



Casino - RFC

Stage 1: Component Testing & Creation

Stage 2 : Integration

Stage 3: Networking - Database

Outline – Casino BJ Stage 1



- Assignment Objectives
- H1 Shoe Class
 - Shoe() // 5 Decks Default
 - Shoe(int nDecks) //nDecks 1..7
 - Shoe(int nDecks, boolean hasMarker)
 - Where has Marker means set a marker 75%-85% in shoe
 - Deal a -99 when you hit the Marker
 - Dealer will call reshuffle
- Assignment H2 Card or CardValue Class
- Assignment H3 13.9 Page 444 Modified

Assignment Objectives



- Assignment H1
 - Shoe : Where Dealer Deals Cards From
 - Default is 5 Decks, constructor allows 1..7 Decks
- Assignment H2
 - Add Card/Hand Value Class, use shoe, output value/suite
- Assignment H3 13.9 Page 444 Modified
 - Java Web Start Application
 - But Display value and suite Under Card
- Group Assignment Pick one of Three Games
 - Blackjack, video Poker or Texas Hold-em
 - Later Network Multi-user w DB Interface

H1 Shoe/Card Critical Specs



- Cards int Value 1..52,
53 Black Joker, 54 Colored-Red Joker – But No Jokers
- $(Card-1)/13 =$ Suit: Spades, Hearts, Diamond, Clubs
 - Reverse Lexicographic Order
- $(Card-1)\%13 =$ Ace, 2..10, Jack, Queen, King
- One Deck Shoe has 52 Cards, values 1..52
- Card Images @ <http://cim.saddleback.edu/casino/card/1.png>
 - Where 1.png is the Ace of Spades
 - 15.png is ($14\%13$) Two of Hearts (note $15\%13$ is 2)
- Two Decks – int Array with 104 Integers...
 - 2 Ones, 2 twos .. 2 52's
- Default Shoe 5 Decks

Shoe Class Considerations

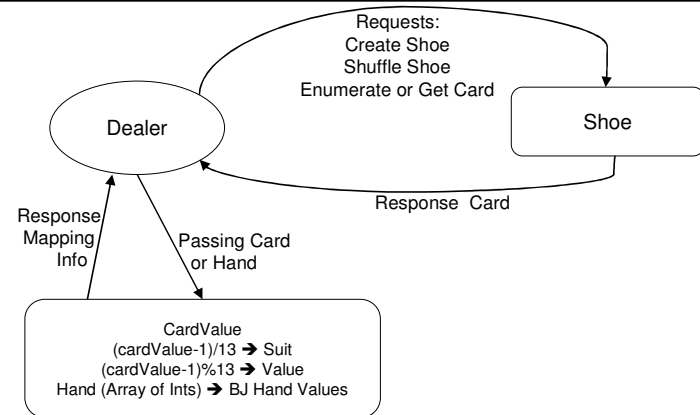


- Dealer Easy Access One Card at a Time
- Proposed Shoe Class
 - Need Method to Shuffle Cards
 - Could Implement enumerator or iterator
 - Reshuffle whenever new Enumeration is done
- Typical Shoe class Constructors
 - One Sets the Number of Decks
 - Second Sets Number of Decks & Initializes Marker

© 2/09 & 2/2010 Professor T. DeDonno

5

Assignment H2 Automata



© 2/09 & 2/2010 Professor T. DeDonno

6

Refactoring



- Alter Part w/out Modifying Behavior
- Improve Parts w/out changing results
- Refactoring Can Cause Major Problems
 - 10K Line Program Need to Improve One Class
 - Can I alter one Class & Maintain Program Integrity?
- Methodology to Verify Class Changes
 - Main testing (main function for each class) or
 - JUnit Testing

© 2/09 & 2/2010 Professor T. DeDonno

7

JUnit Testing

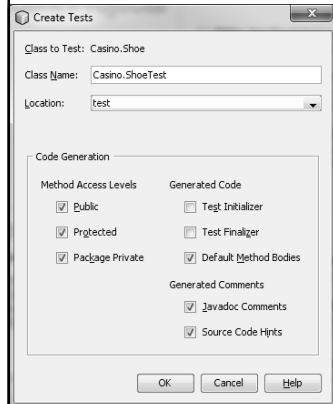


- JUnit – Java Unit Testing
 - Test Class/Methods Independent System
- Test Folder with Test Classes
 - Test Class Has Method to Test Each Method
 - Junit Tests both Shoe and CardValue
- Junit testing built into Netbeans & Eclipse
- junit.org (Main Site Has API)
Chapter on Junit...
- http://java.sun.com/developer/Books/javaprogramming/ant/ant_chap04.pdf

