Group IV

Multi Touch Poker Table_{MTPT}

Final Presentation

Nathaniel Boucher
Christopher Herod
Raeginald Timones

Project Description

- Multi-Touch screen to allow four users to enjoy a game of Texas Hold'em

Project Description

- Two cards will be sent to the players iPhone/iPod Touch application
- Players can then fold using application or bet using the touch screen table.
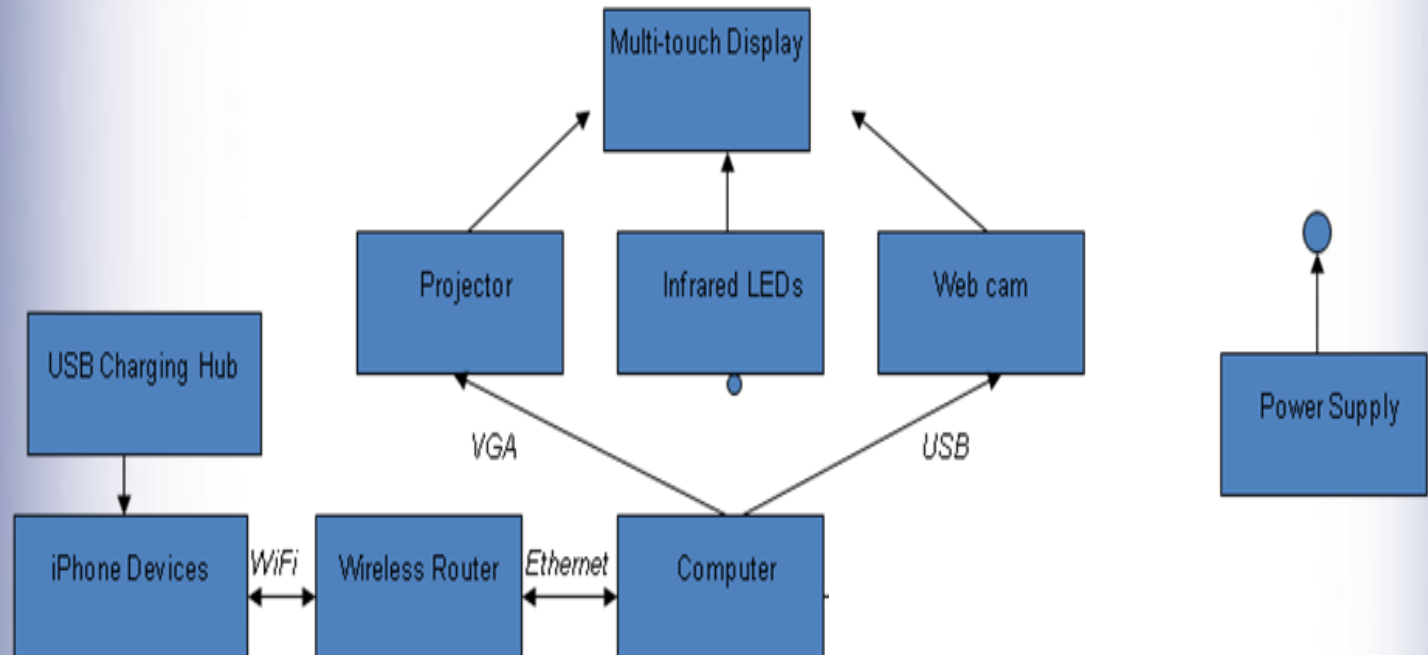
Goals and Objectives

- To explore alternative multi touch techniques
- To create an entity that will compliment a public environment such as a café or restaurant
- To learn basic constructs of iPhone development
- To become more proficient in C++ software development

