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)

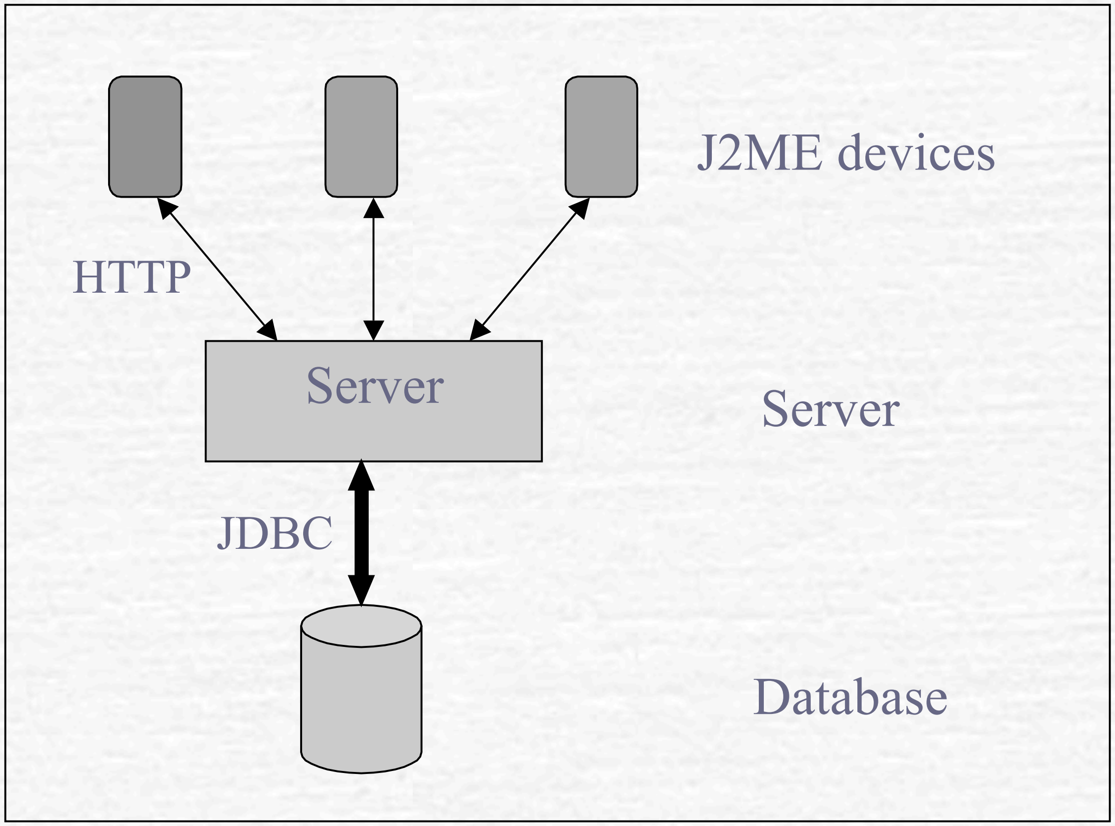
Other technologies used

- Java Servlets
- Java Database Connectivity (JDBC)
- MySQL Database

Operating Environment

Application	**Operating Environment**
Game Client	J2ME Wireless Toolkit, Windows CE/ME
Game Server	Java 1.4.1 or higher installed on Windows
Database	Oracle 9i / MySQL

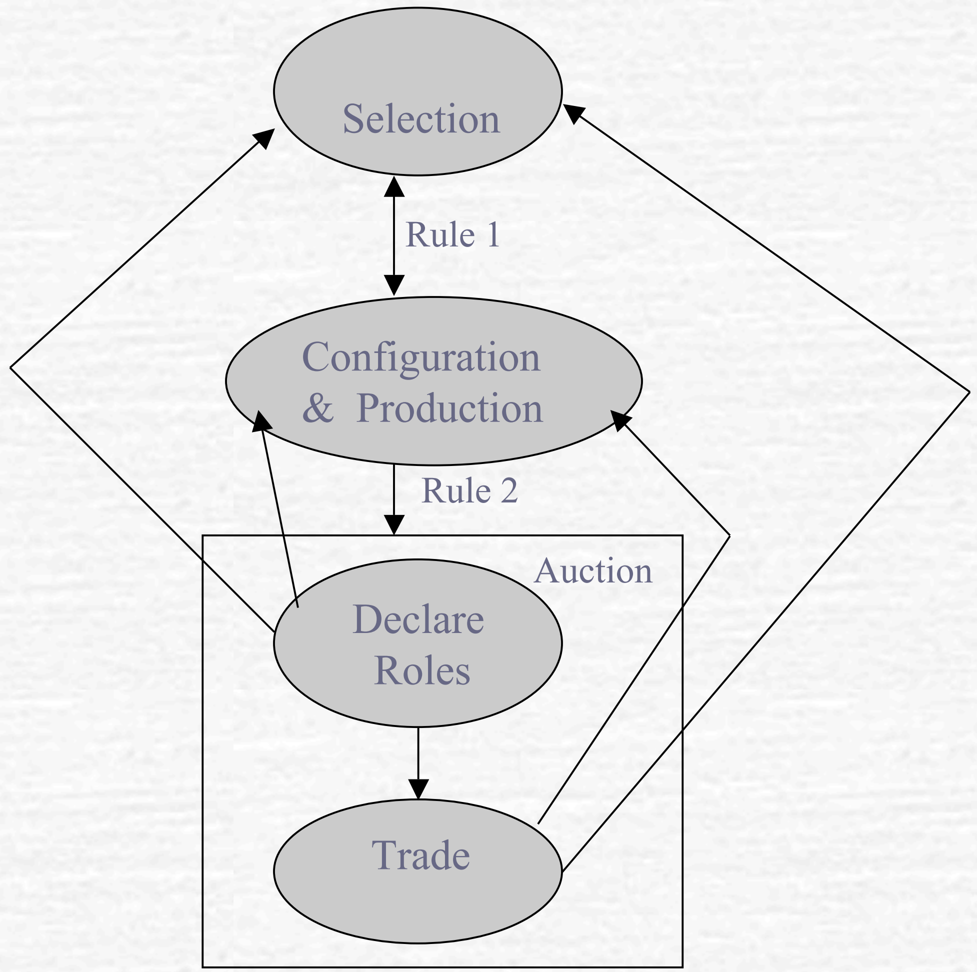
Design and Implementation
System Architecture



Server Design

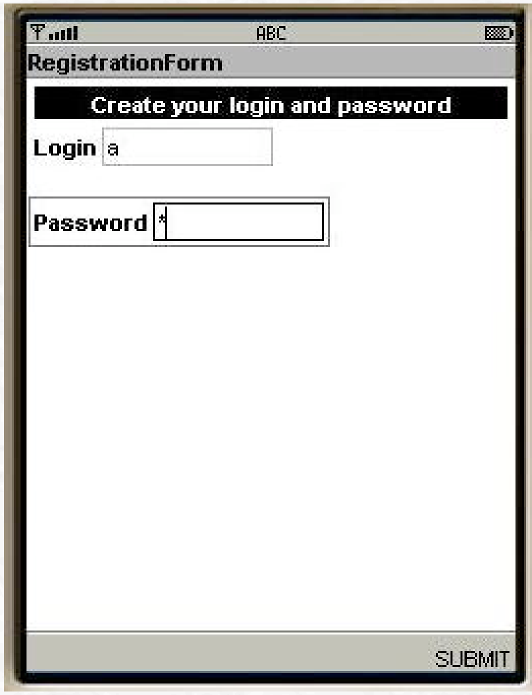
- Register & Login
- Create New Game World
- Update
  - On Selection
  - On Configuration
  - On Role Declaration
  - On Trade
  - On Transfer Of Units
  - On Log off
- Get Score

