



Project 4 ByteMe

Brian Martin, Stephen Porsche, Evan Salzman

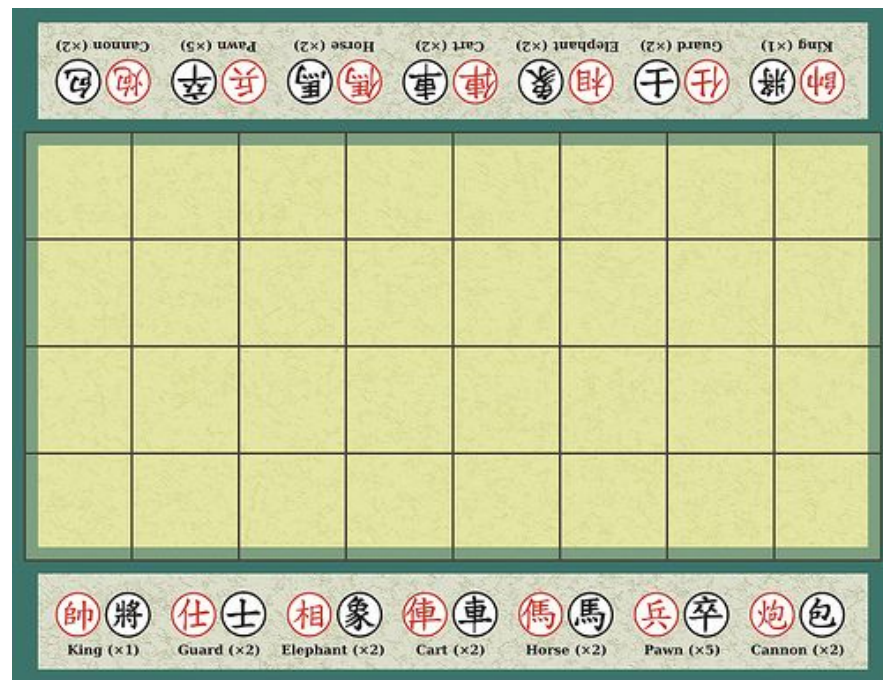
Banqi Rules



- Each player takes turns making a move
- A valid move can be either:
 - Flipping a piece over
 - Moving one space
 - Capturing an enemy piece
- Moving a piece
- Moves can only be up/down/left/right (no diagonal movements)
- Host starts first by flipping over any piece
 - The color of the piece the host turns over is the color of his/her pieces for the game
- Capturing only happens to a piece of equal or lesser rank
 - Exception: General cannot capture soldiers, but soldiers can capture the General
- A player loses when they lose all their pieces and can no longer make any moves

Banqi Board Setup

- 4x8 game board
- 32 total pieces (16 red, 16 black)
- Pieces placed randomly face down
- Pieces have different ranks
- Name, Number per Color of pieces
 - General, 1
 - Advisor, 2
 - Elephant, 2
 - Chariot, 2
 - Horse, 2
 - Cannon, 2
 - Soldier, 5



Development Manual



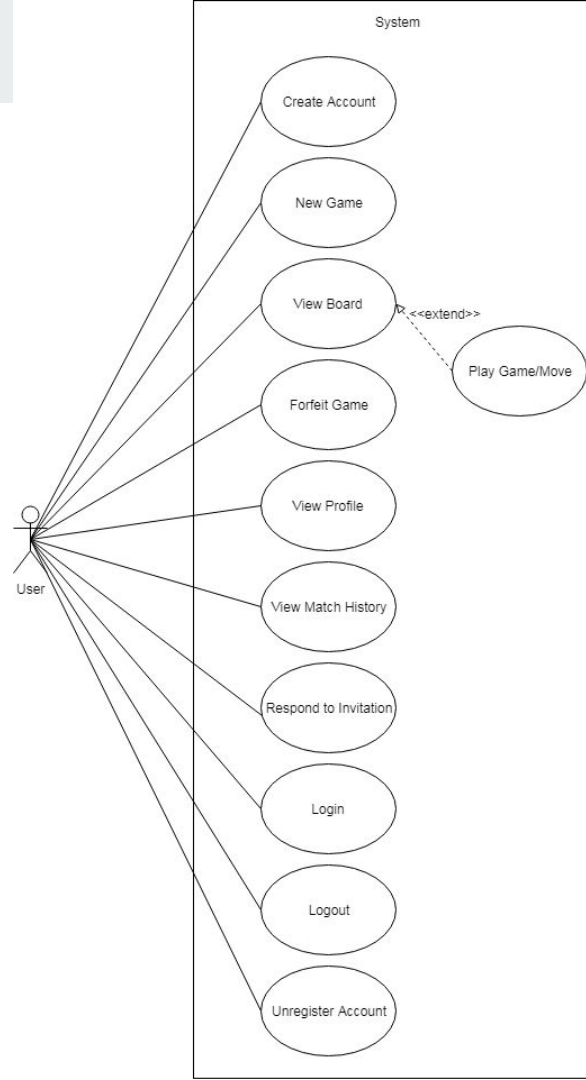
Manual sections:

1. Software Required to Build/Run/Test the System
2. Download the Repository
3. Download and Begin Working on the Project
4. Run the Main Banqi Game Program in Single-Device/Doubl-Device mode
5. Testing the program

Manual:

<https://github.com/FoxXix/cs414-f18-001-ByteMe/blob/master/Sprint%204/Development%20Manual.md>

Use Case Diagram

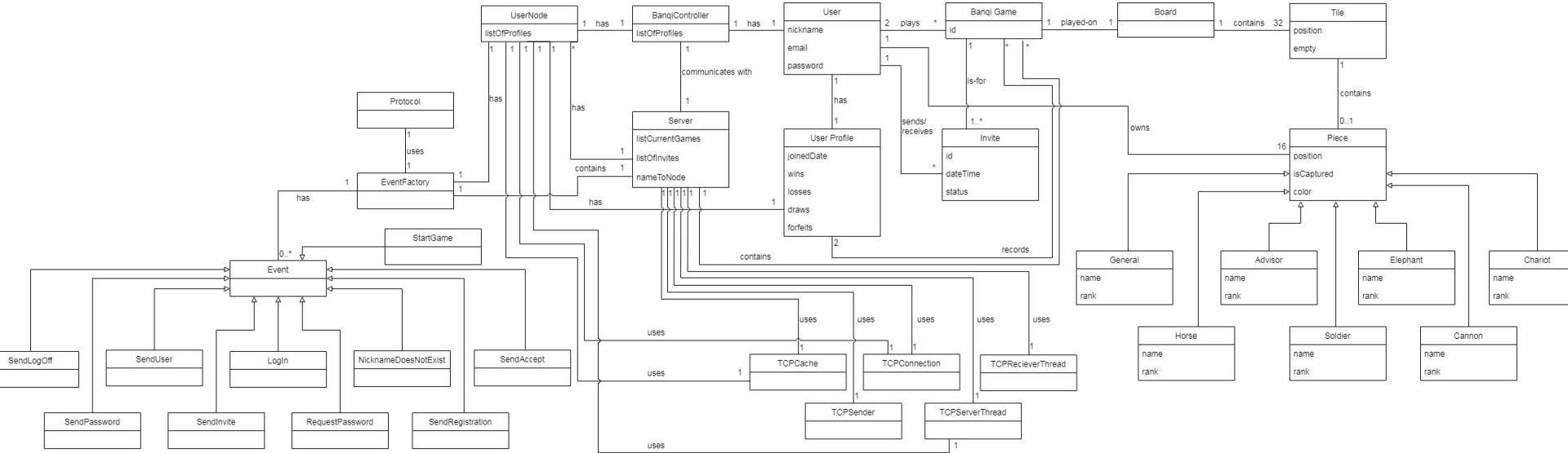


Use Case Document



<https://github.com/FoxXix/cs414-f18-001-ByteMe/blob/master/Sprint%2004/ByteMe%20Use%20Cases%20v3.0.pdf>

Domain Model



Design Document



<https://github.com/FoxXix/cs414-f18-001-ByteMe/tree/master/Sprint%2004>

Traceability Link Matrix



<https://github.com/FoxXix/cs414-f18-001-ByteMe/blob/master/Sprint%2004/Traceability%20Link%20Matrix.pdf>

Testing Document



<https://github.com/FoxXix/cs414-f18-001-ByteMe/blob/master/Sprint%2004/Testing%20Document.pdf>

Refactoring and Design Pattern List



Refactoring Operations:

1. Extracting methods
2. Moving features between objects
3. Replace error code with exception
4. Pull up method

Current Design Patterns:

1. Builder (Creational Design Pattern): BanqiController.java
2. Factory Method (Creational Design Pattern): EventFactory.java

Challenges Faced



- Applying course material (i.e. modeling, design patterns) to the project
- Addressing ambiguity in the project specifications
- Gameplay on two computers
- Working as a group of 3, rather than 5 or 6

Lessons Learned



1. Adapting to change, which can commonly occur when the client's needs and expectations are not solidified until the product is complete.
2. Working with ambiguous specifications (not all features/ functionality were completely clear)
3. Having to forego some features due to having a small group
4. Using teamwork to collectively to implement a Banqi Game system
5. Working as a team is effective not only because it allows for the completion of bigger projects, but also because each individual member has their own skillsets.
6. Implementing refactoring operations and design patterns

Demo



Banqi is a two player game, thus, we will now demonstrate playing Banqi on two different computers.