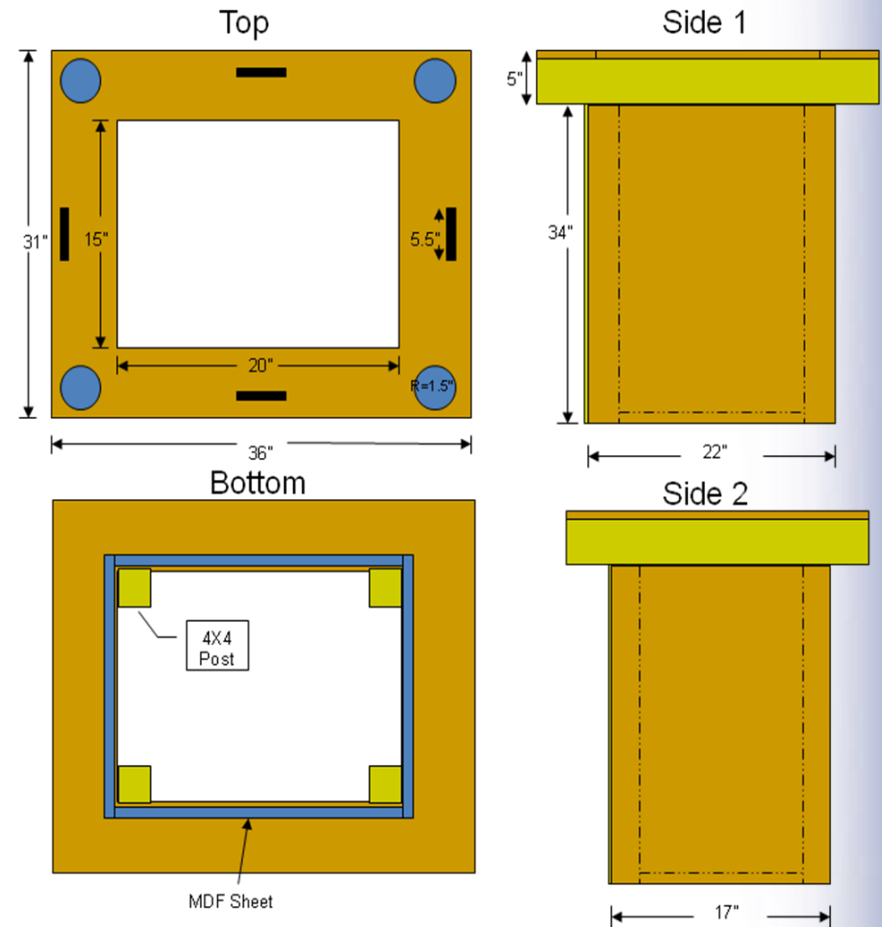# System Block Diagram
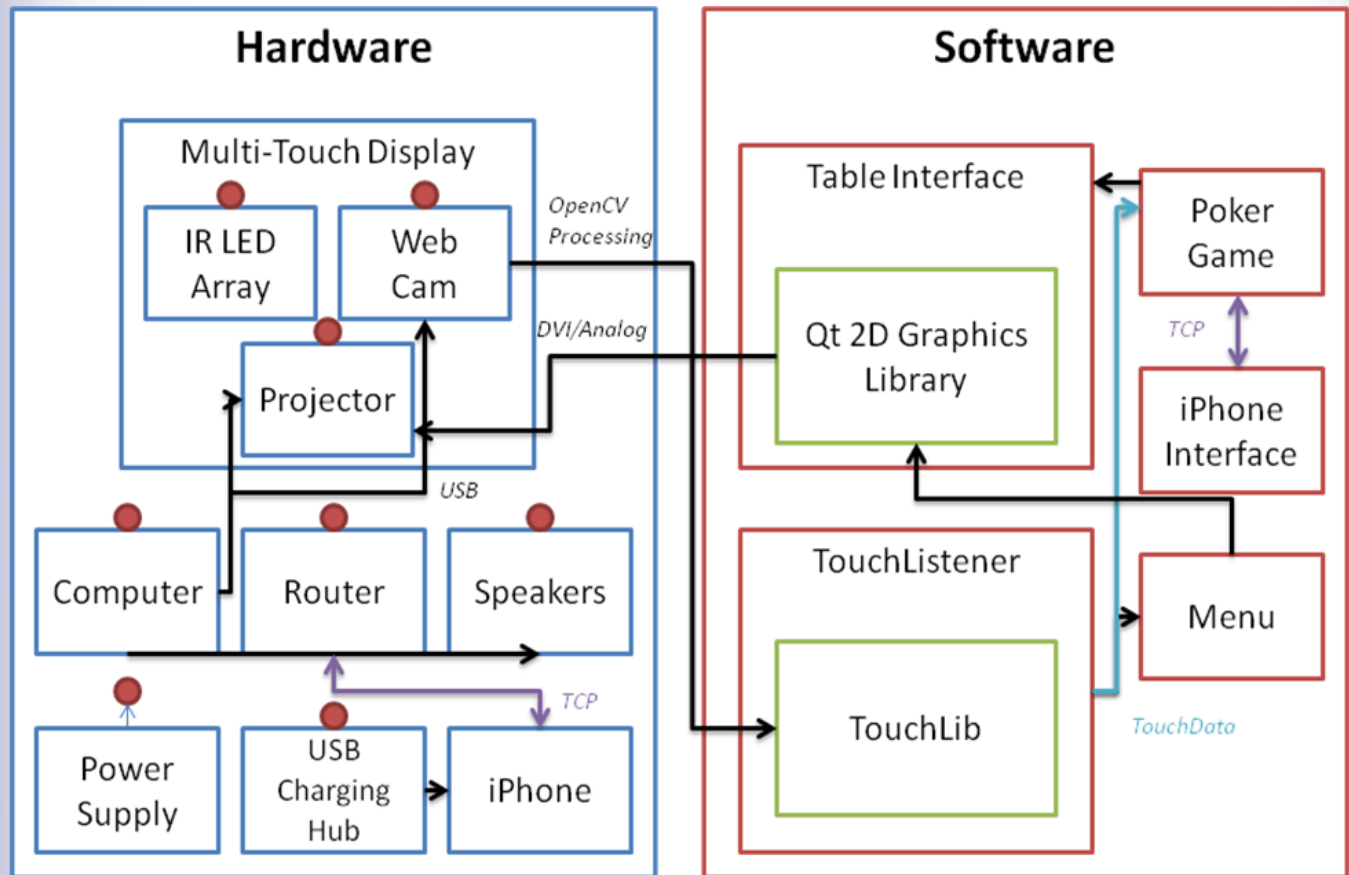
Hardware Components

Framework

- GOALS

- Used in Coffee shop

- Phone holder with charging capability

- REQUIREMENTS

- 25" screen

- Height < 40"

Framework Build

- One 4'x8' sheet of Birchwood

- Two 1"x6" pieces of Pine
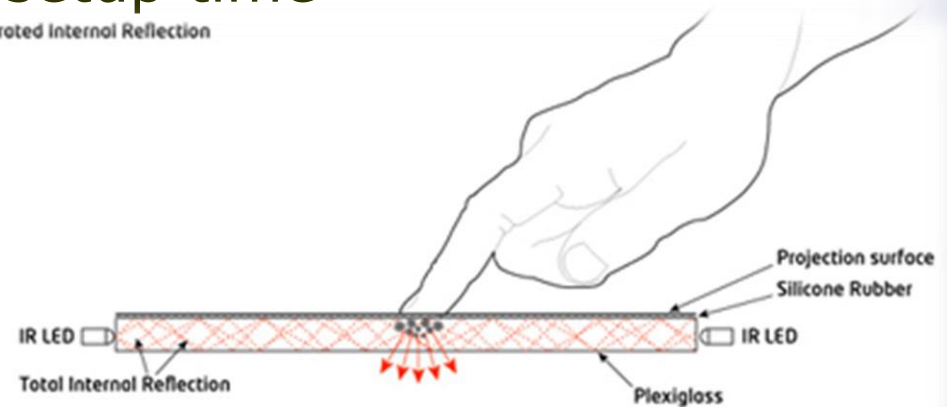
Multi-Touch Display