© 2/09 & 2/2010 Professor T. DeDonno

8

NetBeans JUnit Testing



- Select Class File
- Right^Click Tools → Create Junit Tests
- Test Initializer Setup Methods to Run Before Class Creation and Before Each Method
- Creates ShoeTest.java in test sub-Folder

9

@ annotation



- Provide Program Data not Part of Program
- `@Override` //Annotation overriding a method
`public String toString() {...}`
- `@Test`
Runtime Annotation org.junit.Test Executes this Method
drop `@Test` to Skip Testing
- `@Before`
Methods Executed Before Each Test Method
- `@BeforeClass` done Once at Start
 - Expensive Data Resource (Database)

© 2/09 & 2/2010 Professor T. DeDonno

10

org.junit.Assert



- Not Part of Standard J2SE junit.org Site
- Throw `assertError` on not Equals
- Exception Message is Optional 1st Argument
- Static `assertArrayEquals(expected, actual)`
- Static `assertEquals([msg,] expected, actual)`
 - For Double/Float delta Value
- Static `fail("String"); //Fail Test`
 - Drop Fail once you write test Method

© 2/09 & 2/2010 Professor T. DeDonno

11

Creating JUnit Tests

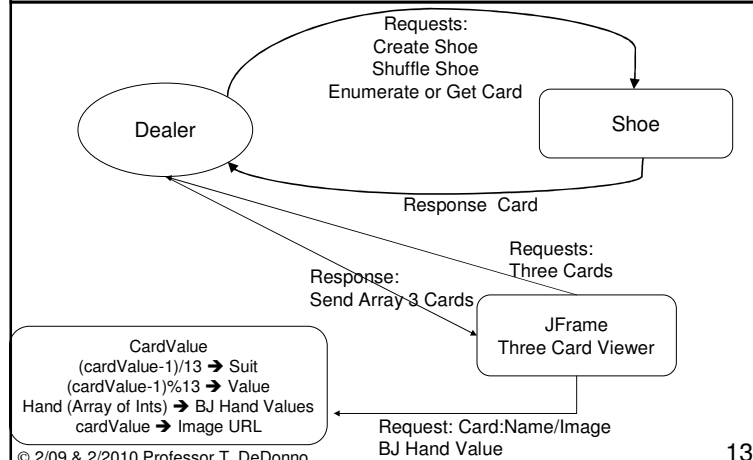


- Add Code to each `@Test` Method
- Make sure you know Answer
- Assert Method Verify Output
- Run File in Test Folder
or Test File in SRC Folder
- Run → Test Project
 - Runs all JUnit Tests

© 2/09 & 2/2010 Professor T. DeDonno

12

Assignment H3 Automata Add Card Viewer & Dealer Interface



H3 – Card Viewer



- Deal Three Cards Images
 - Map Card 1 to 1.png
 - new URL(<http://cim.saddleback.edu/casino/card/1.png>);
 - Use Label(icon ImageIcon)
 - Where ImageIcon From URL
- Print out the BJ Values of the Hand
- Add Button to Deal next three Cards
- You may add Other GUI Components

© 2/09 & 2/2010 Professor T. DeDonno

14

H4 Home Page



- ftp filezilla Video on Blackboard
- Create index.html put to public_html
- Implement Java Applet from Applet Chapter
- Make h3 a Java Web Start
- Make sure home has both applet and jnlp link

© 2/09 & 2/2010 Professor T. DeDonno

15

Assignment H5

Create
Either BJ, Video Poker or Texas Hold EM

© 2/09 & 2/2010 Professor T. DeDonno

16

XML Format



- `<gamePlayers>`
`<gameName>bj</gameName>`
- `<round>#Rounds</round>`
- `<player>`
 - `<name>PlayersUniqueName</name>`
 - `<hand>0 13 20</hand>`
 - `<bet>20</bet>`
 - `<serverRequest>Hit?</serverRequest>`
 - `<playerResponse>Stand</PlayerResponse>`
 - `</player>`
- Stats in player? IP in player?
- Face Down Card is 0

