Chess'N'Chat

Team members:

Adigun Jide Idris 7128525

Chenglong Zhang 6842666

Kun Wang 6713084

Pavithra Keshavamurthy 6827527

Sun liu 6758878

Due: 2nd Feb

# Summary of Project

Chess ‘N’ Chat project is an open source project, a game which is written in java and it implements multiple functionalities of a basic chess online game, it can be compiled and tested on Eclipse. It is a stable project. The current release of the project is in 2011. There were two developers working on this project. Chess game is responsible for creation of graphical entities like icons, characters, squares, board etc. and involves the population of the characters/pieces on the chess board. Players can set their names and their icons to their accounts. This project contains two players as primary actors. This system provides different functionalities for users. For example, help and record of the time, save, load, record moves, and record captured pieces, chat etc. This chat functionality establishes a connection bridge. The combination of Chess’N’Chat application will make the system more interactive and challenging.

# Personas, Actors, and Stakeholders

### John Lennon (main persona)

|  |  |
| --- | --- |
| Image courtesy : Google Images | |
| **Name:** | John Lennon |
| **Age:** | 21 year |
| **Professional:** | Student |
| **Biography:** | John Lennon is 21 year old who looks for different strategies for playing games. He always loves to play games that especially create motivation and challenge. He likes to play games relate to brain teasers which improve concentration and logical thinking. Chess is one of his favorite games. He has good knowledge of using computers and often spends time browsing the web and chatting with people online. He is very enthusiastic in playing different games which need a plan or strategy or critical and rapid thinking for winning the game. Therefore, he is in need of this Chess’N’Chat game which can enhance his memory power and concentration |
| **Goals:** | To play online chess game with communication with friends. |

**Adela Blackburn (additional persona1)**

|  |  |
| --- | --- |
| these-six-regular-people-just-got-invited-to-use-google-glass  Image courtesy : Google Images | |
| **Name:** | Adela Blackburn |
| **Age:** | 24 year |
| **Professional:** | UI designer |
| **Biography:** | Adela Blackburn is a young female UI designer working with google, She is a huge fan of intelligence games and often actively involves with all kinds of online communication software. She often plays games for relax. She always has interests in board games. So she is the potential user of the Chess’N’Chat game. |
| **Goals:** | To learn chess by using online chess game |

**Joyce Lalagos (additional persona2)**

|  |  |
| --- | --- |
| C:\Temp\passport photo.jpg  Image courtesy : Google Images | |
| **Name:** | Joyce Lalagos |
| **Age:** | 23year |
| **Professional:** | Student |
| **Biography:** | Joyce Lalagos a student whose major is software engineering. She always like to try some new applications and at meantime she plays chess since she is a child. She often plays chess with other friends, but she finds out that caring the real chess board with her is really inconvenient. |
| **Goals:** | Interested in trying out the chess online game |

**Actors and Stakeholders**

**Player1**: This is the primary actor involved in starting the game and forms the heart of this Chess’N’Chat game. These players are responsible for playing the game and performing moves on the chess board. There are white and black pieces involved in this game and each one is assigned to each of the actors. The player is at his will to decide which character or piece he/she wants to move on the board. Corresponding to the chat application, player1 takes the role of server and is responsible for initializing the connection with the second player.

**Player2:** This player takes the role of secondary actor and also forms the heart of Chess’N’Chat game. Based on the move of player 1, player 2 decides which piece can be moved so that it will facilitate the capture of the player1 or vice versa. The captured pieces will be stored in the corresponding white and black sides on the

# Informal Use Case

Provide 7 use cases. 3 can be descriptive paragraphs, 4 should show be a number sequential list of steps. **Bold** the noun phrases and underline the verb phrases that become entities and associations, respectively, in your domain diagram. Make two tables with the following headings: 1. noun, synonyms 2. Verb, synonyms.



**Figure 1: Use Case Diagram**

**Player goal use cases**

|  |  |
| --- | --- |
| **Use Case Name** | Connect **System** |
| **Identifier** | **UC-1** |
| **Description** | 1.a Player1 indicates to start a new game as a server  System informs Player1 to wait for connection  Player2 gets connected to the system  System indicates game start  1.b Player1 indicates to start a new game as a client  System prompts to enter the IP address of server  Player1 keys in the correct address  System gets connected successfully  System indicates game start |

|  |  |
| --- | --- |
| **Use Case Name** | Move **Pieces** |
| **Identifier** | **UC-2** |
| **Description** | 1.a System indicates Player1 turn to move  System starts a move timer corresponding to Player1  Player1 selects a particular piece to move within the chess board  Player1 selects the destination square according to chess game rules  1.b System indicates Player2 turn to move  System starts a move timer corresponding to Player2  Player2 selects a particular piece to move within the chess board  Player2 selects the destination square according to chess game rules |