Game Design



Rule 1: Produce >= critical limit (for each type of the product)
Rule 2: Finished auction started in the previous round

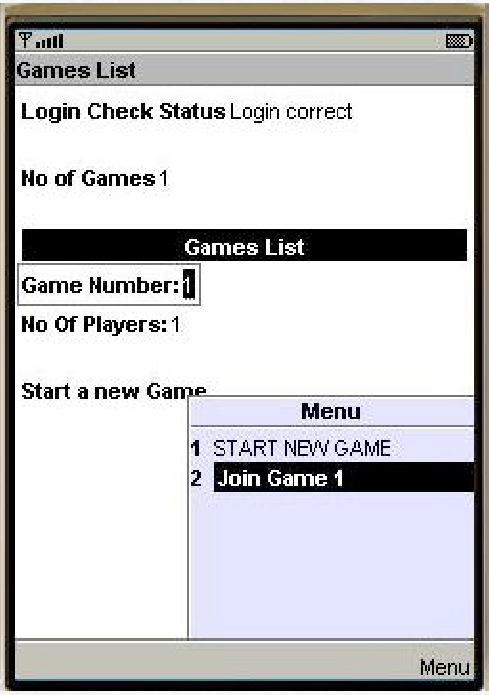
Logging in


RegistrationForm

Create your login and password

Login a

Password *


Games List

Login Check Status Login correct

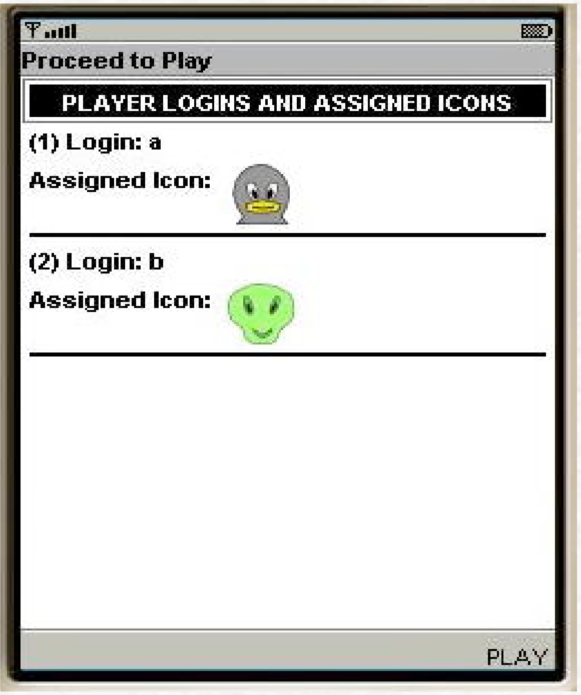
No of Games 1

Games List

Game Number: 1

No Of Players: 1

Start a new Game


Proceed to Play

PLAYER LOGINS AND ASSIGNED ICONS

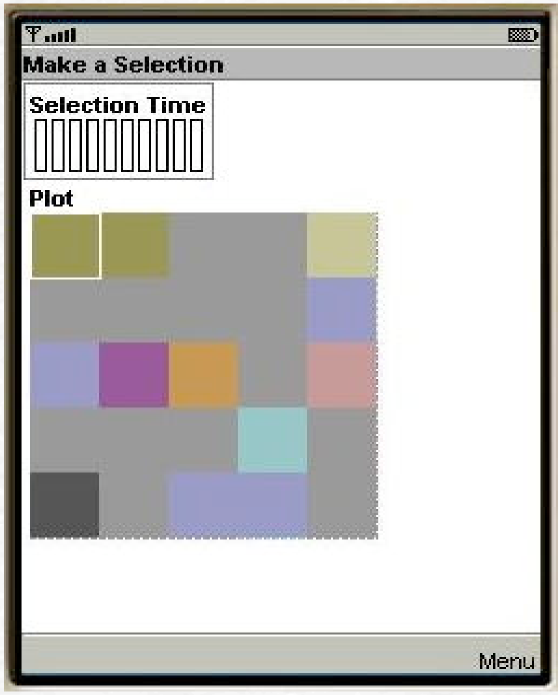
(1) Login: a

Assigned Icon:

(2) Login: b

Assigned Icon:

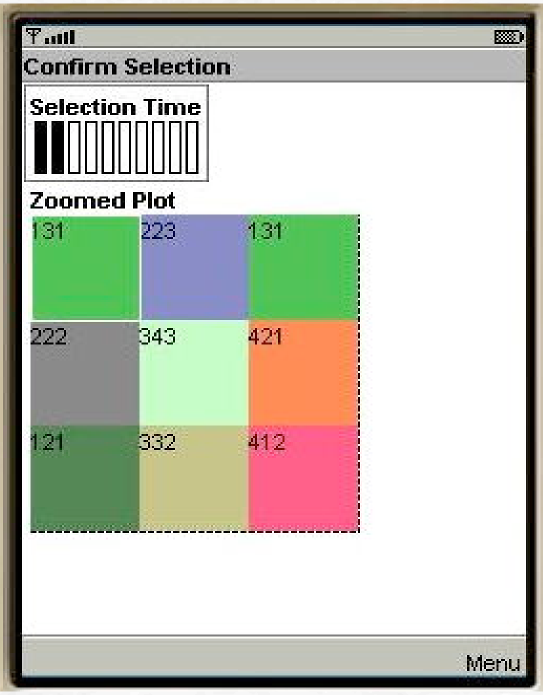
Selection Phase



Make a Selection

Selection Time

Plot

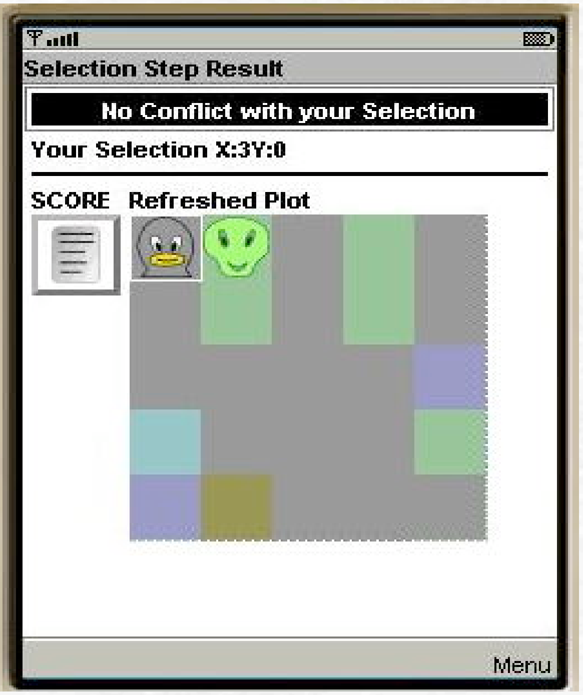


Confirm Selection

Selection Time

Zoomed Plot

131	223	131
222	343	421
121	332	412



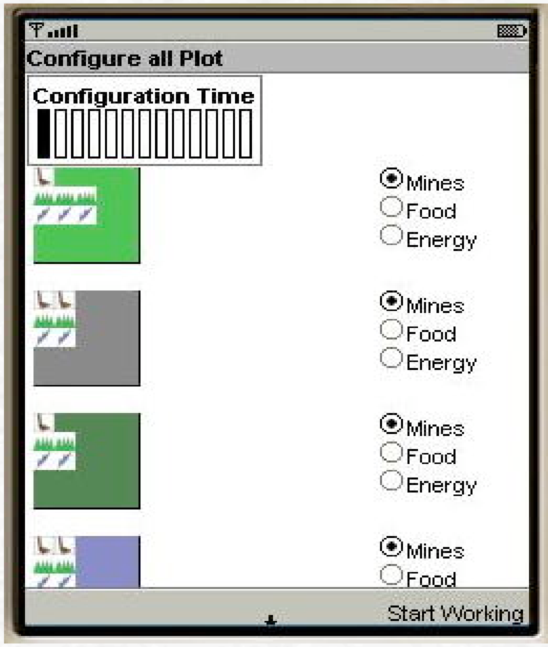
Selection Step Result

No Conflict with your Selection

Your Selection X:3Y:0

SCORE Refreshed Plot

Configuration/Production


Configure all Plot

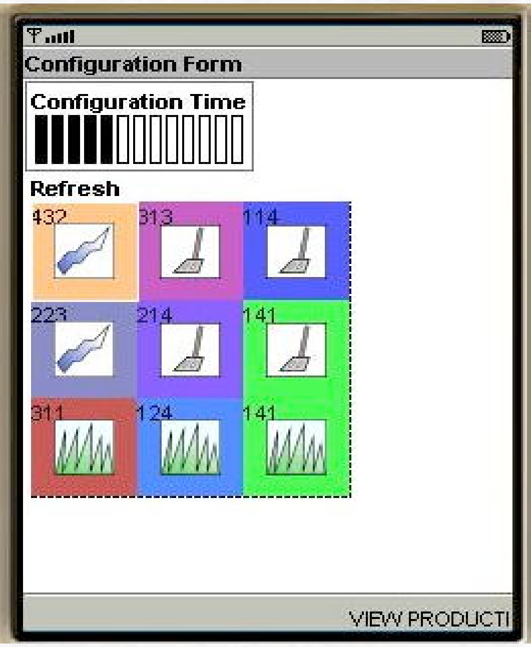
Configuration Time

● Mines
○ Food
○ Energy

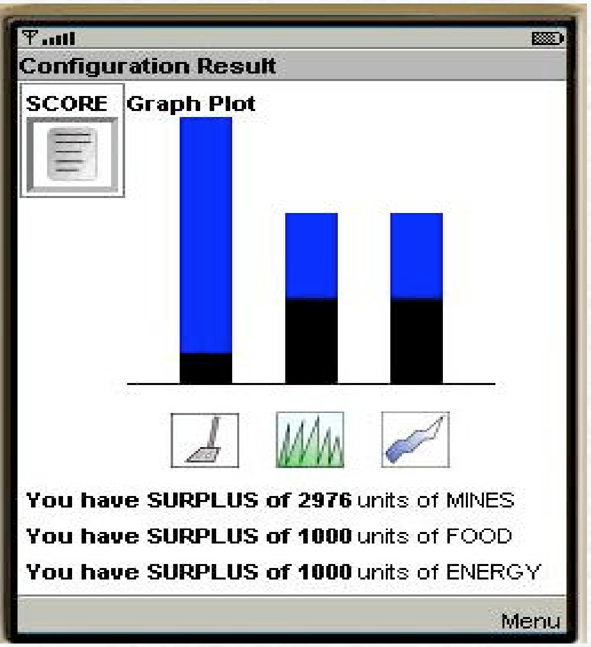
● Mines
○ Food
○ Energy

● Mines
○ Food
○ Energy

● Mines
○ Food


Configuration Form

Configuration Time

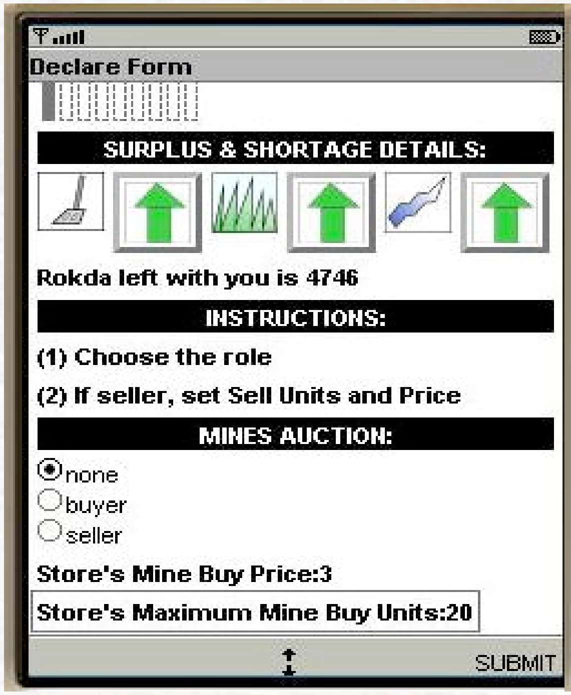

Configuration Result

You have SURPLUS of 2976 units of MINES

You have SURPLUS of 1000 units of FOOD

You have SURPLUS of 1000 units of ENERGY

Auction Stage



Declare Form

SURPLUS & SHORTAGE DETAILS:

Rokda left with you is 4746

INSTRUCTIONS:

(1) Choose the role

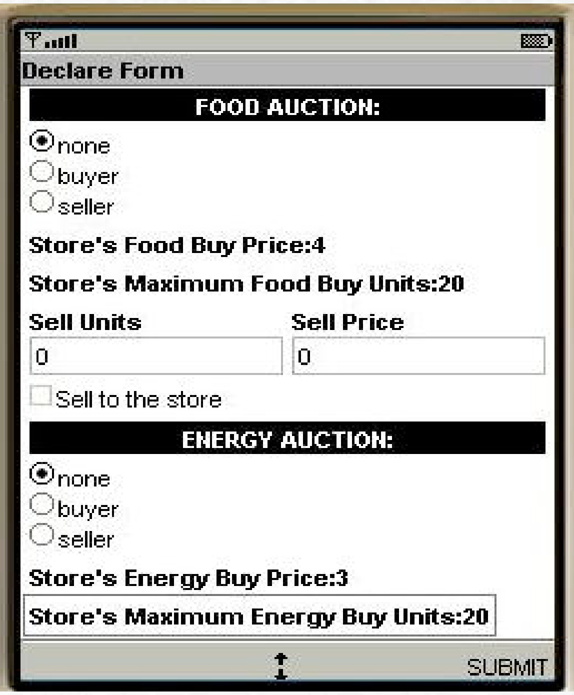
(2) If seller, set Sell Units and Price

MINES AUCTION:

● none
○ buyer
○ seller

Store's Mine Buy Price:3

Store's Maximum Mine Buy Units:20



Declare Form

FOOD AUCTION:

● none
○ buyer
○ seller

Store's Food Buy Price:4

Store's Maximum Food Buy Units:20

Sell Units	Sell Price
0	0

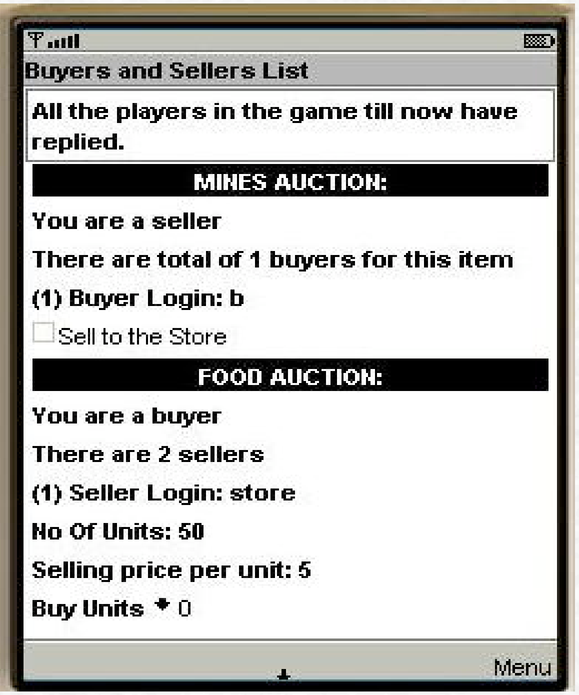
□ Sell to the store

ENERGY AUCTION:

● none
○ buyer
○ seller

Store's Energy Buy Price:3

Store's Maximum Energy Buy Units:20



Buyers and Sellers List

All the players in the game till now have replied.

MINES AUCTION:

You are a seller

There are total of 1 buyers for this item

(1) Buyer Login: b

□ Sell to the Store

FOOD AUCTION:

You are a buyer

There are 2 sellers

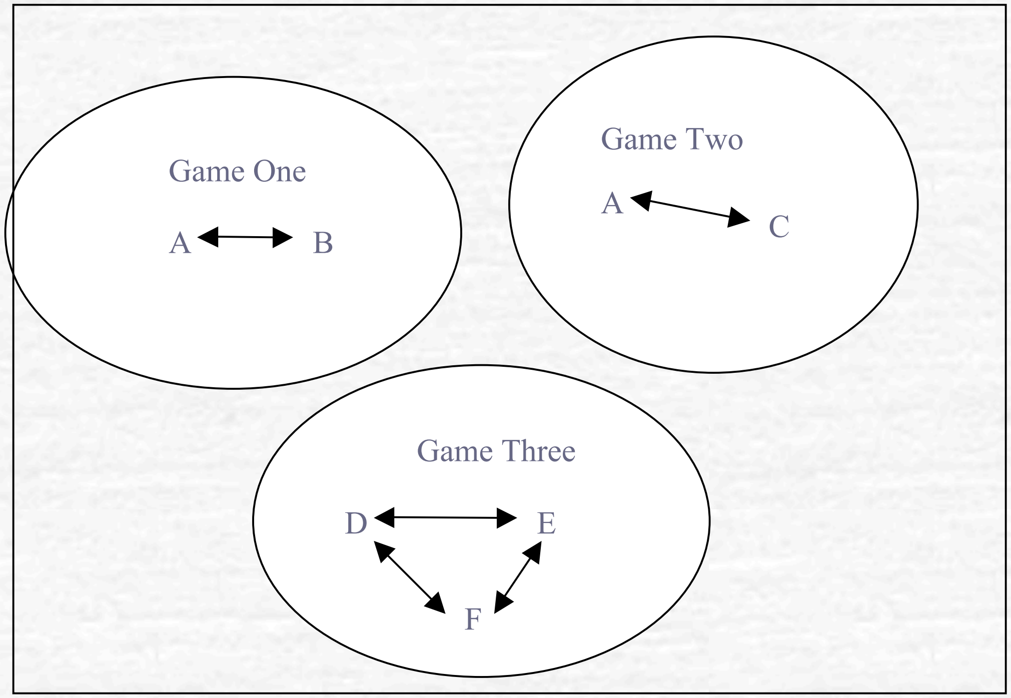
(1) Seller Login: store

No Of Units: 50

Selling price per unit: 5

Buy Units 0

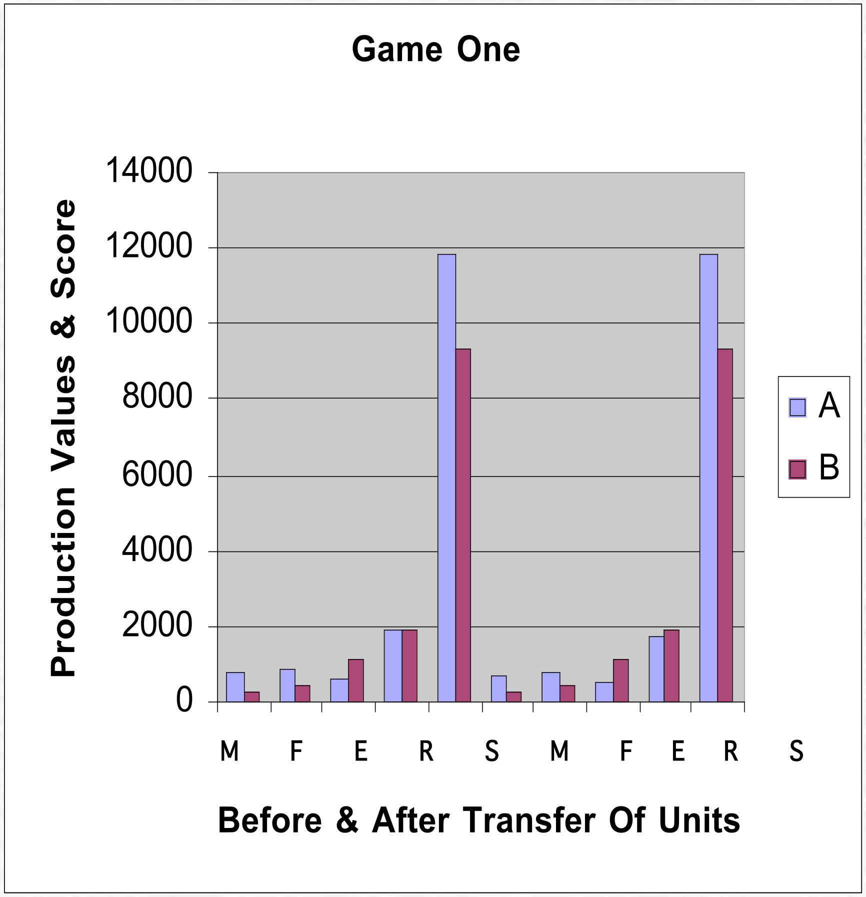
Local Auctions

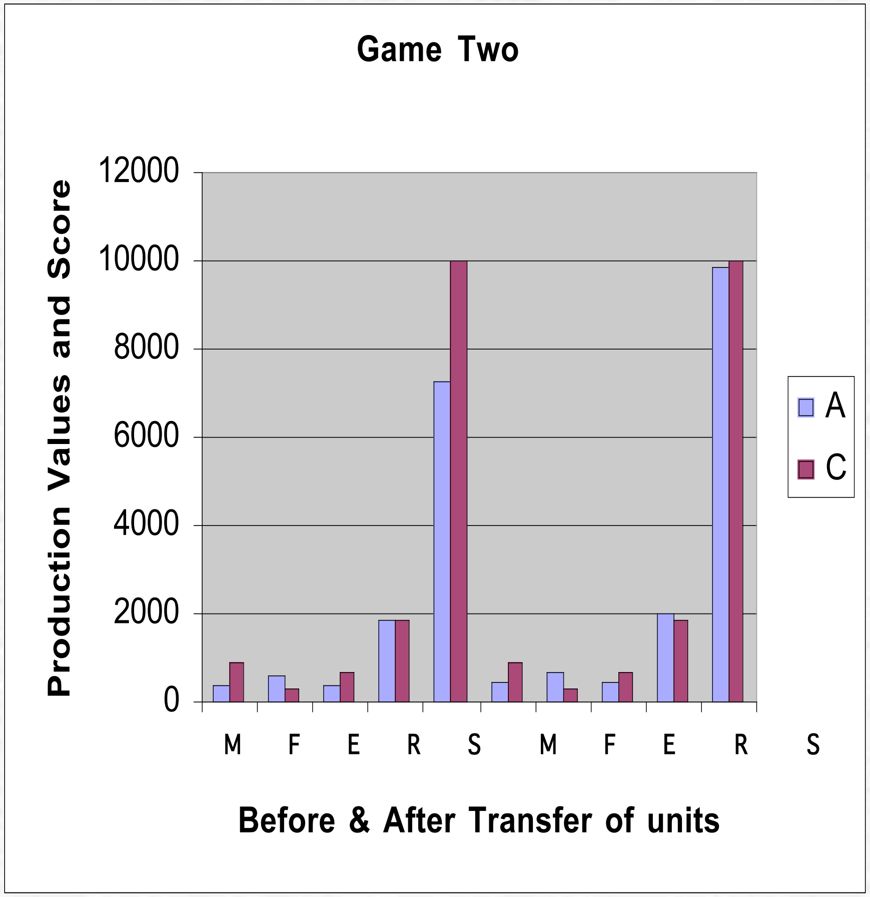


- Game 1, 2 & 3 can be seen as 3 different local auctions
- A is involved in both Game 1 & 2
- So, A can transfer units between the two local auctions

Experiments

Realization of local auctions





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

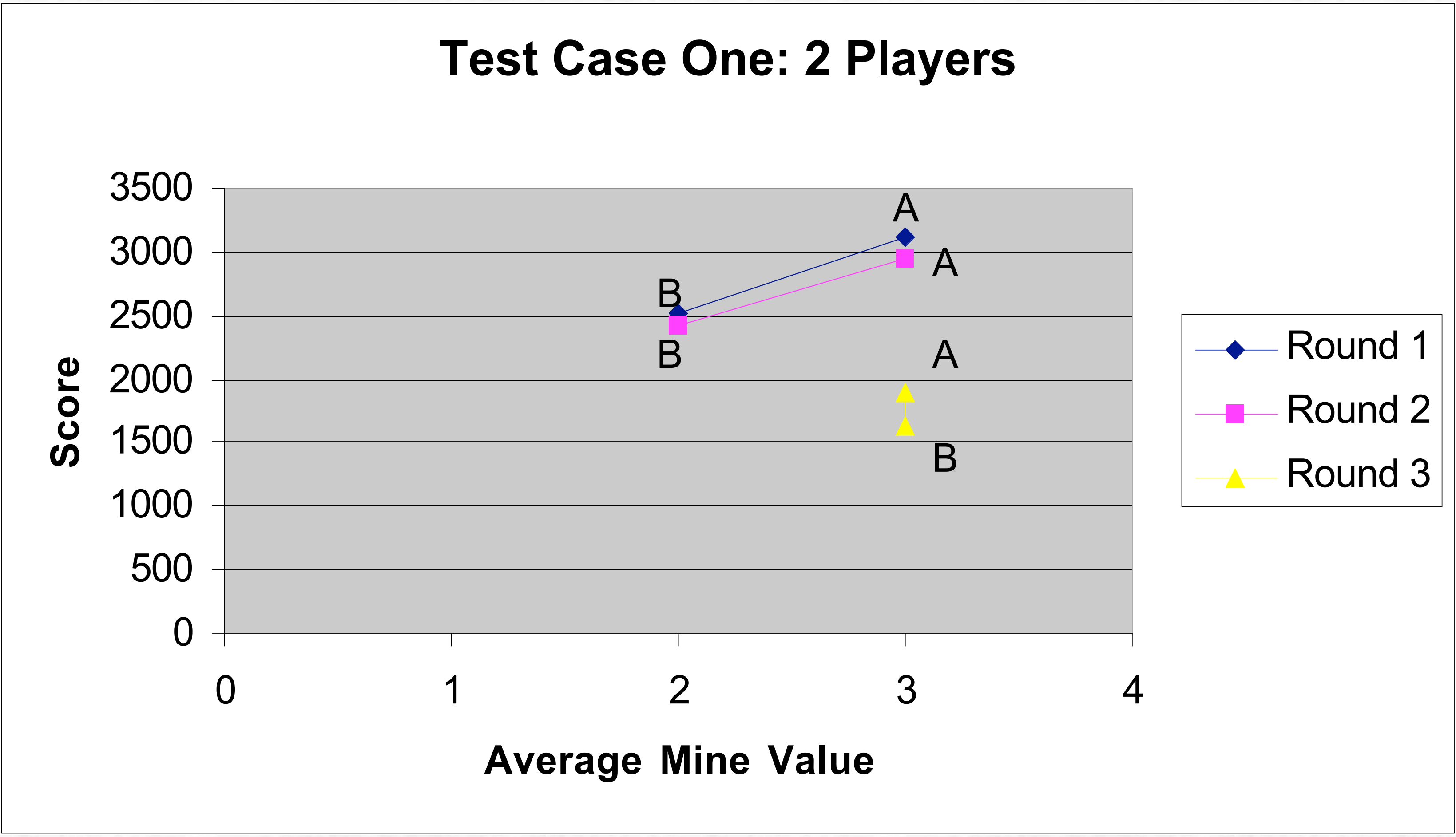
- Description
  - Making one product tougher to produce compared to other products and that particular product is sold at a higher price
  - For this test case, I chose mines to be the tougher to produce product.
  - Machinery Expenditure formula is modified so that less mines are produced.