Goals

- High touch sensitivity
- Operation in bright and dark rooms
- Touch pressure sensitivity
- Low cost

Multi-Touch Technique

- FTIR

  – IR light flooded through Acrylic surface

  – High sensitivity
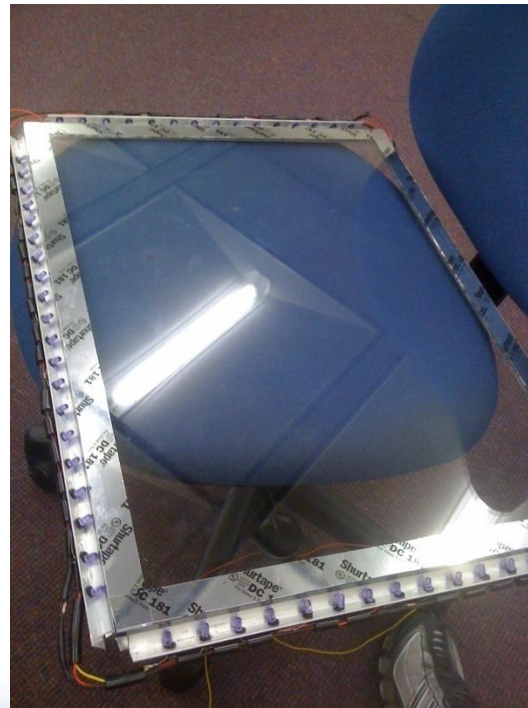
  – Long setup time

FTIR - Frustrated Internal Reflection

Projection surface
Silicone Rubber
IR LED
IR LED
Total Internal Reflection
Plexiglass
IR Camera

Display Build

- 70 IR LED's mounted in aluminum bracket

- 22"x17" acrylic sheet
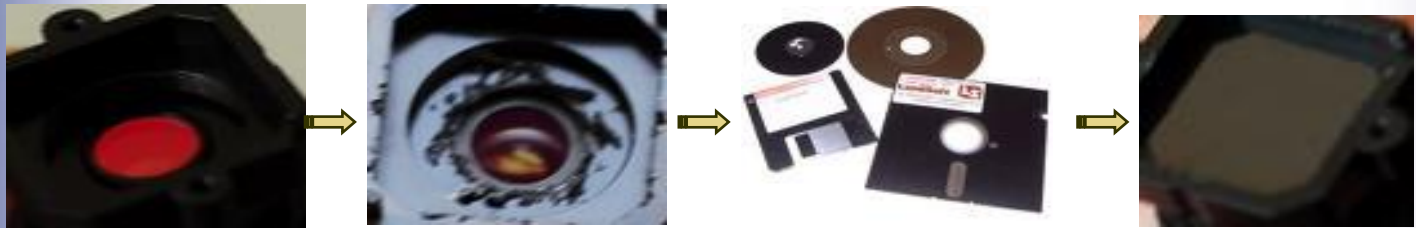
IR Camera

- GOALS
  - Min. Touch response delay
  - High touch accuracy
- REQUIREMENTS
  - minimum resolution of 640x480
  - frame rate of at least 30 FPS
  - transfer data at a minimum of 480 Mbits/s

Product Description	Manufacturer/Website	Price $	Procurement Status
Playstation 3 Eye, USB 2.0, 640 x 480 @ 60 fps ¦ 320 x 240 @ 120 fps	Dell	$34.99	AQUIRED
Unibrai Fire-I digital camera, IEEE 1394 (FireWire), 640 x 480 @ 30 fps	Office Depot	$104.99	
Xbox Live Vision Camera, USB 2.0, 640 X 480 @ 30 fps	Newegg.com	$24.99	
Logitech Quickcam Communicate STX Web camera, USB 2.0, 640 x 480 @ 30 fps	Newegg.com	$34.99	

IR Camera(cont)

- PROBLEM
  - Web cams record in visible light spectrum
  - We need camera to record in IR spectrum

- SOLUTION
  - Disassemble Web cam
  - Remove IR Filter
  - Install Bandpass Filter (Exposed Film)

IR Camera Build

- 3 different filters tested (Mag. Tape, CT film, Film Neg.)

- 2 pieces of Film Negative used

Temperature Control

- Requirement = compartment Temp < 90°F

- 2 Yate Loon Fans (120mm, 1650rpm, .3A)
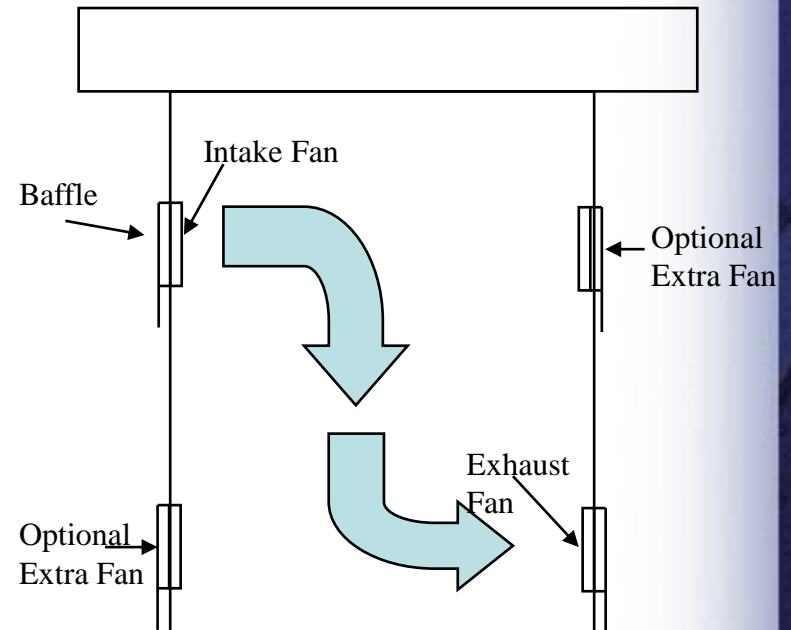
Temperature Control Build

