

Assignment 3

COMP3358 Distributed and Parallel Computing

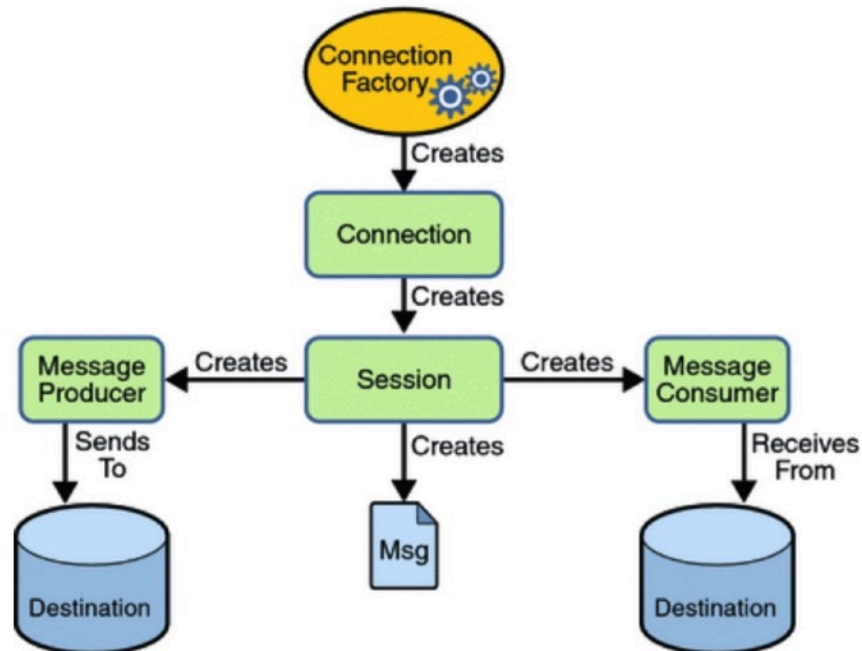
Assignment 3

- ▶ Finishing the 24 games
- ▶ Three major components
 - ▶ The game-play **GUI** (client)
 - ▶ The game-play **mechanism** (server + client)
 - ▶ **Evaluating** (and **validating**) answer
- ▶ You must use **JMS** to support the game-play mechanism

JMS setup

- ▶ JMS provider: GlassFish 5
 - ▶ <https://javaee.github.io/glassfish/download> (Download Full Platform)
- ▶ Connection factory: `jms/J Poker24GameConnectionFactory`
- ▶ Destination: `jms/J Poker24GameQueue`, `jms/J Poker24GameTopic`

JMS Programming Model



JMS Demo

- ▶ You can download a pair of demo program from Moodle
 - ▶ `QueueSenderExample.java`: Sending message to JMS queue
 - ▶ `QueueReceiverExample.java`: Receive message from JMS queue
- ▶ You need to setup **GlassFish** for the program to be executed
 - ▶ See the following slides

Starting GlassFish server

- ▶ Server can be started using command line
- ▶ Command-line utilities: `asadmin`
 - ▶ In a command console, execute “`asadmin start-domain`” to start a domain

Admin console will be accessible after a domain is started

Admin console: `http://localhost:4848`

Connection Factory

Admin console: <http://localhost:4848>

Home About...

User: admin | Domain: domain1 | Server: localhost

GlassFish™ Server Open Source Edition

Tree

Common Tasks

- Domain
 - server (Admin Server)
 - Clusters
 - Standalone Instances
 - Nodes
 - Applications
 - Lifecycle Modules
 - Monitoring Data
 - Resources
 - Concurrent Resources
 - Connectors
 - JDBC
 - JMS Resources**
 - Connection Factories**
 - jms/ __defaultConnectionFa
 - Destination Resources
 - JNDI

JMS Connection Factories

Java Message Service (JMS) connection factories are application resources that are used to create other JMS objects programmatically. Click the name of a connection factory to view its properties.

Connection Factories (1)

New... Delete Enable Disable

Select	JNDI Name	Logical JNDI
<input type="checkbox"/>	jms/ __defaultConnectionFactory	java:comp/DefaultConnectionFactory

New JMS Connection Factory

The creation of a new Java Message Service (JMS) connection factory also creates a connector connection pool for the factory and a connector resource.