Java XML



- Create Java Class Represents Nodes
 - Properties XML Nodes
 - XML Can Require Certain Events
- XML is a Tree Structure
- Use Java Parser XML → Java Node Class

- J2SE5 `Javax.xml`.
- J2EE `javax.xml.parsers`

Java API for XML Processing (JAXP)



- DOM (Document Object Models)
 - Build document object in memory
 - Used on a DOM Page
 - Database → XML → Display a Web Page
- SAX Simple API for XML
 - Serial Data Stream (Socket R/W)
 - Handle Content Trigger Action
 - Dealer Request Hit or Stand

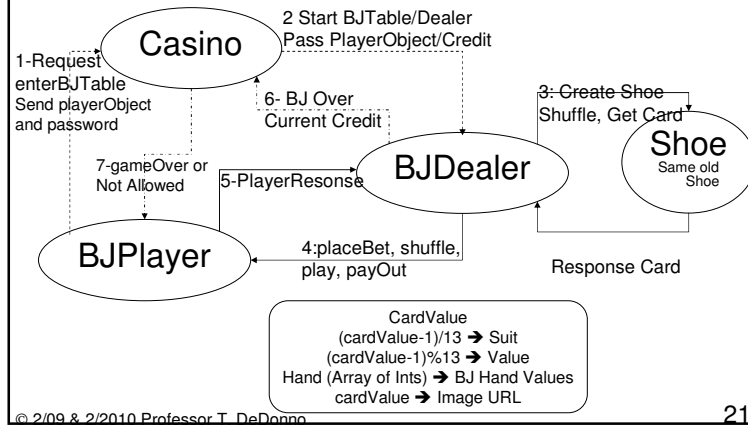
JAXP API



- Part of J2EE
- `Javax.xml.parsers`

- Two Classes
 - `DocumentBuilder`
 - `SAXParser`

H2 Automata



© 2/09 & 2/2010 Professor T. DeDonno

21

Casino Class



- Derived Class of CasinoBaseClass
 - Abstract Class CasinoBaseClass
 - abstract boolean enterBJTable(BJPlayer player, String password)
 - Player Object calls enterBJTable,
 - Returns false To User - not allowed on Table
 - Allowed pass Credit & BJPlayer object To Dealer
- Manages the Dealers, Players
 - Initially Hard code username, password & credit
- Stage Three Networked Based Casino
 - Database of Username/Password/Credit
 - Will Play Other Students

© 2/09 & 2/2010 Professor T. DeDonno

22

Dealer Class



- Instantiated by Casino and Receives PlayerObject w Credit
- Casino's Dealer Sets Specific Pay outs
 - Hit soft 17, what to pay on bj21 & 21, 5 Card Charlie
- Initially 1-Deck Shoe, Dealt to Last Card
- Hit less 17; Stand on 17
- Initiates game by calling BJPlayer Abstract methods

© 2/09 & 2/2010 Professor T. DeDonno

23

abstract class BJPlayer



- abstract String getName()
- abstract int placeBet(final int credit)
 - return 0 - leaveTable, call enterTable to start again
- abstract void shuffle(final int decks, final float marker)
 - Inform Player of a Reshuffle
 - marker 80.0 80% (Complete round)
- abstract void newPlayer(String playerName);
 - //playerLeave Inform Player of Table Change
- Initial Players just return on shuffle and newPlayer
 - A Future card counter player needs to know about shuffle
- abstract void payOut(final int pot, final int yourHand, final Hand hands[]);
 - //pot may become an array;
 - Pot is negative for loss; positive for win

© 2/09 & 2/2010 Professor T. DeDonno

24

BJPlayer Game Play Methods



- abstract void payOut(
final int pot, final int yourHand, final Hand hands[])
 - Dealer is hands[0]
 - you are hands[yourHand], int vs Stack?
 - Each table has 7 Hands;
 - empty chair hands[] is empty
 - On play You See Only Dealers 2nd Card,
 - All player cards are Face Up
- abstract int play(Same as payOut)
 - int: enum type hit=0, hold=1, doubleDown=2, split=3
 - Don't implement Splits
 - Your Casino has only one player

