of the target
  - (x1, y1): left most corner of target
  - (x2, y2): right most corner of target
  - pixels: number of pixels in the target
  - confidence: amount of the target box that is actually the right colour



Camera.h methods

- Camera_Handler
  - Should be called in every slow loop
  - Takes care of the communication with the camera
    - Initialization
    - Updating T_Packet_Data
  - The only important method in camera.h, most others are used internally to communicate with the camera.

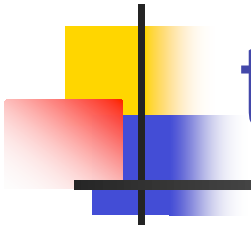


Aiming with T_Packet_Data

- Use the information to guide your robot to the target

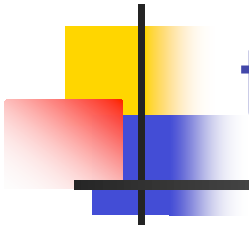
```
if (T_Packet_Data.mx < target)
        turn right
else if (T_Packet_Data.mx > target)
        turn left
```

- Use the same idea with the “my” value to move your robot forward or backwards.



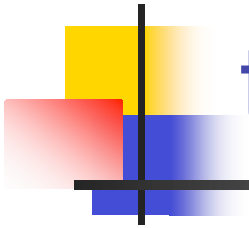
tracking.h definitions

- PAN_SERVO and TILT_SERVO
  - Specify the pwm outputs the servos are connected to
- SEARCH_DELAY_DEFAULT
  - # of slow loops to wait before moving to next search step
- CONFIDENCE_THRESHOLD_DEFAULT
  - Minimum confidence value to switch from searching to tracking



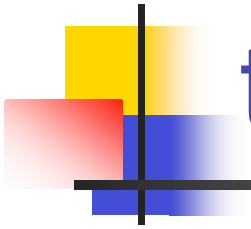
tracking.h definitions (2)

- PAN/TILT_GAIN_DEFAULT
  - Set the speed of the tracking. Change if the camera is tracking too slowly, or is oscillating/jumping past the target.
- PAN/TILT_ROTATION_SIGN_DEFAULT
  - Set the direction the servo moves to follow the target. Change these if the camera seems to “run away” from the target.



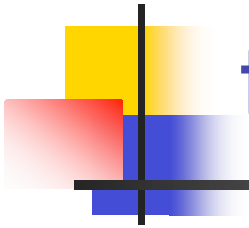
tracking.h definitions (3)

- PAN/TILT_TARGET_PIXEL_DEFAULT
  - The mx/my values where the tracking tries to keep the target in the camera view.
- PAN/TILT_ALLOWABLE_ERROR_DEFAULT
  - # of pixels the camera can be off from the target pixel values.



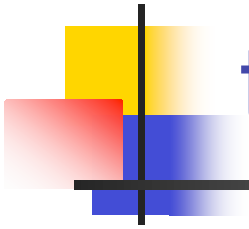
tracking.h definitions (4)

- PAN/TILT_MIN/MAX_PWM_DEFAULT
  - The range of PWM values that are sent to the servos.
  - Specifies the range that the camera will search/track within
- PAN/TILT_CENTER_PWM_DEFAULT
  - PWM values when the camera is pointed directly at the target.



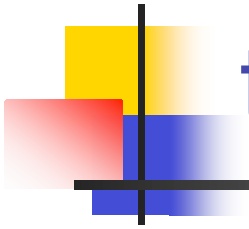
tracking.h definitions (5)

- PAN/TILT_SEARCH_STEP_SIZE_DEFAULT
  - How far the servos jump on each search step



tracking.h methods

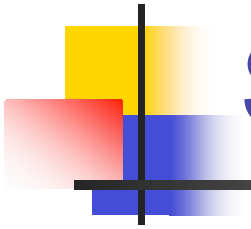
- Servo_Track
  - Call this every loop, handles the updating of the servo values based on camera values
- Get_Tracking_State
  - One of three states:
    - SEARCHING: can’t see the target
    - TARGET_IN_VIEW: can see the target, but not locked on to it
    - CAMERA_ON_TARGET: locked on to the target



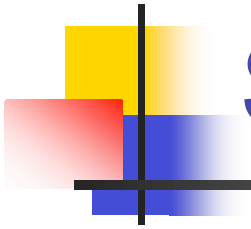
tracking.h methods (2)

- Set_Tilt/Pan_Servo_Position
  - Override the current positions for the tracking servos.

Aiming with Servo Values



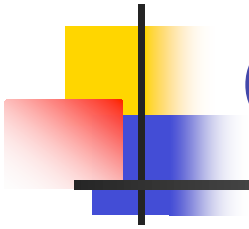
- When using the camera mount, the mx and my values should always be centred on the target.
- Instead of using these values, use the servo values of the pan and tilt servos.
- These values are outputs, but can be read as well.



Servo Value Example

- PAN_SERVO and TILT_SERVO are defined in tracking.h

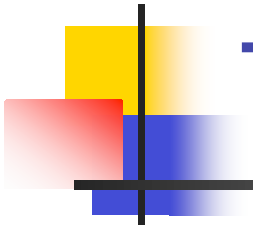
```
if (PAN_SERVO < 117)
        turn left
else if (PAN_SERVO > 137)
        turn right
```



General Programming Ideas

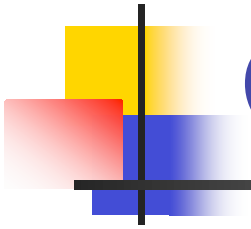
- KISS: the most straight forward solution is almost always the most effective.
  - Don’t need to do trig to find distances, you can use the servo values to tell you the angle.
- Build your program in small progressive steps
  - Set goals in order and move through them in order
  - It’s better to have the camera help a little effectively than try to help a lot and fail to succeed.
- Don’t rely on the camera too heavily
  - It is a very delicate device, ensure that your robot can be effective even if it fails.

Trouble Shooting: Target Light



- A lot of problems with the camera come back to problems with the light.
- The light must have the proper power.
  - If powering off batteries, change often.
- Environment Lighting
  - Fluorescent lights can confuse the camera (it thinks they are the target). Up the noise filter value (NF) in camera.h

Trouble Shooting: Camera Cables



- If the power cable is disconnected, no lights will be on
- If the TTL cable is disconnected, both the green and red lights will be on.
- After plugging cables back in, reset the robot to reinitialize the camera.

Trouble Shooting: Locking on to Target

- Interference from lighting
  - Increase noise filter value
- Out of focus
  - Use Labview or java GUI to see what the camera sees. Rotate lens to change focus.
  - Java GUI: http://www.cs.cmu.edu/~cmucam2/downloads.html