General Settings

JNDI Name: *

Resource Type:

Description:

Status:

Name: jms/TestConnectionFactory
Type: javax.jms.ConnectionFactory

OK Cancel

Destination

Common Tasks

- Domain
 - server (Admin Server)
- Clusters
- Standalone Instances
- Nodes
- Applications
- Lifecycle Modules
- Monitoring Data
- Resources
 - Concurrent Resources
 - Connectors
 - JDBC
 - JMS Resources**
 - Connection Factories
 - Destination Resources**

JMS Destination Resources

JMS destinations serve as the repositories for a new destination resource. Click the name of its properties.

Destination Resources (0)

New... Delete Enable Disable

Select	JNDI Name	Status	Enabled	R
No items found.				

New JMS Destination Resource

The creation of a new Java Message Service (JMS) destination resource also creates an admin object resource.

OK Cancel

JNDI Name: *

Physical Destination Name *

Resource Type: *

jms/TestQueue

TestQueue

Destination name in the Message Queue broker does not exist, it will be created automatically when needed.

javax.jms.Queue

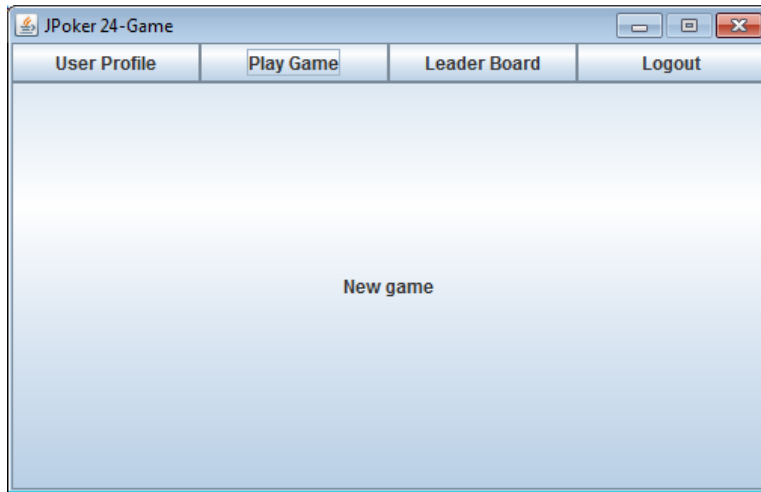
JNDI Name: `jms/TestQueue`
Physical name: `TestQueue`
Type: `javax.jms.Queue`

Setting up in Command Line

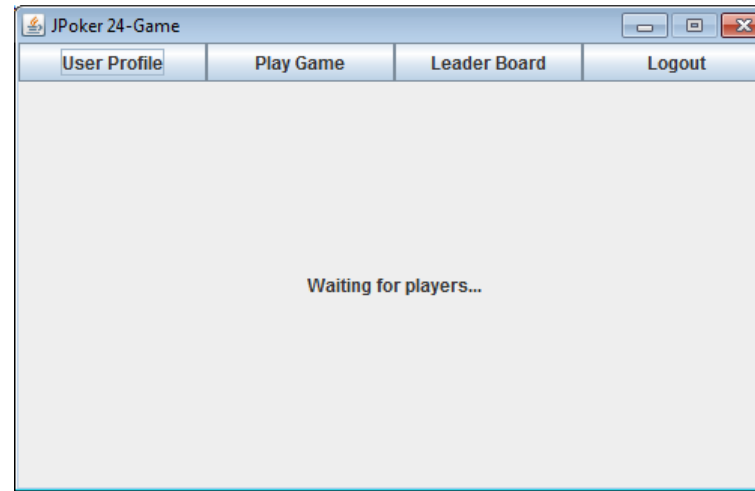
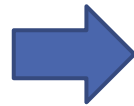
`export CLASSPATH=$CLASSPATH:/path/to/<GlassFish folder>/glassfish/lib/ gf-client.jar`

```
leo@ubuntu:~/A3/Test$ export CLASSPATH=$CLASSPATH:/home/leo/glassfish5/glassfish
/lib/gf-client.jar
leo@ubuntu:~/A3/Test$ java QueueSenderExample
Mar 22, 2024 5:38:11 PM com.sun.enterprise.v3.server.CommonClassLoaderServiceImp
l findDerbyClient
INFO: Cannot find javadb client jar file, derby jdbc driver will not be availabl
e by default.
Mar 22, 2024 5:38:13 PM org.hibernate.validator.internal.util.Version <clinit>
INFO: HV000001: Hibernate Validator 6.0.10.Final
Mar 22, 2024 5:38:14 PM com.sun.messaging.jms.ra.ResourceAdapter start
INFO: MQJMSRA_RA1101: GlassFish MQ JMS Resource Adapter: Version: 5.1.3 (Build
1-a) Compile: January 27 2019 1625
Mar 22, 2024 5:38:14 PM com.sun.messaging.jms.ra.ResourceAdapter start
INFO: MQJMSRA_RA1101: GlassFish MQ JMS Resource Adapter starting: broker is REMO
TE, connection mode is TCP
Mar 22, 2024 5:38:14 PM com.sun.messaging.jms.ra.ResourceAdapter start
INFO: MQJMSRA_RA1101: GlassFish MQ JMS Resource Adapter Started:REMOTE
Sending message 1
```