Operations Summary



- Create Casino Object
- Create Player Extends BJPlayer
- Call enterBJTable(playerObject, String psswd)
 - Casino has enterBJTable Method
 - Casino calls Dealer Passes playerObject & Credit
 - Player can Only be at One Table at a Time
- Dealer Controls Table/Game
 - Game Terminates...
 - Bankrupt Player or placeBet of Player returns 0
 - Dealer returns Player Credit to Casino

OOP - Elements



- New Player and Casino
 - Polymorphism
 - Based on a BJPlayer and CasinoBaseClass
- Dynamic Binding
 - At Runtime determine Call
 - enterBJTable
 - Dynamic binding to a Casino
 - Dealer class Dynamically Binds to BJPlayer

Assignment Goals



Initial Models – Stand Alones



- Write Program that runs 1000+ times
- H1 is a Shoe
 - Standard Alone Java Application
- H2 JFrame
 - Deal Three Cards
 - Verifies Shoe and Ability to Calculate Hands
- H3 Stand alone BJ Program

© 2/09 & 2/2010 Professor T. DeDonno

29

Interface – View/Controller



- H4 Create Your Two CIM Home Pages
- H5 BlackJack Java Applet
- H6 Java Web Start
- H7 Form Casino Teams

© 2/09 & 2/2010 Professor T. DeDonno

30

Casino Stage II

Form Teams (2-5 Students/Team)
Web Interface Video Poker & BJ Casino
Database Interface
BJ Agent – Computer Can Play BJ

© 2/09 & 2/2010 Professor T. DeDonno

31

Team Tasks



- Critical Tasks (2-3 People)
 - Web Interface for VideoPokerGame and BJGame
 - Web Interface to Casino Database
 - BJ Agent Player
- Will consider allowing larger teams...
 - Write an additional Casino Dealer and Player
Either Texas Hold-em or Draw Poker Player
- QA: Quality Assurance Engineer: JUnit Testing

© 2/09 & 2/2010 Professor T. DeDonno

32

Video Poker Game



- Interactive Game not a Player
- Web User gets 5 cards, can exchange 4
- Two pair of Jack or better you win
- One Deck Shoe – Shuffle After Each Hand

- Card Images: Ace of Spades is Value 1:
<http://cim.saddleback.edu/casino/card/1.png>

- Compute Hand and Payouts...

Hands and Payouts



- Flush same suit $(\text{value}-1)/13$
- Pair or better same values $(\text{value}-1)\% 13$
- Straight: sorted Sequence value range 0..12
- Note other hands are just combinations
 - Full House is Three of a Kind + Pair
 - Straight Flush Flush + Straight

- Info on Rankings and Video Game Payouts....
- http://en.wikipedia.org/wiki/Hand_rankings#Flush
- <http://vegasclick.com/games/videopoker.html>

Database



Database Scheme
casino.sql Query Comments

Database



- Main Casino Database will have
 - Active Casino URL, Username
 - Database SBUsername, password, credit
 - Tables: player, agent, casino, game, table, playerTable, casinoGame

- Local Casino have Guest Accounts
 - Play with Local Casino House Money

Casino Database




Table: player				
playerID	name	credit	password	email

Table: agent					
agentID	gameID	playerID	Version	Java className	creator


Table: game				Table:casinoGame	
gameID	name	description	Rules payout	casinoID	gameID


Table: casino					
casinoID	name	url	password	playerID	Description


Table: gamePlayer - messageServer (table lounge)					
gameTableID	playerID	net gain money	rounds	Gameswon	clientIP

Table: gameTable				
gameTableID	gameID	casinoID	startTime	endTime NULL ActiveGames

© 2/09 & 2/2010 Professor T. DeDonno 37

- ## Casino Database
- 
- <http://cim.saddleback.edu/~premiere>
 - Links to Assignment, Database Schema
 - All Students Must Individually complete DB Assignments H8-h10
 - Each Teams must have Casino DB Interface
 - Join Queries have the most meaning
 - List of names and IP for players at active BJ Table
 - Name of most profitable Casino Game
 - This Team Player should have CIM205a or equivalent
- © 2/09 & 2/2010 Professor T. DeDonno 38