- Originally grills were covered with speaker cloth

- Removed due to inadequate airflow

LED Power Supply

- 70 (100mW) IR LED's
- Using Ohm's Law and Voltage/Current divider
  - Total power dissipation = 10910 mW
  - Total current draw = 900 mA
- PS uses transformer, bridge rectifier, cap, voltage regulator, potentiometer
- Peak output of PS = 1.5A

Final Internal Layout

The image cannot be displayed. Your computer may not have enough memory to open the image, or the image may have been corrupted. Restart your computer, and then open the file again. If the red x still appears, you may have to delete the image and then insert it again.
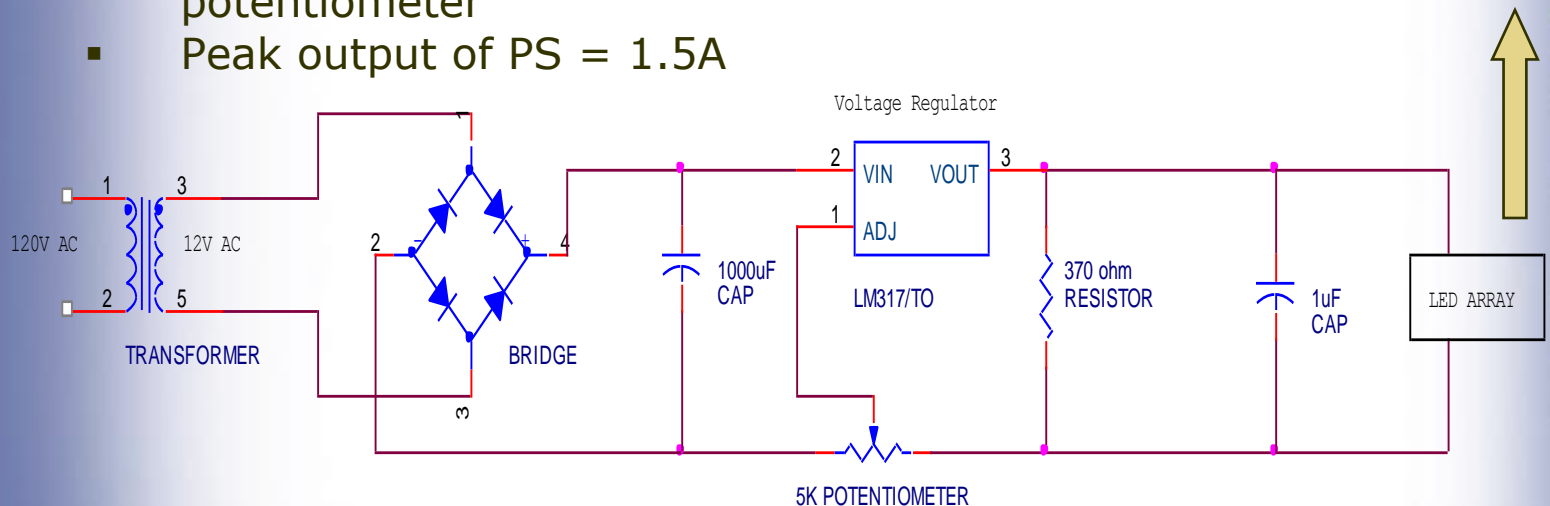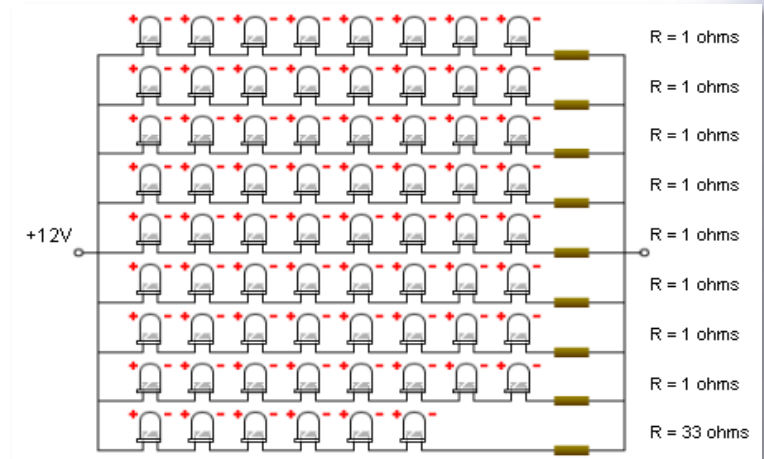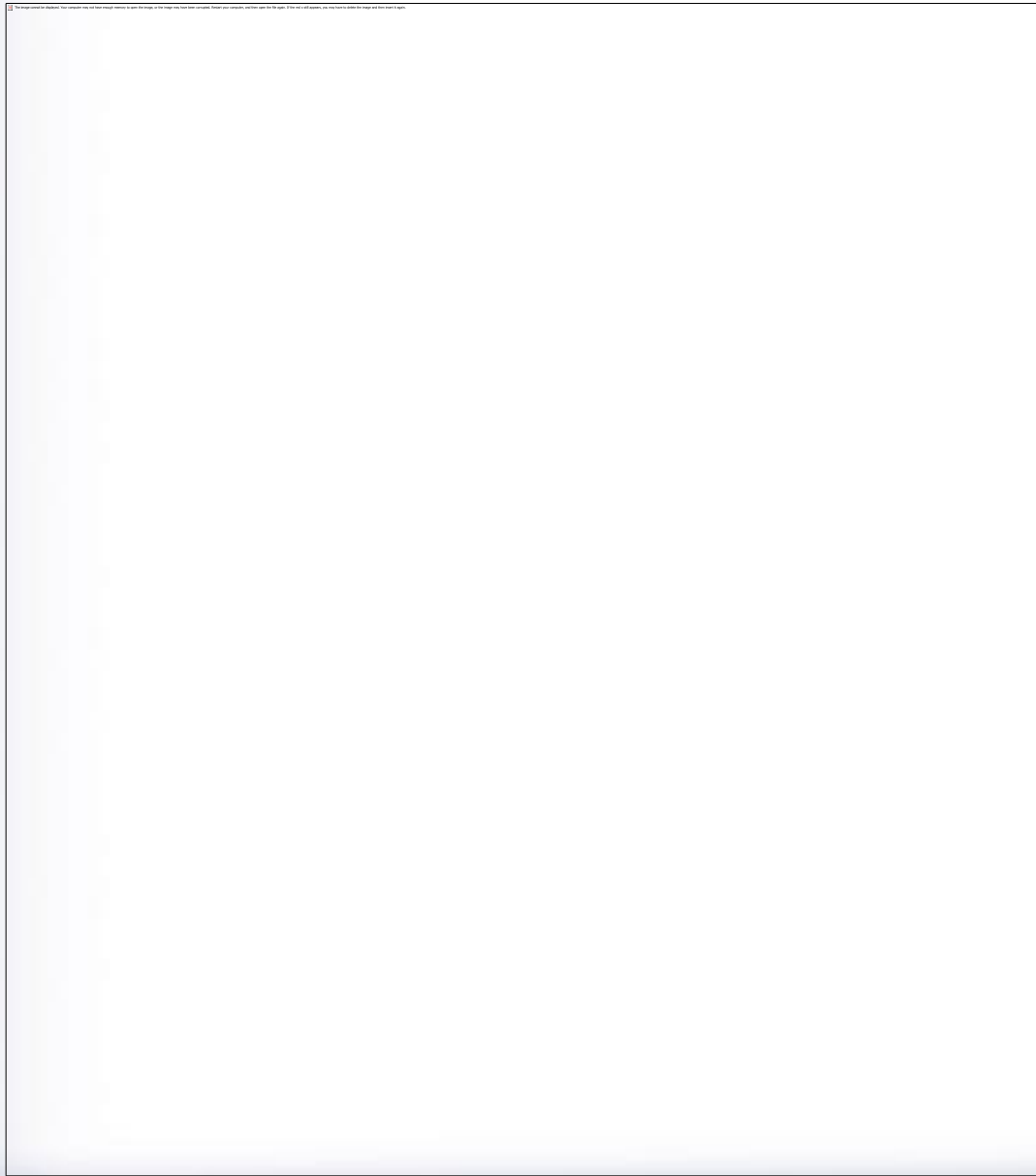
Software Requirements Specification Summary(updated)