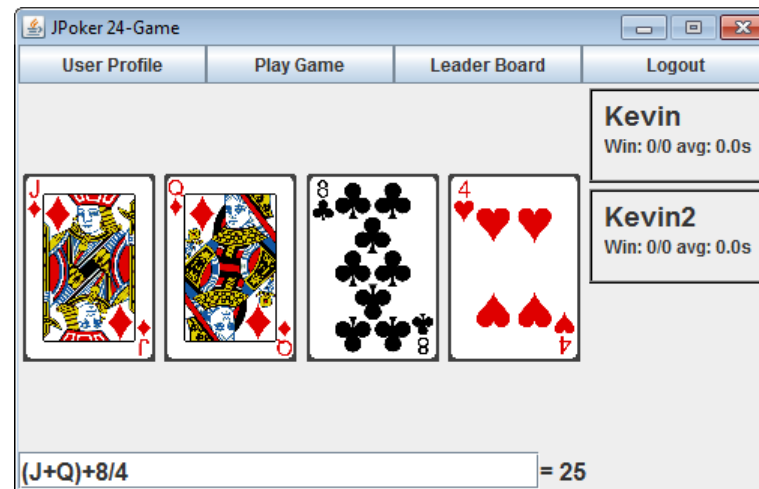

Game Stages



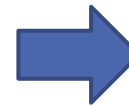
Initial



Game-joining

