From Students…
…to Professionals

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
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

The Capstone Experience

Team Boeing Project Plan
Screen Mockups
Screen Mockups

**Player 1**
- P1 X+R
- P1 Hits on P2
- P2 Targets
- P1 Scan Guesses

**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 T1 Pos: x,y,z
- Timeout Request
- Updated Timeout Positions

**Total Components Hit**
- T1: 0/1
- T2: 1/2
- T3: 0/3
- T4: 3/4

**# Turns Taken**
- Player 1: 8
- Player 2: 10

**Current Turn**
- Player 1: P1
- Player 2: P2

**Current Time Remaining**
- Player 1: 10:23
- Player 2: 10:23

**# Scan Attempts**
- Player 1: 100%
- Player 2: 100%

**% Scan Success**
- Player 1: 45%
- Player 2: 50%

**% Hit Attempts**
- Player 1: 8
- Player 2: 8

**% Hit Success**
- Player 1: 25%
- Player 2: 50%

**Timeout Remaining**
- Player 1: 2
- Player 2: 2

**Copyright © 2011 The Boeing Company**

**Image steward: Team MSU Boeing BAPS Spring 2011**

**All rights reserved**
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

- **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