- Poker Game
  - The MTPT will support Tournament style No Limit Texas Hold'em Poker.
  - The game supports four players.

- Restaurant Menu
  - The menu can be viewed anytime during the game
  - Only one menu will be visible

Software Requirements Specification(cont.)

- Table Interface
  - Chips will be displayed on the table
  - Community cards will be displayed on the center of table.
  - The interface will display the game information during the course of the game
  - The interface will player actions with touch commands

Software Block Diagram

- Input-Captured image from webcam

- Output-Graphics on touch screen
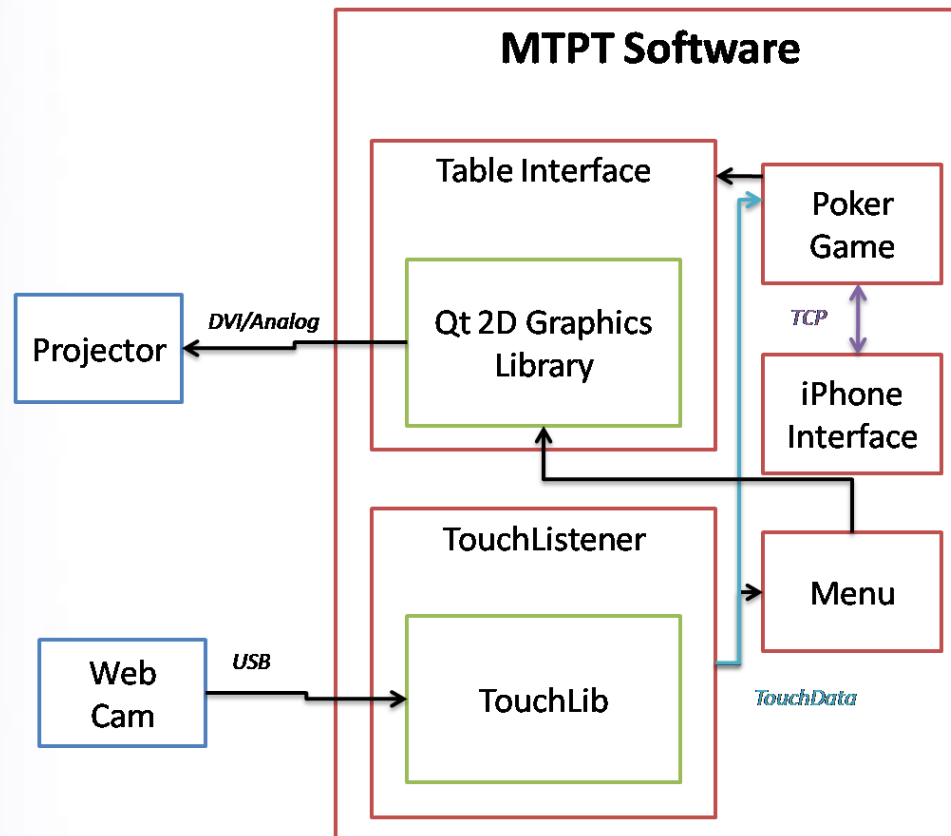
Software Libraries