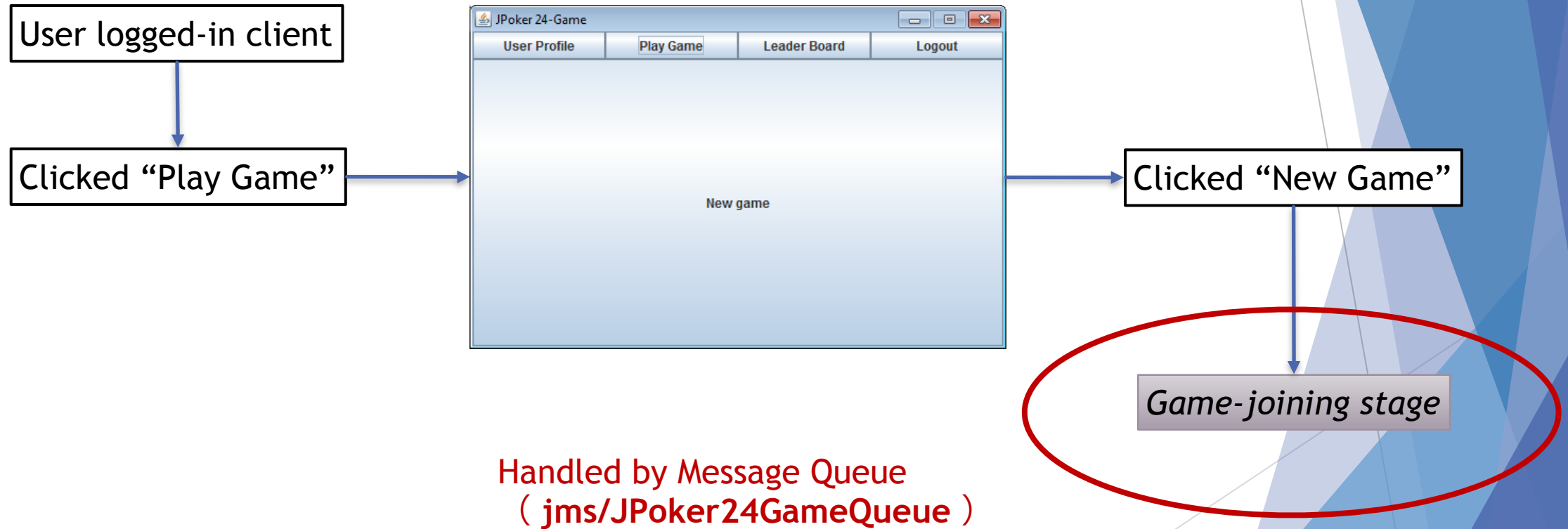


Game-playing

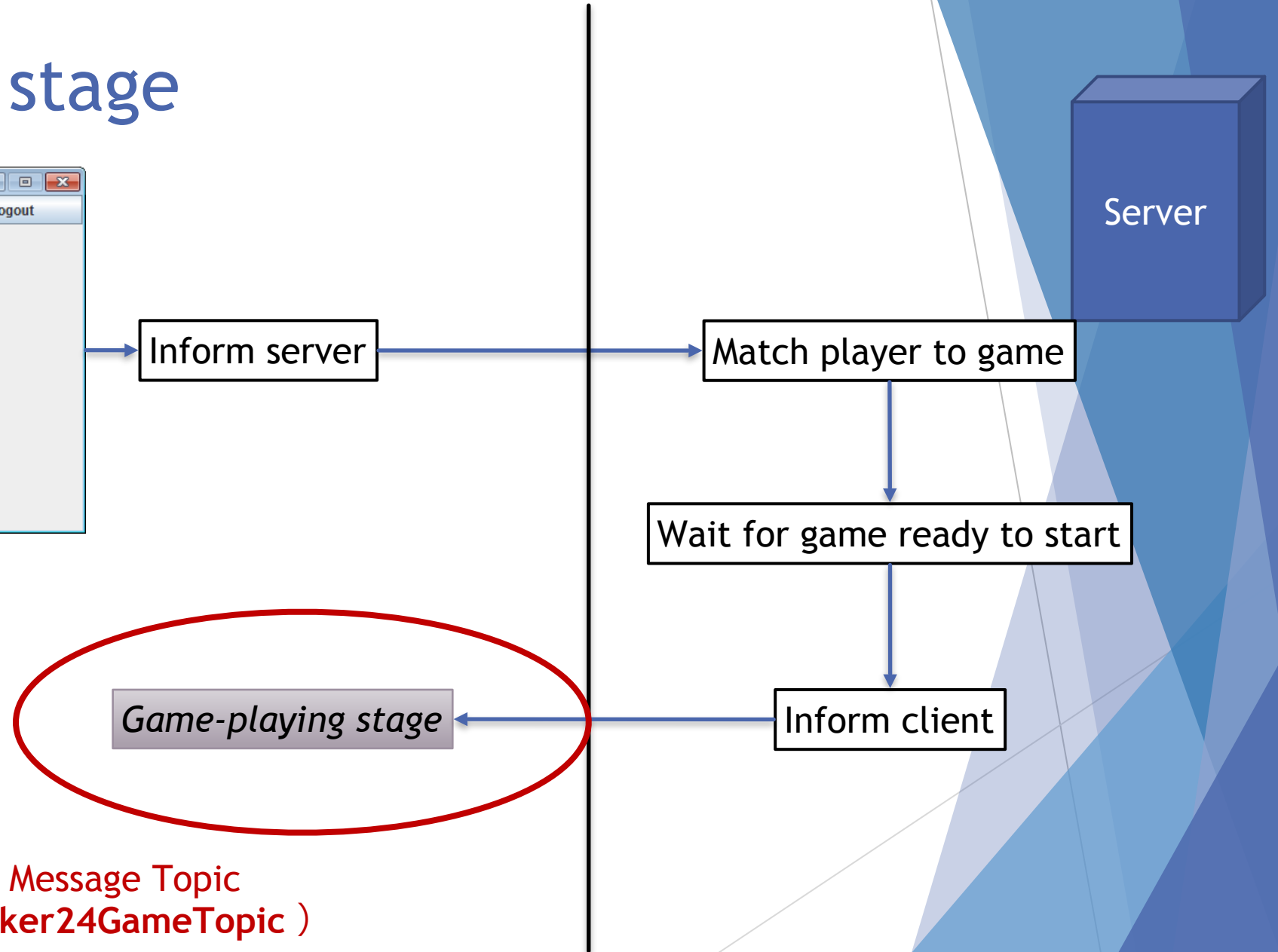
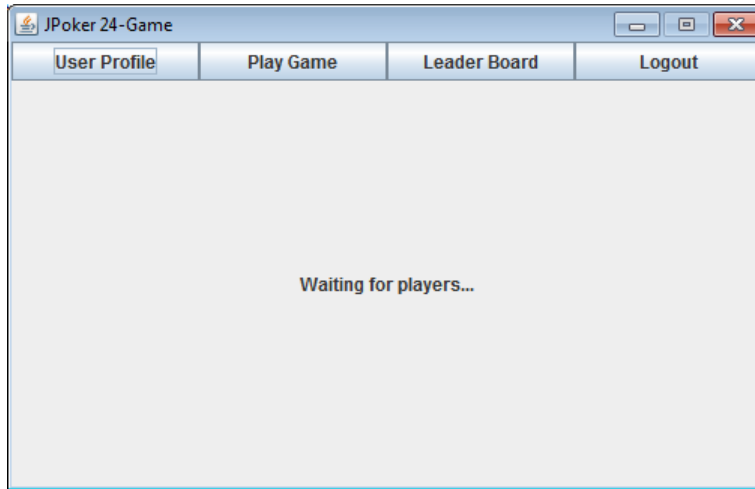