Production = (Type_Of_Product * Appropriate_Property_Value)

MachineryExpenditure = 10 * (Sum of Property Values)

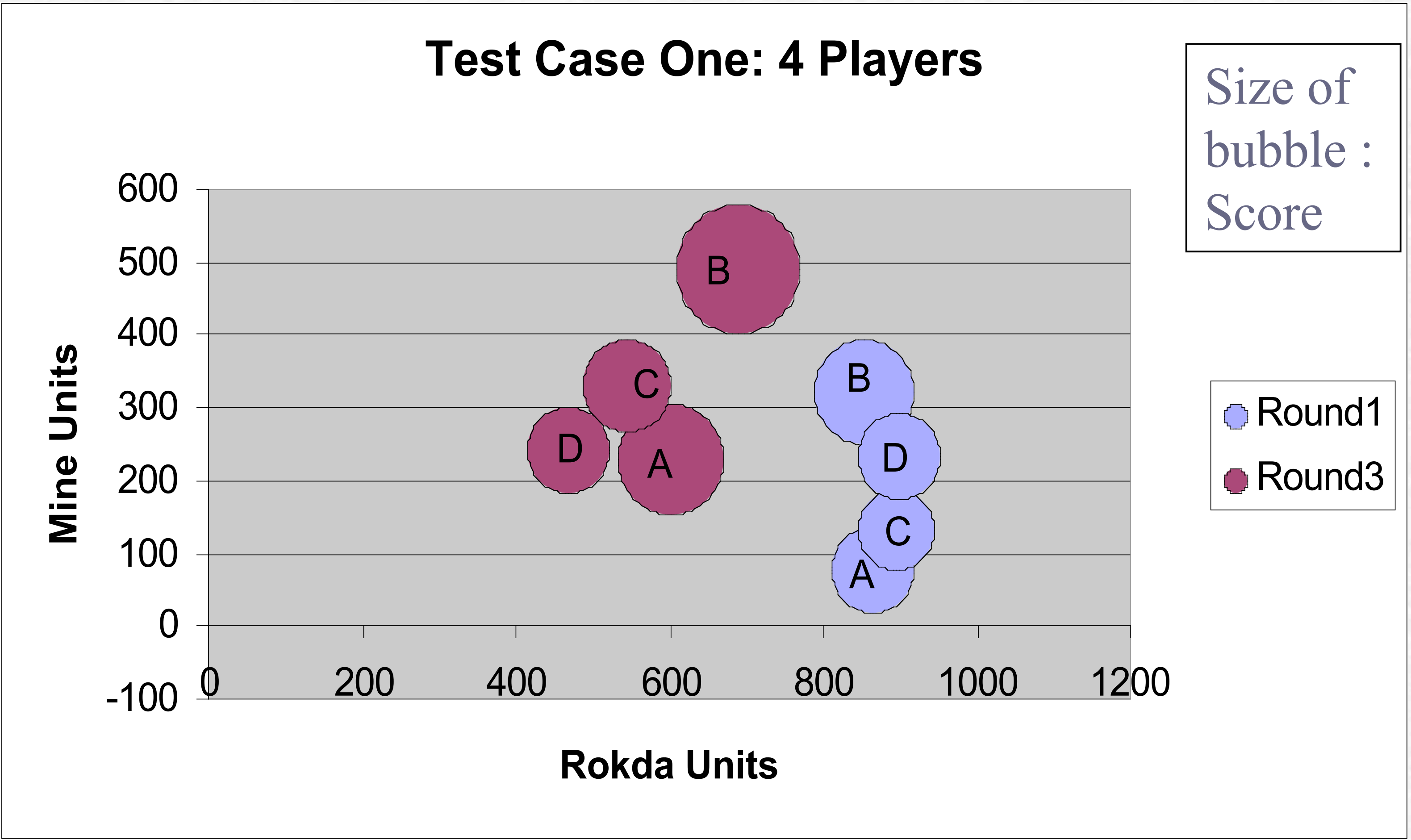
Test Case One: With 2 players

Results



Test Case One: With 4 players

Results



Size of bubble : Score

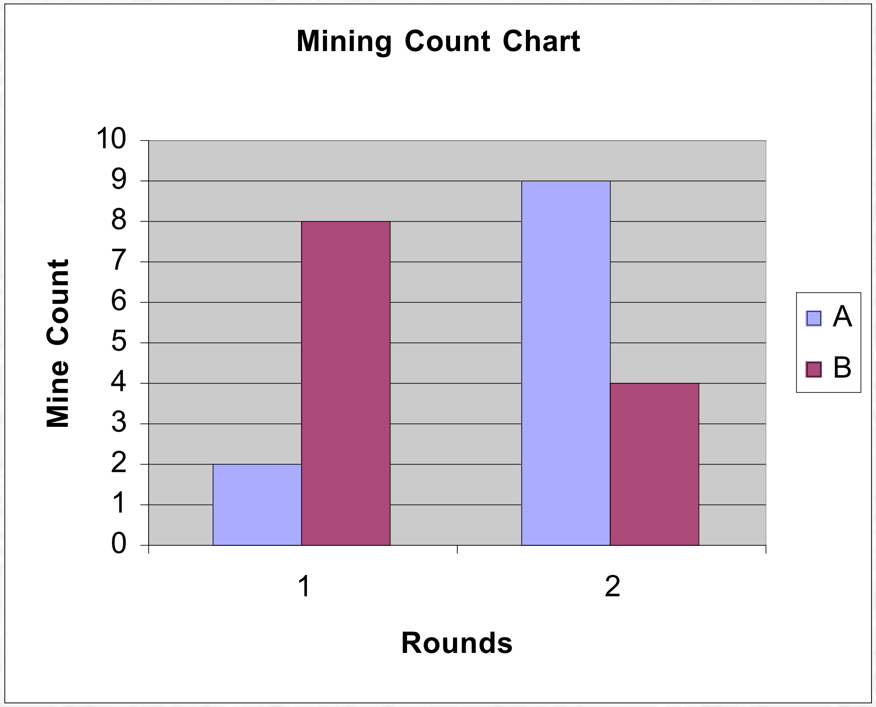
Test Case Two

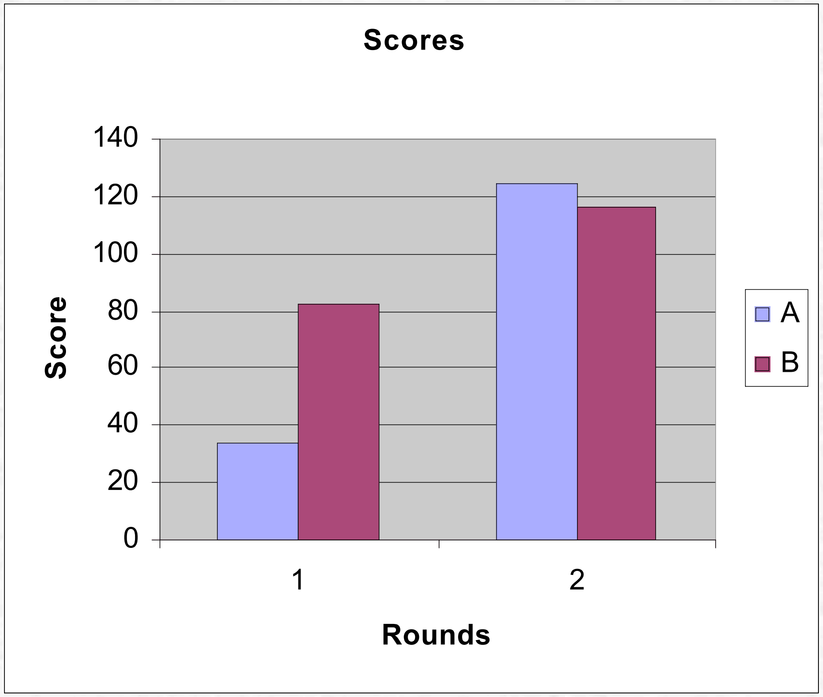
- Description
  - Just deciding to produce a particular product will increase the score
  - For this test case, I chose that if the player chooses to mine, he would have a better score