|  |  |
| --- | --- |
| **Use Case Name** | Send **Messages** |
| **Identifier** | **UC-3** |
| **Description** | 1.a System indicates the Player1’s turn to move  System starts a move timer corresponding to Player1  Player1 selects a particular piece to move within the chess board  Player1 selects the destination square according to chess game rules  1.b System indicates the Player2’s turn to move  System starts a move timer corresponding to Player2  Player2 selects a particular piece to move within the chess board  Player2 selects the destination square according to chess game rules |

|  |  |
| --- | --- |
| **Use Case Name** | Record captured **pieces** |
| **Identifier** | **UC-4** |
| **Description** | The **player** makes a move with one **piece**. The move results in the capture of an enemy piece. The **system** move the captured piece to a **“captured pieces”** area. |

|  |  |
| --- | --- |
| **Use Case Name** | Move **Pieces** |
| **Identifier** | **UC-5** |
| **Description** | The **player** select save **game**. The system save the chess board as-it-is. The player makes some moves. The **player** select load **game**. The system turn the chess board reverse to when it saved. |

|  |  |
| --- | --- |
| **Use Case Name** | Flip **board** |
| **Identifier** | **UC-6** |
| **Steps** | 1. **Player** indicates that he wants to **flip** the **board**.  2**. System** flip the **board** and shows the result on the screen.  3. **Player** indicates that he wants to flip the **board** again.  4. **System** flip the **board** again and display the result on the screen. |

|  |  |
| --- | --- |
| **Use Case Name** | Record  **moves** |
| **Identifier** | **UC-7** |
| **Steps** | 1. **Player** move the **piece** on the **board**  2.**System** record the moves  3.**Player** chooses to check the **history**  4.**System** display the **history** of recorded moves |

|  |  |
| --- | --- |
| **Noun** | **Synonyms** |
| Player1 | Client1 |
| Player2 | Client2 |
| Pieces | Characters |
| Turn | Chance |
| Rules | Standards, Regulations |
| Board | Platform |
| Game | entertainment |
| System | structure |
| Captured pieces | Captured characters |
| History | Background |
| Recorded moves | Progress |

**Table 1: Nouns and Synonyms of the system**

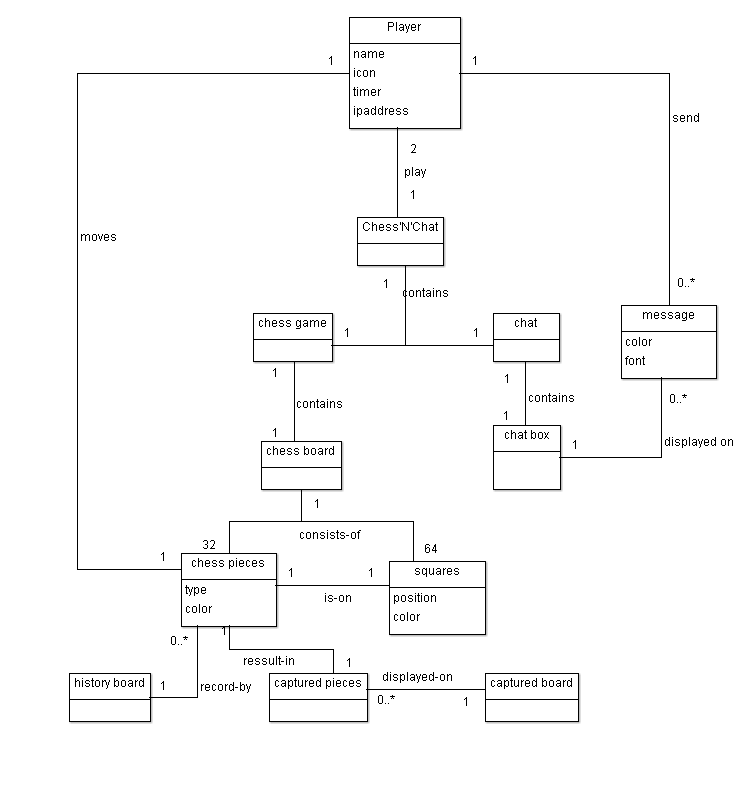
|  |  |
| --- | --- |
| **Verb** | **Synonyms** |
| Connect | Link |
| Move | Migrate |
| Send | Transfer |
| Flip | Exchange |
| Save | Store |
| Load | Retrieve |
| Indicate | Specify |
| Show | Express |
| Display | Demonstrate |
| Capture | Acquire |
| Record | Accomplish |
| Check | Confirm |