Game-over

Initial stage

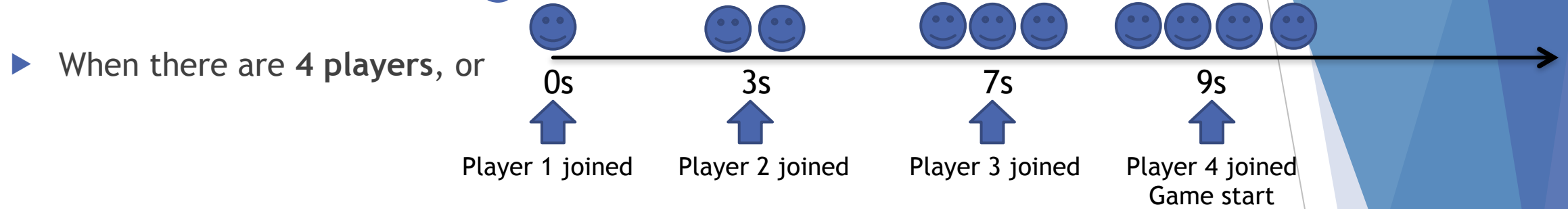


Game-joining stage

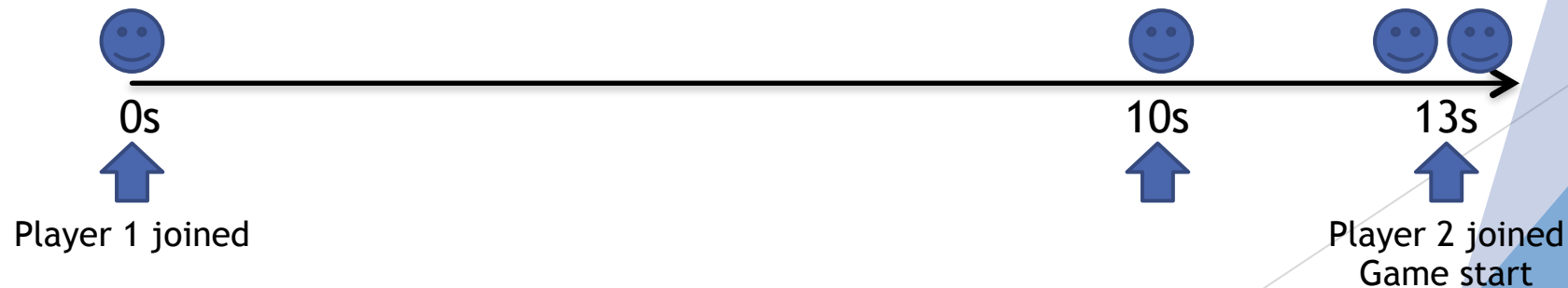
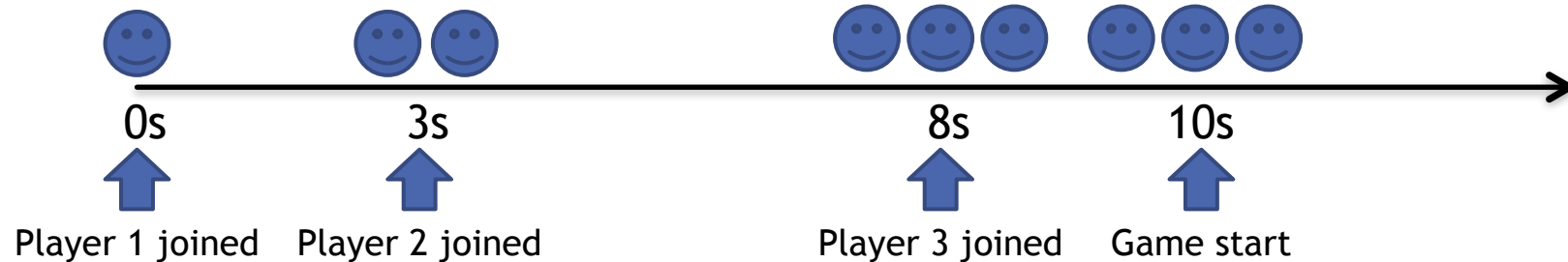


Handled by Message Topic
(`jms/JPoker24GameTopic`)

When to start a game?