Score = (Mine_Count * 10) + (Farm_Count *2)
+ (Energy_Count * 2)

Test Case Two:With 2 players

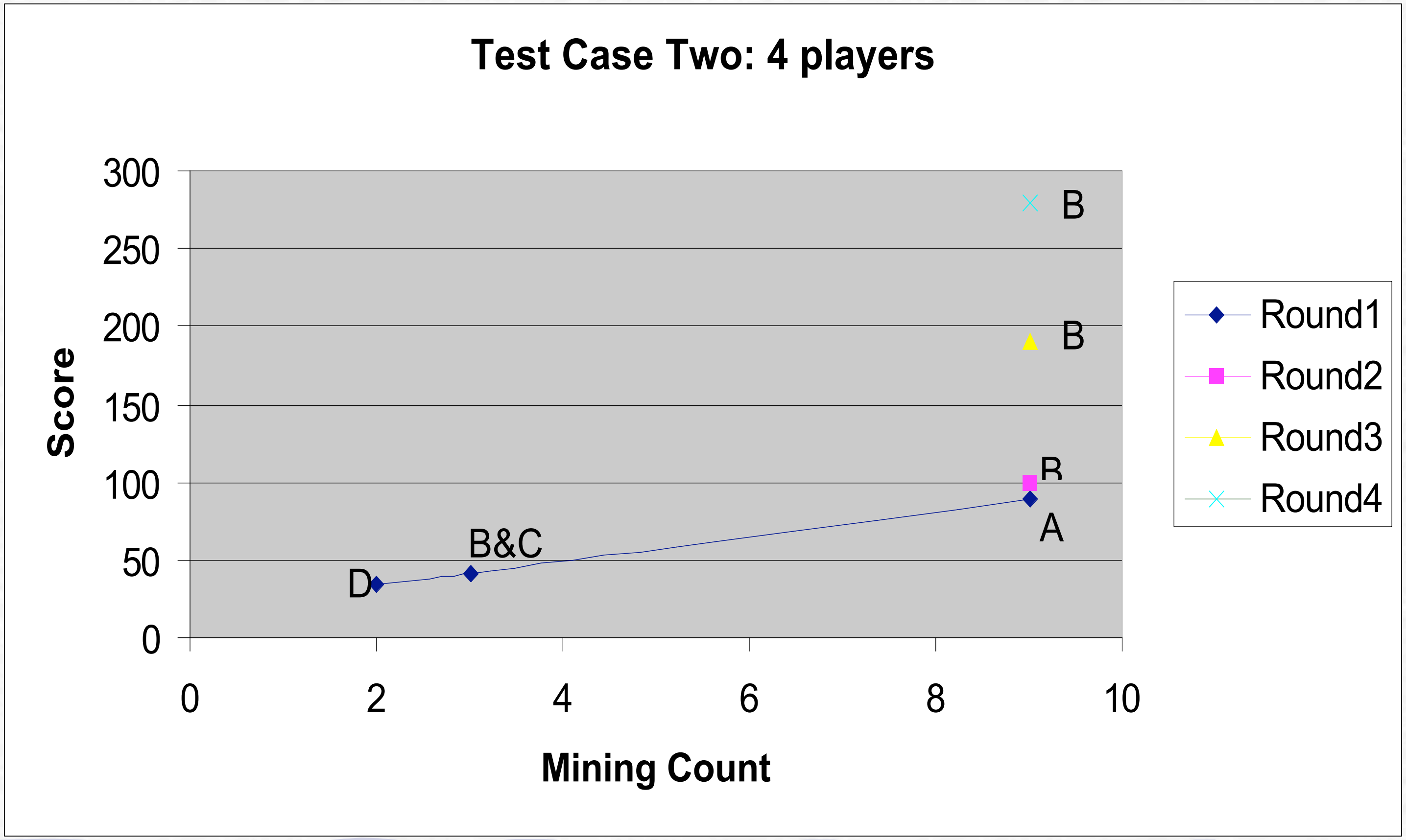
Results





Test Case Two:With 4 Players

Results



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))

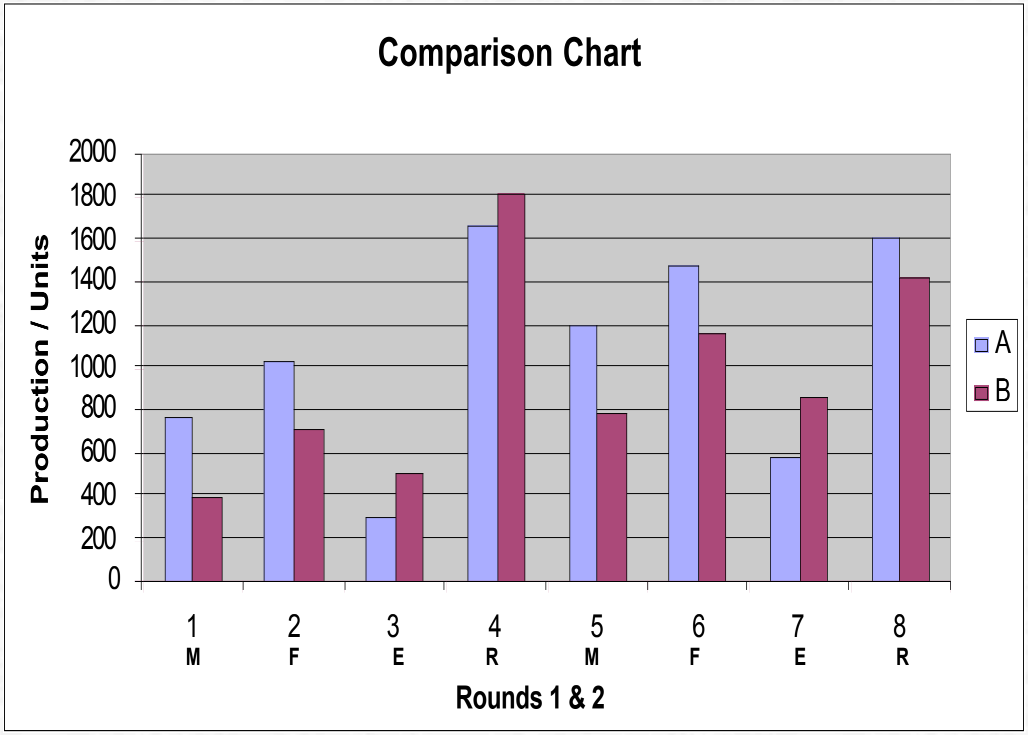
LandExpensesForFarming=(2*(Sum Of Property Values))