- Qt (4.5.2)
  - Open Source C++ application and User Interface Framework
  - QGraphicsView, Threading, data types

- Touchlib
  - Processes input video from web cam
  - Sends touch events to application
    - E.g. "Finger Up, Finger Down"

- PokerSource
  - Poker card/hand representation
    - 64-bit Hand Mask
  - Poker hand evaluation

PokerExec Class

- State Machine
  - Used to control game state
  - Dependent on player actions
  - Runs per poker hand
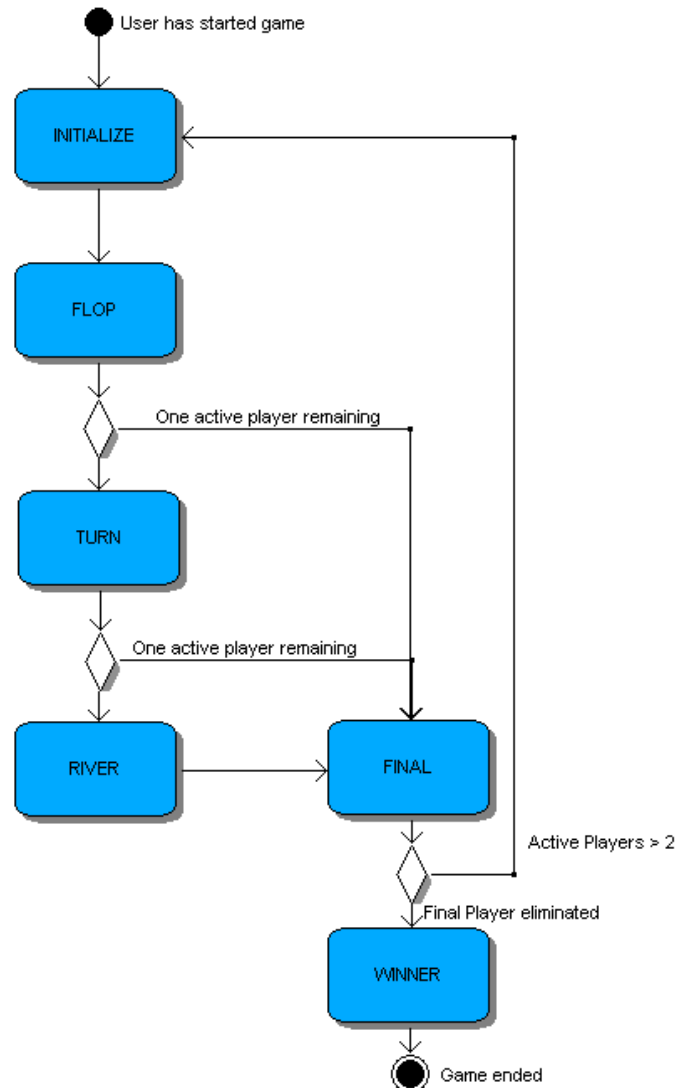
PokerPlayer Class

- Holds crucial game information for each player
  - Hole cards(value and graphics)
  - Total chips
  - Current Bet
  - State

- Four objects are implemented in PokerExec as an array of PokerPlayers
  - Index(0 to 3) will be used to keep track of positions

PokerPlayer

- holeCards : StdDeck_CardMask
- playerName : QString
- chips : uint
- currentBet : uint
- playerPosition : int
- action : bool
- active : bool
- eliminated : bool
- playerTcpSocket : QTcpSocket

+ PokerPlayer(in : void)
+ ClearPokerHands(in : void) : void
+ SetPlayerHoleCards(in cards : StdDeck_CardMask) : void
+ GetPlayerName(in : void) : QString
+ GetPlayerPosition(in : void) : int
+ GetHandValue(in communityCards : StdDeck_CardMask) : HandVal
+ IsActive(in : void) : bool
+ IsEliminated(in : void) : bool
+ SetActive(in active : bool) : void
+ AddChips(in add : int) : void
+ AddCurrentBet(in addAmount : uint) : void
+ GetCurrentBet(in : void) : uint
+ SetCurrentBet(in betAmount : uint) : void
+ SetPlayerRead() : void
+ Bet(in amountToCall : int, in bigBlind : uint) : uint
- start() : void
- SetPlayerName(in name : QString) : void
- SetPlayerWrite() : void

PokerCardGraphicsItem Class

- Originally planned to have card graphics as QGraphicsSvgItems(used on top of PokerSource types)
  - Particular can be called from the cards.svg file by passing a string(e.g. "queen_hearts")
  - Needed a better way to implement animations