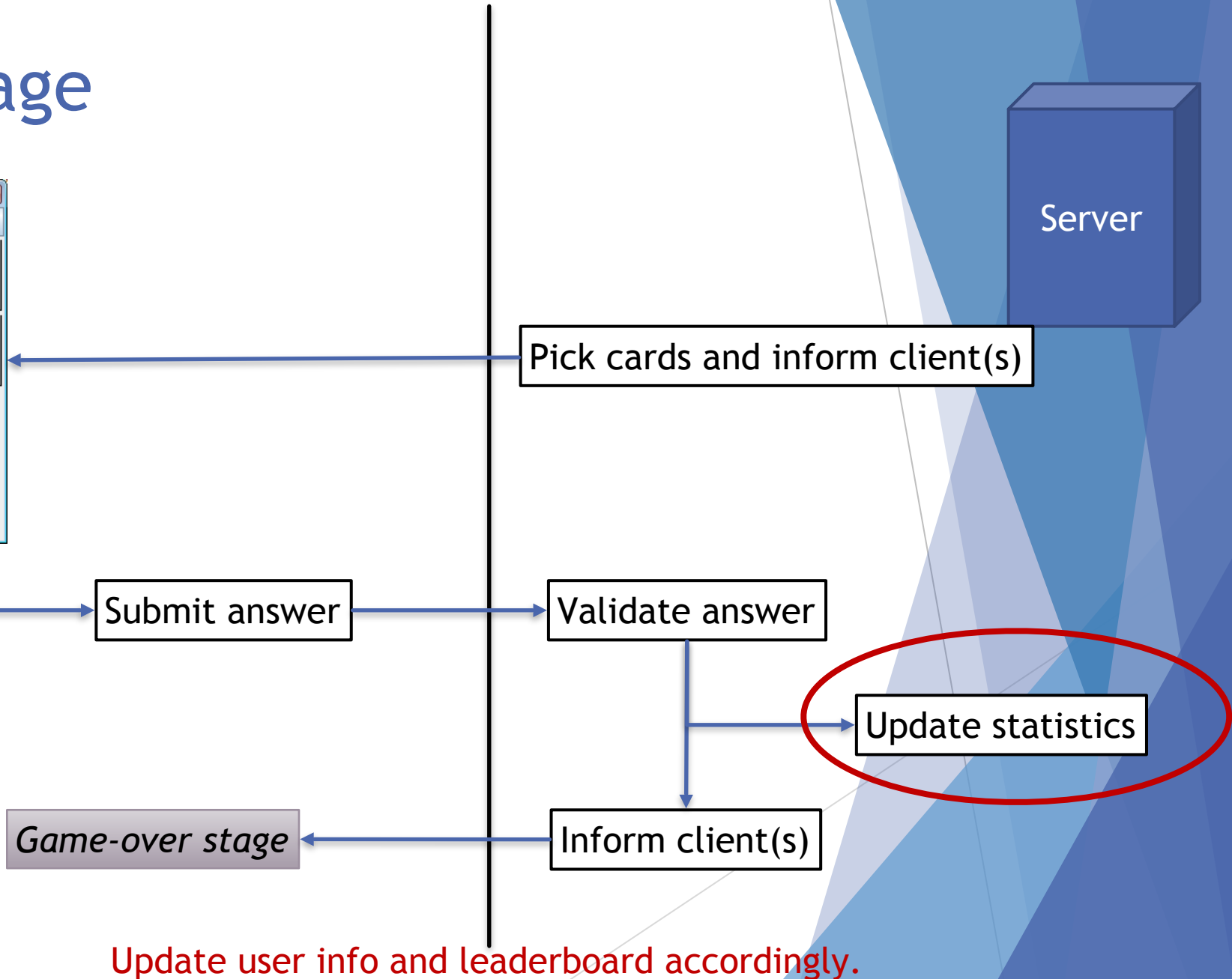
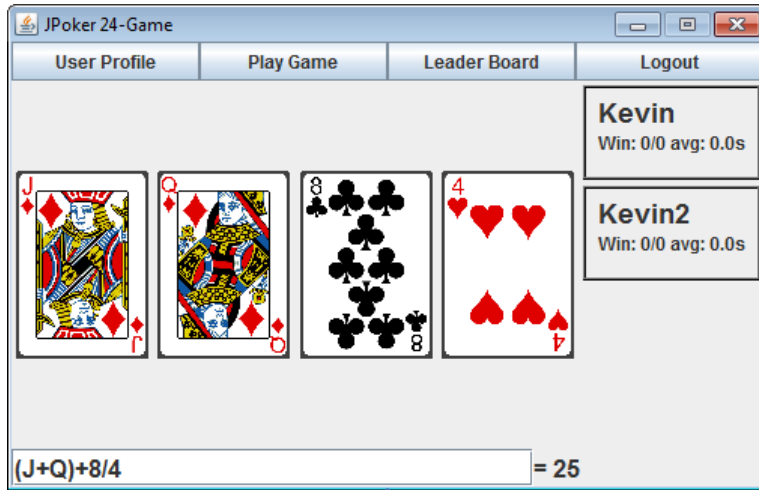