- ## Local Guest Database Not implement Yet
- 
- Local Casino Guest Pass
UserName, Password, netWinnings and Email
 - Guest uses Casino Applet House Money
 - Connect to Server, using Applet Casino Player
 - Guest Plays as Applet Casino Player
 - At Game Over,
 - Central Updates House Money
 - Local casino update DB netWinnings
- © 2/09 & 2/2010 Professor T. DeDonno 39

- ## Assignment: casinoCS4B.sql
- 
- 1) Display table names in Casino:Database
 - 2) Display URL of all Casinos
 - 3) Display name and credit for all players, with highest
 - 4) Display each row for all active game tables
 - 5) Display length as hours (compute fractional) all completed game tables.
 - 6) Display each rows for all Games
 - 7) Display gameID for VideoPoker
 - 8) Display player info for all bankrupt players
 - 9) How many software agents do we have.
 - 10) What is the total credit of all casino players.
 - 11) Agents id, creator and Java Class Name
 - 12) List playerID, and net Money Gain for each gamePlayer (group by clause)
 - 13) List ID, and number of different games for each casino
 - 14) current number of different types of games supported by casino
 - 15) Display all player info sorted by name in lexicographical order
 - 16) Display player info of with highest credit limit
 - 17) Display playerID for all players who played in gameTableID 2002
 - 18) insert gameID=4, name=TexasHoldEM description=Texas rules=standard
 - 19) Display contents of table:game
 - 20) Describe field names and types for table:game
- © 2/09 & 2/2010 Professor T. DeDonno 40

Assignment: fruitCS4B.sql



- 1) Display contents of fruit Table by increasing price
- 2) Display name and price of most expensive fruit
- 3) Display all inventory records
- 4) Display total quantity as total for fruitID of 1
- 5) Display name, price for fruitID of 2
- 6) insert cherimoya, price 3.50, fruitID=9
- 7) Display contents fruit table increasing fruitID order
- 8) Display number of fruits as FruitTypes ?
- 9) Display average fruit price as average
- 10) Display first three rows of inventory table

+3 Player Teams



- XML Data Stream, API, Data Source, RSS
 - Push or Get Data
- cURL Client URL – Encapsulate Site

- Sport Betting Data Source
- Weather
- User IP
- Horoscope (Local DB record Birth Date)
- Cookie Save User Name

Stage Three

Large Teams Only



Video Poker Stage 3 Consideration



- Registered User Submit Username/Password
- Casino Return Credit
- You Need to Update Credit a Game End
- These May be All JSP Files

- Note VideoPoker has Abstract Calls
 - That Can be Human and/or Agent

Large Team Task



1. Agent: BJ Player
2. Agent: Draw Poker Player
3. Agent: Texas Holdem Player
4. Dealer either Draw or Texas Holdem
5. Will Release CasinoBaseClass Additions..
 - enterDrawPokerTable(drawPokerPlayer op, String passwd)
 - enterTexasHoldemTable
6. Applet Local Casino Interface
 - Video Poker – VideoPokerPlayer
 - Display Casino Status Info
 - Possible Other Applet Tasks
 - Web Interface to Play BJ, Draw or Texas Holdem Agents
 - Consider Other Game:Roulette, Slot machine, Craps
7. Q/A Engineer Verify Integrity and Meet Specification
 - **Do It Right Then Better**
 - **Multiple Casino & Shoe Need only One**
 - **Check randomness of Shuffling Strategies**
8. **Database Guest User**

- Need Groups to write ...
 - drawPokerDealer and texasHoldemDealer
 - Must Meet texasHoldemPlayer & drawPokerPlayer interface

Draw Poker Player

Large Teams Only



Draw Poker Player



- Using Same Classes as Video Poker
 - PokerHandValue
- I'm supplying Abstract Classes
- Large Teams need to Write
 - Dealer (Supplying Dealer/Casino Shared Data)
 - Socket Mechanism as Blackjack