**Table 2: Verbs and Synonyms of the system**

**Fully Dressed Use Case**

|  |
| --- |
| **Use Case:** Play Chess and Perform Chat |
| **Primary actor :** Player |
| **Stakeholders and Interests:** Player1 and Player2, Audience |
| **Preconditions:**   * Player should enter his name before start of the game. * Player should choose his icon which is displayed on chess board. * Client and server IP address should be known to make a chat connection |
| **Post Conditions:**   * Moves made by the players are stored in history tab. * Captured pieces are listed in captured white and black pieces. * Player who captures the corresponding king of other player wins |
| **Main Success Scenario**   1. Initialization of Chess board at its standard position 2. First player chooses his side. 3. According to standards, white side is given the first turn to play. 4. After game starts from white side clock starts simultaneously. 5. A player makes a move during his turn to play. 6. Validity of the move is checked. 7. Opponent king checkmate checked 8. A piece is moved on the board accordingly. 9. Turn is given to opponent side. 10. The game is ended if timeout happens of checkmate found 11. Players can also chat online with emotions using smileys. |
| **Extensions ( Alternative Flow of Events)**  1a. If player is not aware of the rules of chess game, player can read instructions from Help Tab.  6a. If the move is not in conformance with rules of the game, the player cannot make a move on the pawn.  10a. Game continues until the checkmate found and clock keeps running.  12a. If IP address incorrect, chat connection is not active. |

# UML Diagram



**Figure 2: Domain diagram of the system.**

The conceptual class **Player** is responsible for playing the chess game and are the primary actors. **Chess’N’Chat** is played by two players. It contains one **chess** **game** and one **chat** for game and chat functionality. The chess game contains one **chess board**. The **chess board** is consists of 32 **chess pieces** and 64 **squares**. One chess piece is on one square. One player can move one chess piece at one time. Multiple moves will be record by one **history board**. One chess piece may result in one captured piece. Multiple **captured pieces** will be displayed on one **captured board**. The **chat** contains one **chat box**. One player can send multiple **messages** and will displayed on the **chat box**.

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Revision** | **Description** | **Author** |
| 20/01/2014 | 0.1 draft | Preliminary version | Pavithra Keshavamurthy |
| 23/01/2013 | 0.2 | Persona and use case update, | Pavithra Keshavamurthy |
| 26/01/2014 | 0.3 | Create Scenarios and editing | Adigun Jide Idris |
| 27/01/2014 | 0.4 | Persona update and create glossary | ChengLong Zhang |
| 30/01/2014 | 0.6 | Draft of UML Diagram and explanation | Pavithra Keshavamurthy |
| 6/02/2014 | 0.7 | Use diagram | Adigun Jide Idris |
| 8/02/2014 | 0.8 | Editing the use cases | Chenglong zhang |
| 8/02/2014 | 0.9 | Domain diagram | Kun Wang |
| 9/02/2014 | 1.0 | Peer view and final | All |

## Glossary

**GUI (graphical user interface)**

Graphical user interface (GUI, sometimes pronounced "gooey") is a type of user interface that allows users to interact with electronic devices through graphical icons and visual indicators such as secondary notation, as opposed to text-based interfaces, typed command labels or text navigation.[1]

**Eclipse**:

It is a multi-language [Integrated development environment](http://en.wikipedia.org/wiki/Integrated_development_environment) (IDE) comprising a base [workspace](http://en.wikipedia.org/wiki/Workspace) and an extensible [plug-in](http://en.wikipedia.org/wiki/Plug-in_%28computing%29) system for customizing the environment.

**Domain**

A domain is a field of study that defines a set of common requirements, terminology, and functionality for any software [program](http://en.wikipedia.org/wiki/Computer_program) constructed to solve a problem in the area of [computer programming](http://en.wikipedia.org/wiki/Computer_programming), known as [domain engineering](http://en.wikipedia.org/wiki/Domain_engineering).

**UML**

Unified modeling language

**Chess**

Chess is a two-player strategy board game played on a chessboard, a checkered game board with 64 squares arranged in an eight-by-eight grid. It is one of the world's most popular games, played by millions of people worldwide at home, in parks, clubs, online, by correspondence, and in tournaments. [2]

**Use case**

In software and systems engineering, a use case is a list of steps, typically defining interactions between a role (known in Unified Modeling Language (UML) as an "actor") and a system, to achieve a goal. The actor can be a human or an external system. [3]

**Stakeholder**

Stakeholder, an entity that can be affected by the results of that in which they are said to be stakeholders, i.e., that in which they have a stake.

## Reference

1. [website] <http://en.wikipedia.org/wiki/Graphical_user_interface>
2. [website] <http://en.wikipedia.org/wiki/Chess>
3. [website] <http://en.wikipedia.org/wiki/Use_case>
4. [website] <http://en.wikipedia.org/wiki/Stakeholder>