Distributed Gaming using J2ME

By Rekha Vaddepalli
Agenda

- Introduction
- Requirements
- Technologies Used
- Design and Implementation
- Experiments and Results
- Conclusion
Introduction

- Mobile devices usage is increasing rapidly
- In tandem, mobile games development is increasing
- Development platforms
  - J2ME, C++ etc
Requirements

Purpose

To develop a distributed, multi-player game with a central server to simulate certain economic scenarios.

Sample scenarios tested

- Making a certain product tougher to produce and also making it costlier than the other products
- Choosing to produce a certain product benefits the player
- Simulation of real world issues

It can also be viewed as a regular strategy game by the people more interested in strategy games
Requirements (contd..)

Scope

To implement *Acquire* game for J2ME-enabled PDAs using J2ME, Servlets, JDBC and MySQL

*Acquire* game borrows ideas from old computer game "Mule"

Game world consists of number of plot areas comprising of nine sub plots

- Each plot has 3 properties
  - Mine Value
  - Farm Value
  - Energy Value
Requirements (contd..)

Game consists of 4 stages
- Selection of plot areas
- Configuration of plot areas
- Production
- Auction

Rules
- If the player satisfies critical resources limits of all type of products, can go for another round of selection
- The player must finish the auction that he started in the previous round to start the auction in the next round
J2ME (Java 2 Micro Edition)

Two key components

- Configuration
  - JVM for each kind of device
  - Defines the Java Runtime Environment and core classes that operate on each device
  - Ex: CLDC & CDC

- Profile
  - Consists of Java classes that enable implementation of features for a particular device or group of devices
  - Ex: MIDP, Foundation Profile etc.
J2ME Concepts (contd..)

- User Interface Classes
  - Used Form class for all the screens
  - Used CustomItem class for game world representation

- Persistent Storage
  - Used for storing intermediate values in Configuration stage

- Generic Connection Framework
  - Used for communicating with the server
MIDP Applications (MIDlets)

Introduction of Verification step after Compilation
- Divided into two steps
  - Pre-verification is done off the device
  - Simple second verification step on the device

Deployment
- Using MIDlet suites
  - JAR file
  - Manifest File (included in JAR)
  - Application Descriptor (outside JAR)
Other technologies used

- Java Servlets
- Java Database Connectivity (JDBC)
- MySQL Database
## Operating Environment

<table>
<thead>
<tr>
<th>Application</th>
<th>Operating Environment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Game Client</td>
<td>J2ME Wireless Toolkit, Windows CE/ME</td>
</tr>
<tr>
<td>Game Server</td>
<td>Java 1.4.1 or higher installed on Windows</td>
</tr>
<tr>
<td>Database</td>
<td>Oracle 9i / MySQL</td>
</tr>
</tbody>
</table>
Design and Implementation
System Architecture

- J2ME devices
- Server
- Database
- HTTP
- JDBC
Server Design

- Register & Login
- Create New Game World
- Update
  - On Selection
  - On Configuration
  - On Role Declaration
  - On Trade
  - On Transfer Of Units
  - On Log off
- Get Score
Rule 1: Produce >= critical limit (for each type of the product)
Rule 2: Finished auction started in the previous round
Logging in
Selection Phase
Configuration/Production
Auction Stage
Local Auctions

Game 1, 2 & 3 can be seen as 3 different local auctions

A is involved in both Game 1 & 2

So, A can transfer units between the two local auctions
Experiments
Realization of local auctions

Game One

Game Two

Before & After Transfer Of Units

Before & After Transfer of units

Production Values & Score

Production Values and Score

Before & After Transfer Of Units

Before & After Transfer of units
Configuration file on the Server

- Helps to create different scenarios
- This file contains
  - Several initial parameters
    - Initial mine units, food units etc.
    - Time slots for Selection, Configuration etc.
  - Expenditure Formulae
  - Production Formulae
  - Score formula
Test Case One

Description

- Making one product tougher to produce compared to other products and that particular product is sold at a higher price.
- For this test case, I chose mines to be the tougher to produce product.
- Machinery Expenditure formula is modified so that less mines are produced.

Production = (Type_Of_Product * Appropriate_Property_Value)

MachineryExpenditure = 10 * (Sum of Property Values)
Test Case One: With 2 players

Results

Test Case One: 2 Players

<table>
<thead>
<tr>
<th>Score</th>
<th>Average Mine Value</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **Round 1**: A, B
- **Round 2**: A, B
- **Round 3**: A

Legend:
- Blue diamond: Round 1
- Pink square: Round 2
- Yellow triangle: Round 3
Test Case One: With 4 players

Results

Test Case One: 4 Players

<table>
<thead>
<tr>
<th>Mine Units</th>
<th>Rokda Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>100</td>
<td>200</td>
</tr>
<tr>
<td>200</td>
<td>400</td>
</tr>
<tr>
<td>300</td>
<td>600</td>
</tr>
<tr>
<td>400</td>
<td>800</td>
</tr>
<tr>
<td>500</td>
<td>1000</td>
</tr>
<tr>
<td>600</td>
<td>1200</td>
</tr>
</tbody>
</table>

Size of bubble: Score

Rokda Units

Mine Units

Round1

Round3
Test Case Two

Description

- Just deciding to produce a particular product will increase the score
- For this test case, I chose that if the player chooses to mine, he would have a better score

Score = (Mine_Count * 10) + (Farm_Count * 2) + (Energy_Count * 2)
Test Case Two: With 2 players

Results

**Mining Count Chart**

<table>
<thead>
<tr>
<th>Rounds</th>
<th>A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>7</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>9</td>
<td>6</td>
</tr>
</tbody>
</table>

**Scores**

<table>
<thead>
<tr>
<th>Rounds</th>
<th>A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>60</td>
<td>40</td>
</tr>
<tr>
<td>2</td>
<td>100</td>
<td>80</td>
</tr>
</tbody>
</table>
Test Case Two: With 4 Players
Results

Test Case Two: 4 players

Score

Mining Count

Round 1
Round 2
Round 3
Round 4
Test Case Three

Description

The worker and land expenses are higher for producing mines than for farming or energy production.

WorkerExpensesForMining = (10* (Sum Of Property Values))
LandExpensesForMining = (10* (Sum Of Property Values))
WorkerExpensesForFarming = (2* (Sum Of Property Values))
LandExpensesForFarming = (2* (Sum Of Property Values))
WorkerExpensesForEnergy = (3* (Sum Of Property Values))
LandExpensesForEnergy = (3* (Sum Of Property Values))
Test Case Three: With 2 Players
Results

Comparison Chart

Score Comparisons in the two rounds
Test Case Three: 4 Players Results

Comparison Chart

Comparison of Score values

Round 1 & 2

Production/Units

Score

Rounds

M: Male
F: Female
R: Male
E: Female

Comparison Chart

Production/Units

Score

Rounds

M: Male
F: Female
R: Male
E: Female

Comparison of Score values
Conclusion

Possible applications of this project
- Economic simulations
- Strategy game

Similar Applications
- The Economics classes are generally passive and applications like these will help in the better understanding of the Economics concepts
  - [http://www.people.virginia.edu/~cah2k/programs.html](http://www.people.virginia.edu/~cah2k/programs.html)
- They claim that they had good success using PDAs in the classrooms for their experiments
Future Enhancements

- Better security features.
- More types of products can be introduced to make it more realistic.
- Implementation of more types of auctions that are possible in the real world.
Questions?