Abstract Class VideoPokerPlayer



- `int anteFirstBet(int credit)`
 - You deposit Only an Initial Bet
- `void Draw(int hand[])`
 - Exchange Cards,
 - to exchange card your Second Card set `hand[1]` Negative
- `void showDown(int bet, int credit, int hand[])`
 - Bet inegative for Loss, Else is Size of your Winnings
 - Credit Represent Your Current Credit
- Web User Interface Same Three methods
 - Wait for User Input

drawPoker – order of Play



- Reuse Video Poker Card Classes
- Ante First Bet (Set Value)
- Deal 5 Cards
- Place Bet (1 Round: Stay In, Fold, or Raise)
- Draw Phase
 - Exchange up to Three Cards
 - Unless You have Ace, exchange all but ace
- Place Bets (1 Round or till all hold)
- ShowDown

placeBet



- Not Sure
- One Round
 - Raise, Fold, or Hold
 - Cannot Raise once you Hold
 - Stay to all Hold or Fold
 - Or Don't Allow Raise

Abstract drawPoker Methods



- `int anteFirstBet(int credit)`
- `int Bet(int currentBet, int bettingChair, int pot
int yourChair, Stack hand[])`
- `void Draw(int pot, int yourChair, Stack hand[]
)`
- `void showDown(int pot, int winningChair,
int yourCredit, int yourChair,`

Draw Poker Methods Part 1



- drawPoker Player
 - c.enterDrawPoker(DrawPokerPlayer, String passwd)
- int anteFirstBet(int credit)
 - Return 0 to leave Table, Default is 1
- int Bet(int currentBet, int bettingChair, int pot, int yourChair, Stack hand[])
 - Dealer Deals 5 Cards to each Person @ table
 - currentBet is from bettingChair Player
 - When bettingChair = yourChair its your turn to bet
 - return -1 to Fold, or currentBet (to Stay in) or currentBet+more to raise on yourBet

Stage II Milestone



- Applets
 - All Local Casino will have Games: Craps, Video Poker, etc.
 - May Want each Game a Separate JPanel
- Player Agents Need to Work at Your Casino
 - Have Button to Test Player Agents, **Right then Better**
- Stage III Network Casino
- Best Bets at Breaking Even in Vegas
 - Beat Dealer 21 Face Card
 - Video Poker Basic Strategy
 - Hit or Crap Out in Craps

Draw Poker Methods Part 2



- void Draw(int pot, int yourChair, Stack hand[])
 - Your cards are Stack table[yourChair]
 - To Exchange a Card Set it to Negative Value
 - Exchange a 3 of Spades Value 3 to -3
- void showDown(int pot, int winningChair, int yourCredit, int yourChair, Stack hand[])
 - You Get to See all Cards;
 - Return credit of winningChair Player

Texas Hold-em



Texas Holdem – Order of Play



- http://en.wikipedia.org/wiki/Texas_hold_%27em
- Ante
- Pre-flop; Deal 2 Cards Face Down To Each Player
- Bet
- Flop: Deal 3 Face Up Community Cards
- Bet
- Turn: Deal 1 Face Up Community Card
- Bet
- River: Deal 1 Face Up Community Card
- Bet
- ShowDown

TexasHoldemPlayer Abstract Methods



- int anteFirstBet(int credit)
- int Bet(int currentBet, int bettingChair, int pot
int yourChair, Stack hand[])
- void preFlop(int pot, int yourChair, Stack hand[])
 - Also flop, turn, and river
- void showDown(int pot, int winningChair,
int yourCredit, int yourChair,
Stack hand[])

TexasHoldem Methods – Part 1



- TexasHoldemPlayer
 - c.enterTexasHoldem(TexasHoldemPlayer, String passwd)
- int anteFirstBet(int credit)
 - Return 0 to leave Table, Default is 1
 - Betting is blind and is Weird
- int Bet(int currentBet, int bettingChair,
int pot, int yourChair, Stack hand[])
 - currentBet is from bettingChair Player
 - When bettingChair = yourChair its your turn to bet
 - return -1 to Fold, or currentBet (to Stay in) or