- Main purpose of class is to animate dealt cards
  - player hole cards, community cards

| QGraphicsSvgItem | | TouchObject | |

PokerCardGraphicsItem

+ PokerCardGraphicsItem(QGraphicsItem * parent, QSvgRenderer * renderer, qreal posX, qreal posY)
+ ~PokerCardGraphicsItem(void)
+ QRectF boundingRect()
+ void TouchMoveEvent(TouchData data)
+ void HideCard(void)
+ void ShowCard(void)
+ void Deal(int direction)
+ void ShowValue(void)
+ void HideValue(void)
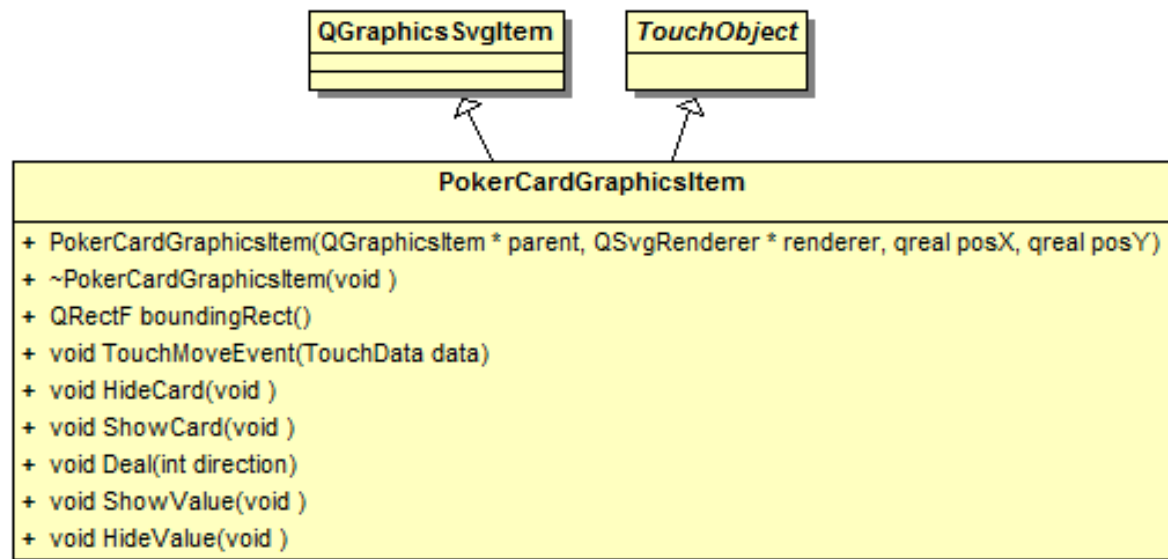
TouchListener Class

- Implements Touchlib to handle touch events
  - fingerDown,Update,Up are virtual functions of ITouchListener
  - TouchData is passed to classes that implement the callback class TouchObject
  - TouchObject*'s are registered using RegisterTouchObject() function
  - Touches were handled in the fingerUpdate function as this was the most frequent function called during testing

TouchListener Implementation

- Objects that wish to receive touch events inherit TouchObject interface class

  - One member, QRectF touchArea and pure virtual function TouchMoveEvent()

- TouchListener checks all registered TouchObjects and if the current touch is in bounds, the callback is called

TouchObject
+ QRectF touchArea
+ void TouchMoveEvent(TouchData data)

Software Development

- MTPT Software was developed using waterfall with prototyping after requirements and initial design in Senior Design I

- Prototyping was done throughout because of user interaction and unchartered territories(QGraphicsView and Touchlib)