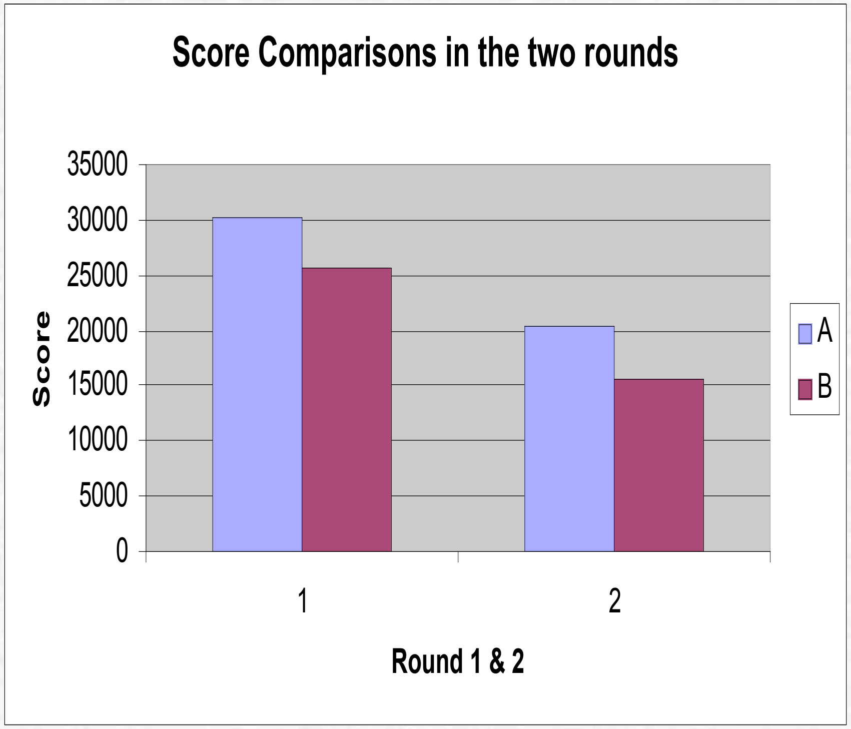
WorkerExpensesForEnergy=(3*(Sum Of Property Values))

LandExpensesForEnergy=(3*(Sum Of Property Values))

Test Case Three: With 2 Players

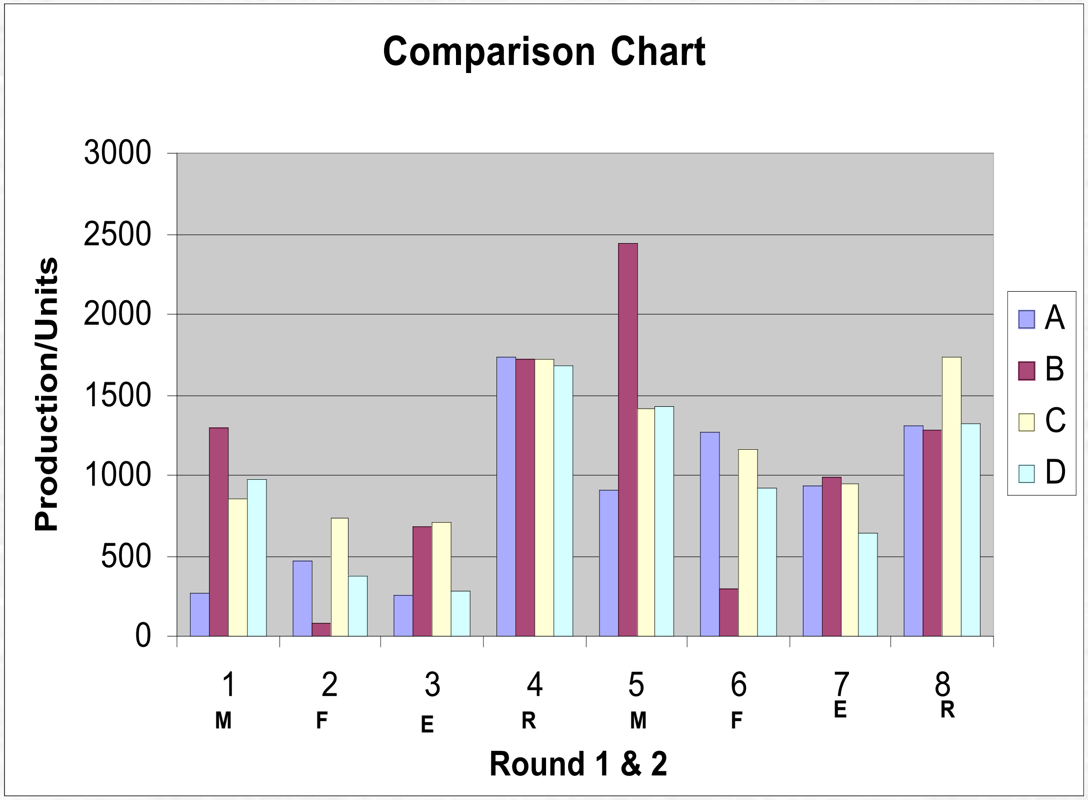
Results

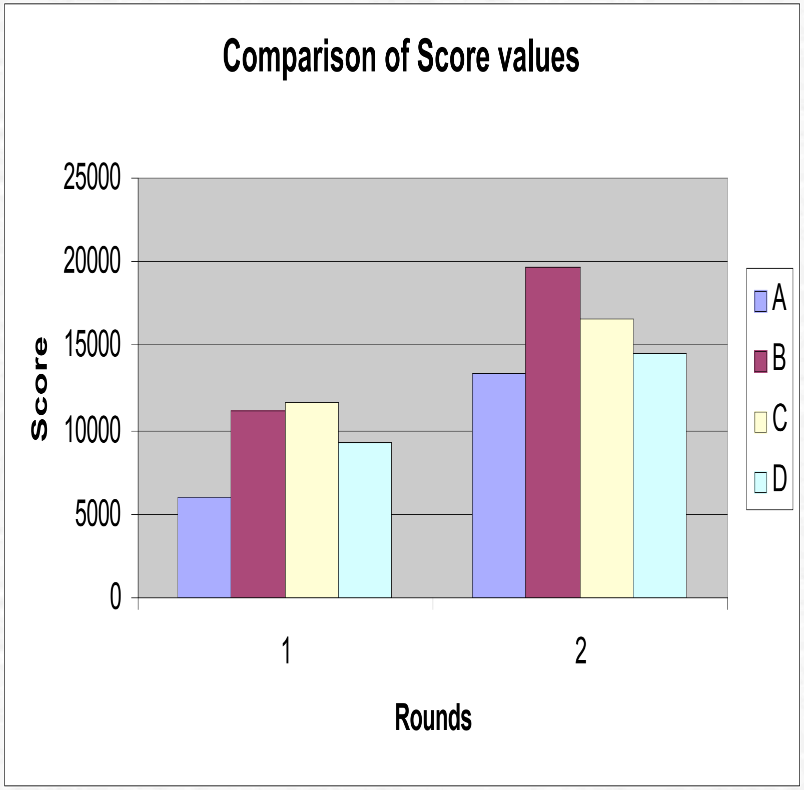




Test Case Three: 4 Players

Results





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.irean.vt.edu/research_workshop_april2003/03_Goad.pdf (Wireless Interactive Training Solutions)
  - 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?