- When there are 2 players and 10 seconds has passed since the first player joined



We can assume the server always starts only one game simultaneously.

Game-playing stage



Leaderboard

The leaderboard should contain the following info:

- Rank (ranked according to games win)
- Player Name
- Games win
- Games Played
- Average winning time



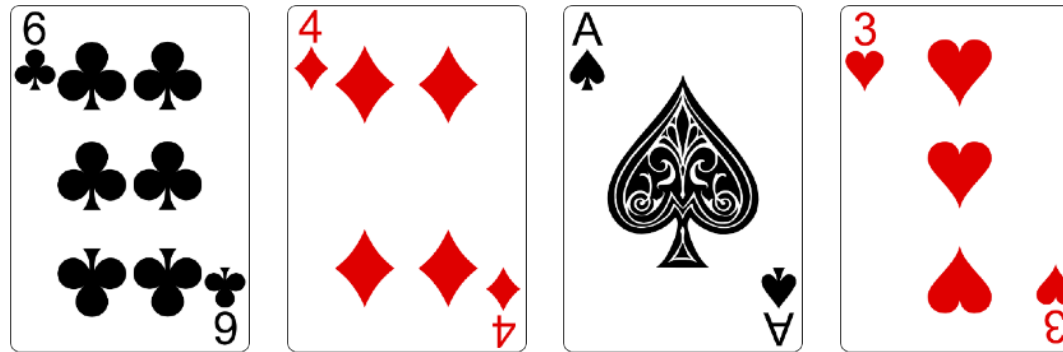
The screenshot shows a web application window titled "JPoker 24-Game". It has four tabs: "User Profile", "Play Game", "Leader Board" (which is selected), and "Logout". Below the tabs is a table displaying the leaderboard data.

Rank	Player	Games won	Games played	Avg. winning ti...
1	Player 4	20	35	10.4s
2	Player 2	18	25	13.2s
3	Player 6	18	31	15.1s
4	Player 8	16	30	12.8s
5	Player 7	10	25	10.2s
6	Player 3	5	7	17.1s
7	Player 5	4	10	15.4s
8	Player 10	1	2	16.2s
9	Player 9	1	4	14.1s
10	Player 1	1	4	18.4s

Do we need transactions in updating leaderboard?

Game rule

- ▶ 4 Cards of **different values** should be drawn.
- ▶ Suit doesn't matters
- ▶ Goal: an expression that equals **24**
- ▶ Available operations: $+$, $-$, \times , \div
- ▶ **Parenthesis** can be used to override precedence
- ▶ **Fraction** may appear in intermediate result



What is the answer?

[https://en.wikipedia.org/wiki/24_\(puzzle\)](https://en.wikipedia.org/wiki/24_(puzzle))

Game-over stage



Clicked "Next Game"

Game-joining stage

Submission requirements

- ▶ Submit all your final *.java file(s) and a document (better in pdf format).
 - ▶ **JPoker24Game.java**: the main client program
 - ▶ **JPoker24GameServer.java**: the main server program
 - ▶ **Other source code you have implemented.**
 - ▶ **lib.zip** (optional): external libraries used
 - ▶ **A document**: The document should include how your program is organized, how to compile your code, how to run your code, and your screenshots of running each of the operations (i.e., the stages mentioned in the slides).