- Software was developed in C++ using Visual Studio 2008

MTPT Interface
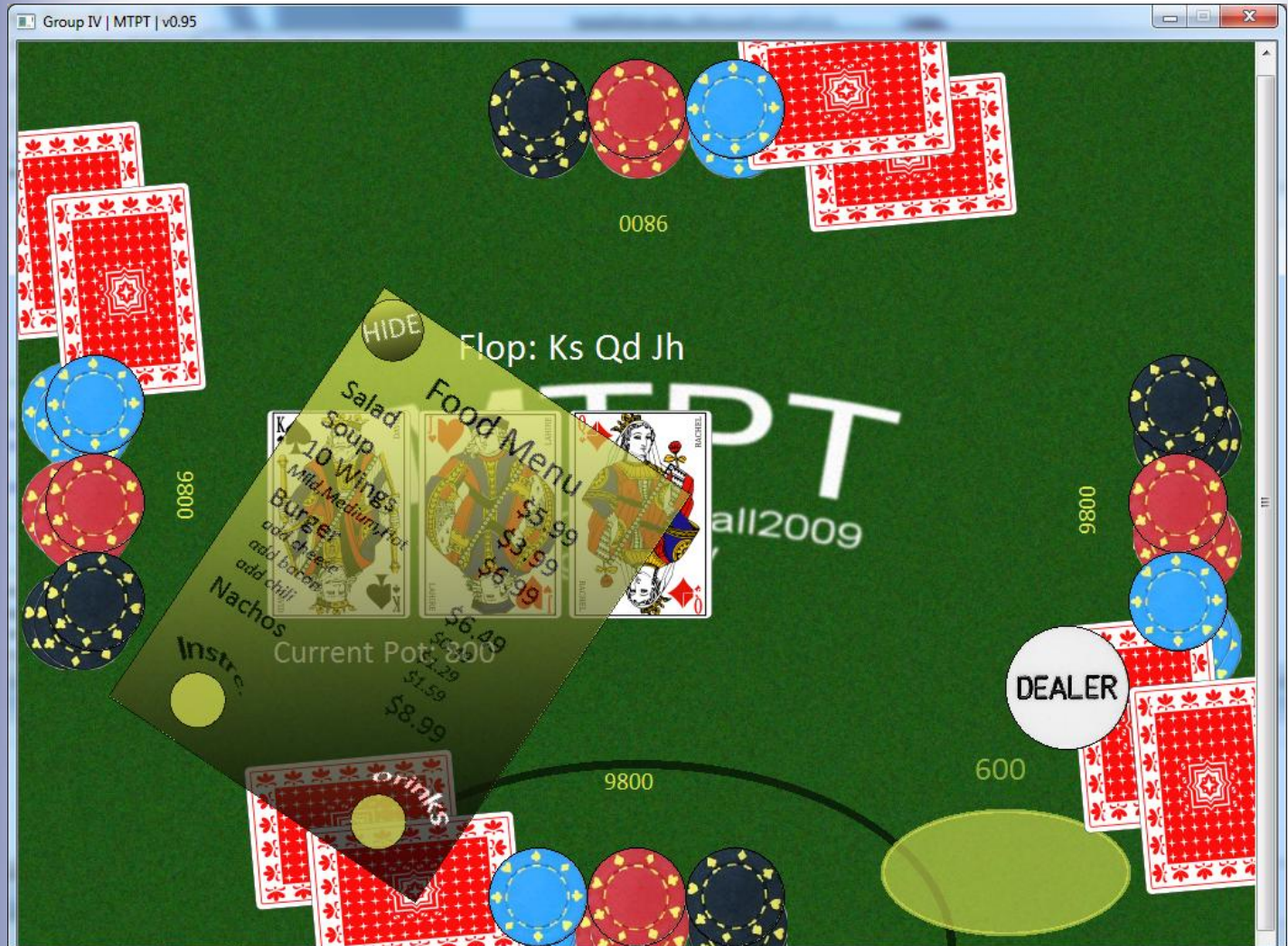
Player Interface

- Black outline is visible when the player is set to act

- Initial design wished the user to manually drag chips into yellow region, difficulty caused group to 'flick' chips to increment current bet

- Cards can be 'peeked' by touching

Software Issues

- Threading issues when hiding and showing QGraphicsItems manually
  - Solved by using Qt's SLOTS and SIGNALS

- Finding complete binaries of Touchlib that include both .dll and .lib
  - Older version was used

- Debugging with Touchlib
  - Web cam would not start process in Visual Studio debugger
  - If this were possible, group may have had better success in Touchlib integration
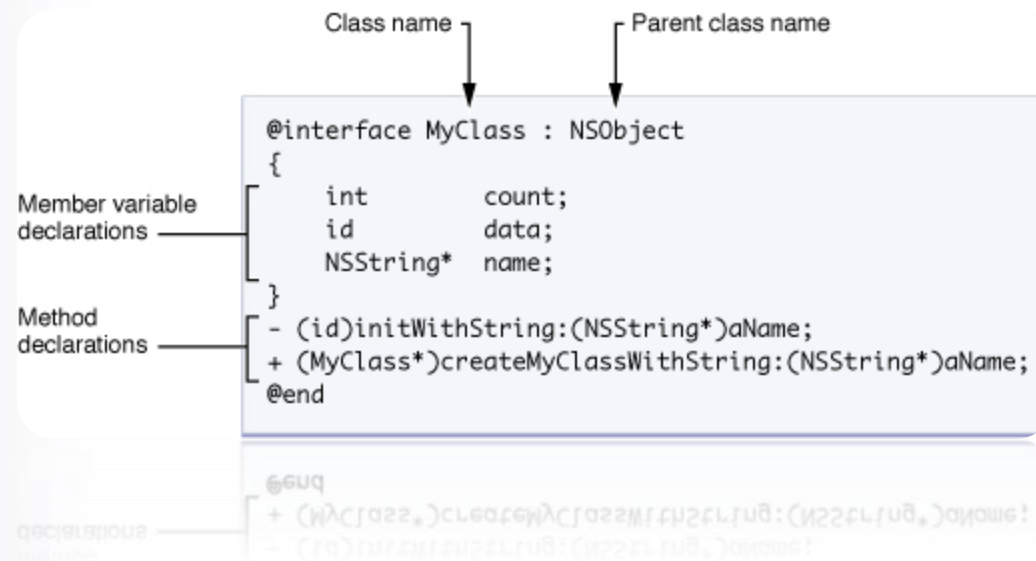
iPhone Development

- Apple OS X 10.6 Snow Leopard

- XCode 3.1

- Objective-C 2.0

XCode Development

Objective-C 2.0

- Superset of ANSI C

- Adds objects and messaging

Communications

- Wireless
  - Router with DHCP
  - Not Ad-Hoc

- Connection Daemon

- Features
  - Location-based connection

iPhone Interface

iPhone Data Entry

Budget

- $198 over budget due to projector

- Price per 100 units = $69k

	QUANTITY	UNIT PRICE	TOTAL PRICE	BUDGET	OVER/UNDER
MULTI-TOUCH SCREEN					
DAP Silicione, 10.1fl oz.	1	$4.50	$4.50		
Rosco 20"x24"	1	$12.00	$12.00		
Acrylic Sheet, 17"x22"	1	$63.90	$63.90		
SUBTOTAL			$80.40	$100.00	-$19.60
LED					
IR LED, 880nm	80	$0.47	$37.60		
SUBTOTAL			$37.60	$50.00	-$12.40
INTAKE/EXHAUST FAN					
Yate Loon	2	$4.99	$9.98		
SUBTOTAL			$9.98	$10.00	-$0.02
IR CAMERA					
Playstation 3 Eye	1	$34.99	$34.99		
SUBTOTAL			$34.99	$40.00	-$5.01
POWER SUPPLY					
LED Power Supply Parts	1	$5.00	$5.00		
Perf Board	1	$5.00	$5.00		
Soder	1	$3.00	$3.00		
Power Cord Extention	1	$10.00	$10.00		
SUBTOTAL			$23.00	$10.00	$13.00
PROJECTOR					
Mirror	1	$5.00	$5.00		
Projector Mount	1	$25.00	$25.00		
Optima EP719	1	$300.00	$300.00		
SUBTOTAL			$330.00	$150.00	$180.00
FRAMEWORK					
Birchwood 4'x8'	1	$39.00	$39.00		
Wood Glue	1	$3.00	$3.00		
Pine 1"x6"	2	$19.25	$38.50		
Nails	1	$2.00	$2.00		
Screws	1	$2.00	$2.00		
Wheels	4	$2.00	$8.00		
SUBTOTAL			$92.50	$50.00	$42.50
		TOTAL	**$688.87**	**$410.00**	**$198.47**
			x100		
		Cost of 100 Units	$68,887.00		

MTPT Demonstration