MICHIGAN STATE UNIVERSITY

Project Plan
BAPS: Battle Aircraft Position Share

The Capstone Experience

Team Boeing

Steven Garske
Andrew Kos
Eric Muller
Adam Cook

Department of Computer Science and Engineering
Michigan State University

Spring 2011

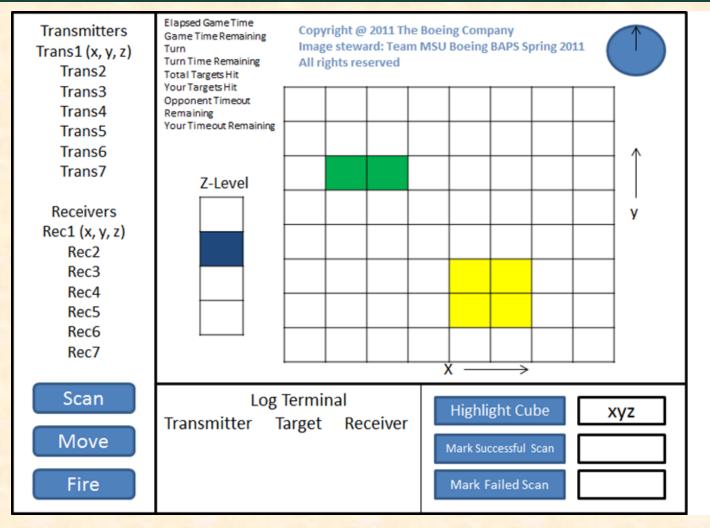*From Students…*
    …to Professionals

Project Overview

- Jayson T. Vincent out of St. Louis Missouri

- Combination of chess, battleship, and billiards

- 4 platforms for each player

- 7 transmitters, 7 receivers for each player

- Radar scanning to find enemy targets

- Firing is turn based

- Weather and shape of target effect scans

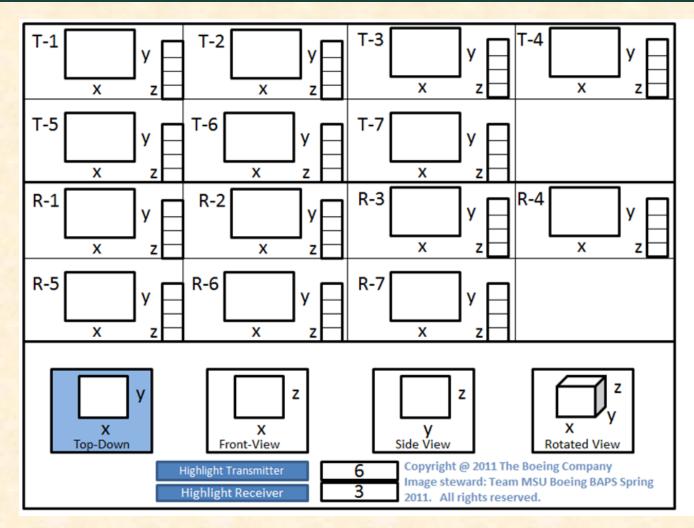# Functional Specifications

- Manager
  - Graphical component
  - Networking component
  - Scoreboard component
- Client
- Game logic
  - Firing
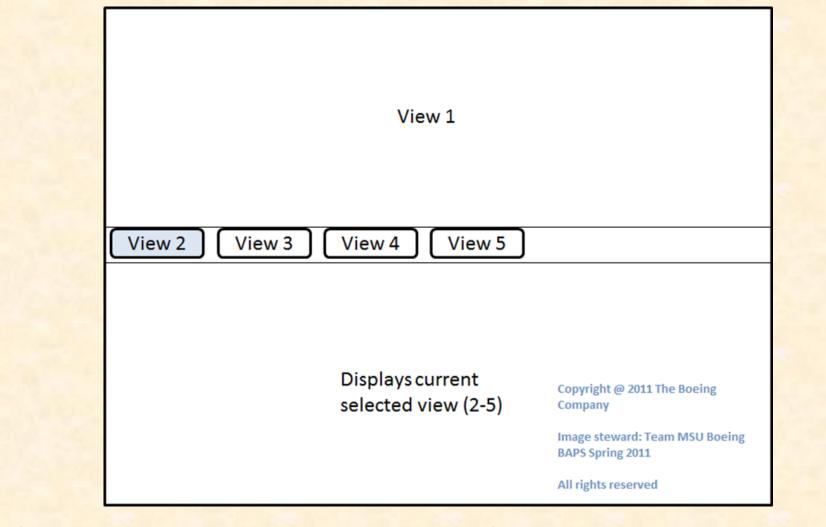 - Moving
  - Scanning