Checkpoints for Assignment3

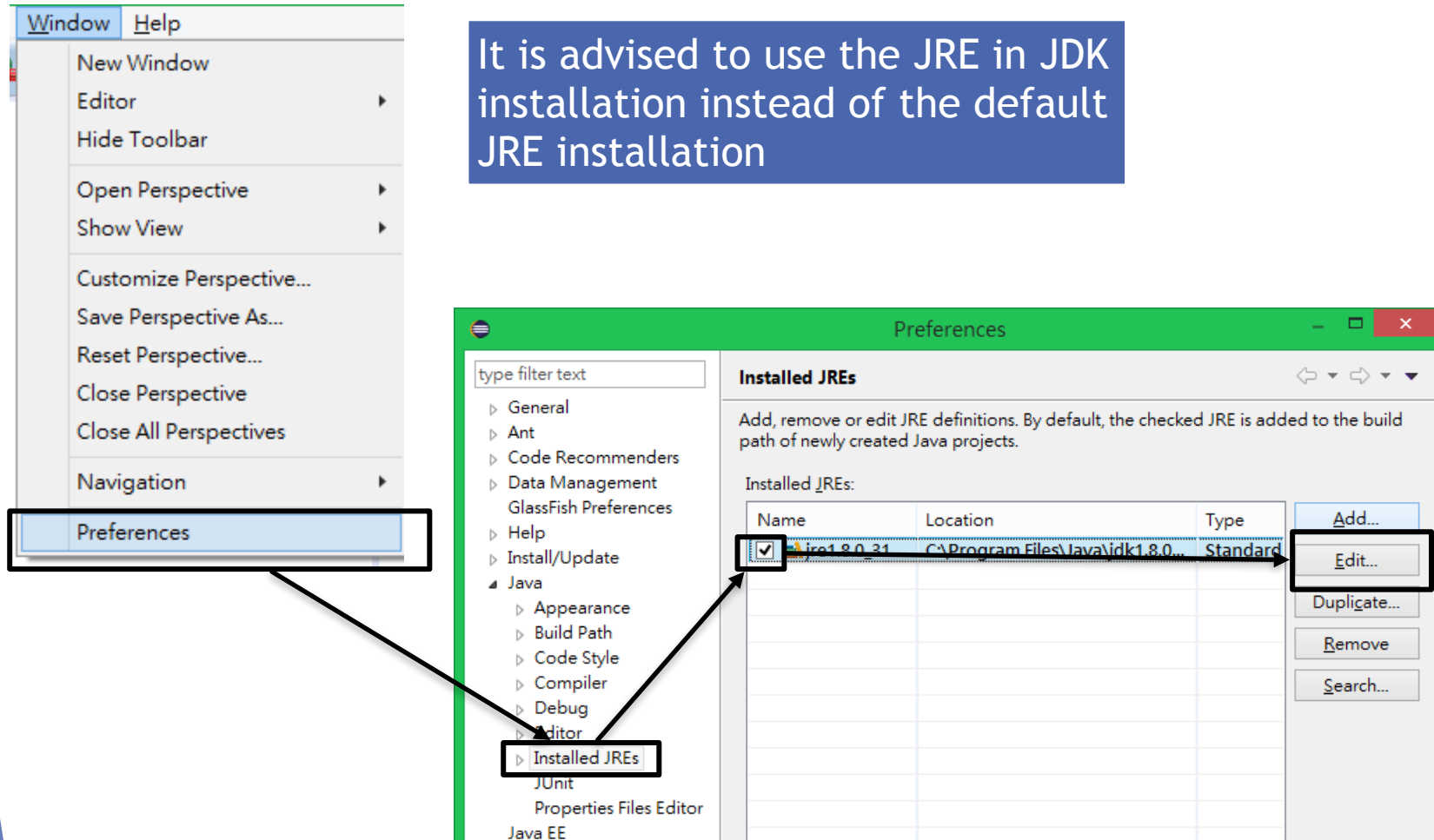
- ▶ Handle Gaming-join stage use JMS correctly (30%)
- ▶ Handle Gaming-playing stage use JMS correctly (40%)
- ▶ Handle Gaming-over stage correctly (5%)
- ▶ Update Leaderboard correctly (25%)

Please provide at least one screenshot for each checkpoint.

Backup Slides (Eclipse)

Setting up Eclipse execution environment

It is advised to use the JRE in JDK installation instead of the default JRE installation



Backup Slides (Eclipse)

Set up project in Eclipse

- ▶ Create a new project and import the two Java files
- ▶ Add **gf-client.jar** to external library in Eclipse (right-click on project → properties)
 - ▶ It can be found in *<GlassFish folder>/glassfish/lib*

You can now execute the programs in Eclipse