TexasHoldem Dealing Cards



- Two down cards,
- Flop 3 Community, Turn +1, River +1 table[0][5]
- void preFlop(int pot, int yourChair, Stack hand[])
 - Community is table[0]
 - You see only Your Two Cards table[yourChair]
- flop(int pot, int yourChair, Stack hand[])
 - Turn and River
 - All that changes is table[0] is given face up cards
- void showDown(int pot, int winningChair,
int yourCredit, int yourChair,
Stack hand[])

Texas Hold-em



- 20 Hands
- Find Best Hand Use Video Poker Engine

Stage 3

Casino Web Architecture
Casino Network Interface

Outline Stage 3



1. Implement JSP
2. Consider JSF Viewer for Agent Players
3. Local Player Database
4. Network Interface – Central Server

Development Milestones



- Casino, BJPlayer, Card Class
- Add Video Poker
- Add Craps, Add Poker Player
- Add Texas Hold'em Player
- Thread Interface
- Socket Pass XML Data
- Add Database



Casino Web Architecture

Web Architecture
Thread/Socket Interface
MVC



Web Architecture

- Casino Service Listens on Port 8000
 - Player Open Socket,
 - Sends Message
 - Responds with JSP File Output or Sends Socket to a Dealer
- Read/Write Serialized Objects
 - Message (Play, payOut) and Serialized Parameters



Casino – Dealer Thread

- Casino no Player at Table,
 - Casino starts dealer Thread
- Synchronous Stack of PlayerObjects w Fields...
 - entering, at or exiting Table
 - serverSocket connection
 - PlayerObject w current credit
 - Casino will Remove exiting Players
- New Player Wants to enter an Active Table
 - Casino Pushes PlayerObjects
- At end of Each hand, Dealer Checks Synchronous Stack
 - Add Players to Table, Update Credit Field, Mark Players Leaving



JSP Enhancement

- Set cookie to remember username
- Send Messages to Other Players
- Application variables

MVC



- Model players, agents
- Controller Servlet (Memory Resident Program)
 - Session, Application Level Variable
 - Start Casino
- Viewer Applets
JSP/JSF Viewer

Casino Network Interface



Synchronized Data
Sockets w Serialized Data
Version 2.0: Considerations

Version – Socket w Serialized Data



- Create Socket Player to Dealer
- Pass Name of Method to Call
- Serialize Parameters
- Note more than One Applet Casino
Can be Active at a Time
- Message List

Tentative Message List



- Video Poker Player, JSP File get Credit from Casino
- enterBJTable
- getBJPlayer
- getTexasHoldemPlayer
- registerCasino
- JSP Files
 - addUser
 - viewUsers
 - getCredit
 - casinoAppletList
 - Recreate Database

Version 2.0 Options



- RMI =
 - Serialized Objects “Naming” JNDI + Socket
- XML Data use XMLHttpRequest
 - Universal Client AS, JS and/or Java Applet
 - Note ActionScript 3.0 Supports Sockets
- JAX-WS Web Methods
 - Web Based RMI or RPC (Remote Procedure Call)

© 2/09 & 2/2010 Professor T. DeDonno

73

XML



- Extensible Mark up Language
- Request/response
 - `<method value=name />`
 - `<chair value=“int “ />`
 - `<table <hand value=“2 3 -1” />`
`<hand value=“51 2” />`
- Work with Other Languages
- Disadvantage w covered Socket/Java IO

© 2/09 & 2/2010 Professor T. DeDonno

74

XML Data Transfer



- XML-RPC
- JSON
- SOAP
- XMLHttpRequest
- <http://en.wikipedia.org/wiki/XML-RPC>
 - Data Types

© 2/09 & 2/2010 Professor T. DeDonno

75

Dropping is in Database Casino Info Request - Response



- Method...
 - `Casino.request(String strReq)`
 - Response is in HTML Format
 - Can write `getStatus`, `getSites`, etc. but you still need

<small>request</small> strReq	HTML Format Response
status	All Info
sites	IP Address and Casino Applet site link
users	Casino Users w Credit Amounts
activeUser s	Casino Users Identifying Active Game s

© 2/09 & 2/2010 Professor T. DeDonno

76

IM



- Game chat
- IP address should be in gamePlayer
- Will be using Direct Play ports
 - Port 6073
- <http://support.microsoft.com/kb/240429>