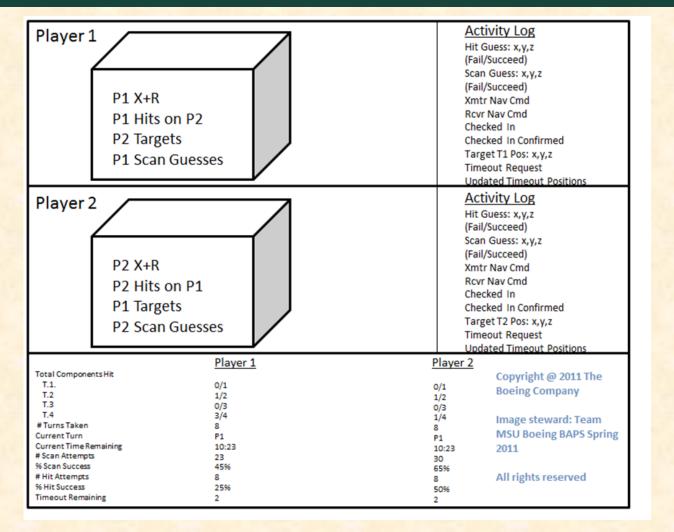# Design Specifications

- Client screens
  - Game board view
  - Help screen
  - Status view
- Manager screen
  - Client views
  - Game statistics
- Clients register with manager
- Manager controls game variables, data, logs

Screen Mockups

Screen Mockups

Screen Mockups

Screen Mockups

	Player 1		Activity Log

Player 1

P1 X+R
P1 Hits on P2
P2 Targets
P1 Scan Guesses

Activity Log
Hit Guess: x,y,z
(Fail/Succeed)
Scan Guess: x,y,z
(Fail/Succeed)
Xmtr Nav Cmd
Rcvr Nav Cmd
Checked In
Checked In Confirmed
Target T1 Pos: x,y,z
Timeout Request
Updated Timeout Positions

Player 2

P2 X+R
P2 Hits on P1
P1 Targets
P2 Scan Guesses

Activity Log
Hit Guess: x,y,z
(Fail/Succeed)
Scan Guess: x,y,z
(Fail/Succeed)
Xmtr Nav Cmd
Rcvr Nav Cmd
Checked In
Checked In Confirmed
Target T2 Pos: x,y,z
Timeout Request
Updated Timeout Positions

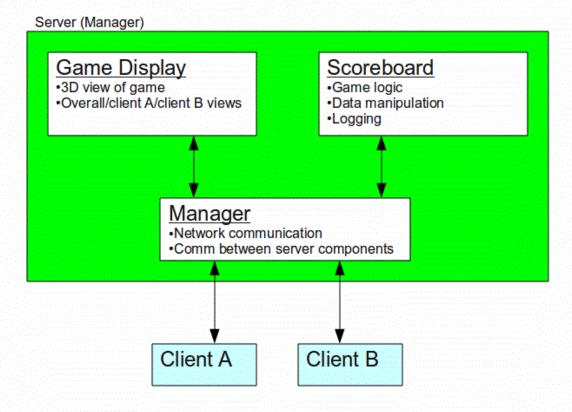|	Player 1	Player 2
Total Components Hit		
T.1.	0/1	0/1
T.2	1/2	1/2
T.3	0/3	0/3
T.4	3/4	1/4
# Turns Taken	8	8
Current Turn	P1	P1
Current Time Remaining	10:23	10:23
# Scan Attempts	23	30
% Scan Success	45%	65%
# Hit Attempts	8	8
% Hit Success	25%	50%
Timeout Remaining	2	2

Technical Specifications

- Shared memory with DLL

- Open Scene Graph

- Network utilizes TCP/IP

- Threading with sockets

- Data model shared between clients and server

- GUI renders from `Environment`

System Architecture

Server Structure and Basic Communication

Server (Manager)

Game Display
- 3D view of game
- Overall/client A/client B views

Scoreboard
- Game logic
- Data manipulation
- Logging

Manager
- Network communication
- Comm between server components

Client A

Client B

System Components

- Hardware Platforms
  - Windows XP
  - Windows 7
- Software Platforms / Technologies
  - Open Scene Graph – to display 3D graphics
  - Windows sockets
  - Windows CryptoAPI for network encryption
  - Blender for 3D modeling
  - Visual C++

Testing

- Test build
  - Separate repository branch
  - Testing done before integrating with master build
- Manual testing
- Automated test suite
  - Existing unit testing framework
  - Custom script

Risks

- Open Scene Graph
  - GUI could be difficult

- Encryption
  - Not 100% sure what lib to use yet

- Testing and Game Balance
  - Features are not completely concrete yet

- Integration
  - Connecting graphical components with backend