Middleware: Ludo game

Group 4

Jean Chan  
Loic Motheu  
Hristo Tarnev

Design document

Contents

[Architecture diagram 3](#_Toc418247198)

[Description of Interfaces 4](#_Toc418247200)

[Methods 4](#_Toc418247201)

[ILudo 4](#_Toc418247202)

[IChat 4](#_Toc418247203)

[Callbacks/Events 4](#_Toc418247204)

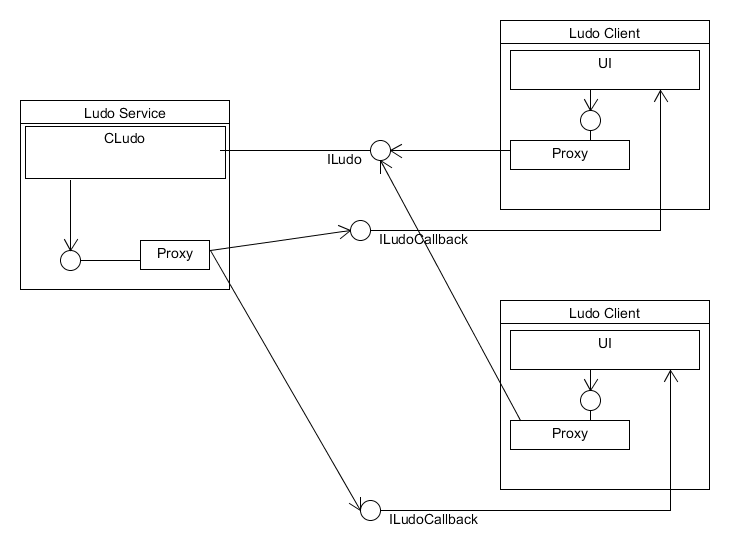
[ILudoCallBack 4](#_Toc418247205)

[IChatCallback: 4](#_Toc418247206)

[Class diagram 5](#_Toc418247207)

[Sequence diagrams 6](#_Toc418247208)

# Architecture diagram



# C:\Users\Loic\Dropbox\MDW\Design\Chat.png

# Description of Interfaces

## Methods

### ILudo

This interface handles all of the player actions during a game.

void MoveToken(Token token, int diceRoll): Moves a desired token x amount of squares where x is the dice roll.

void PlaceToken(Token): When a user rolls a 6, he may place a token on the starting square.

List<Players> PlayerReady(Player player): Marks a player as ready to play

void ChooseColor(Player player): Allows a player to choose a color.

int GetDiceRoll(); Returns the dice roll to every player

void Roll(string userName); Generates a random number between 1 and 6.

void Subscribe(); subscribes other players to the ILudoCallback, so they receive notifications.

void Unsubscribe(); opposite of subscribe.

### IChat

Handles the chatting between players

Void SendMessage(string msg, string sender, string receiver); Can be both public and private, by sending a string message to all players available in the current game, or to a specific player.

Void Start(string name); Checks if player exists in the list of players, and adds them if not found.

Void Stop(string name); Removes the player from the list of players and notifies other players.

## Callbacks/Events

### ILudoCallBack

Handles the notifying the changes within the game to other players.

void PlayerTurn(Player player): Indicates who’s player turn it is.

void TokenMoved(Token token, int diceRoll): When a player has moved a token, now all the

void StartGame(List<Player> players): Needs a list of players who are ready to be able to start the game.

void showDiceRoll(string userName, int diceNumber); Shows the specified user the current dice roll.

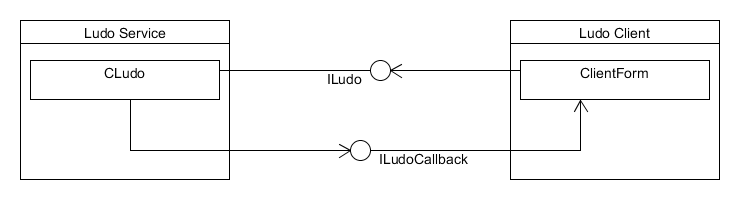
IReceiveChatService

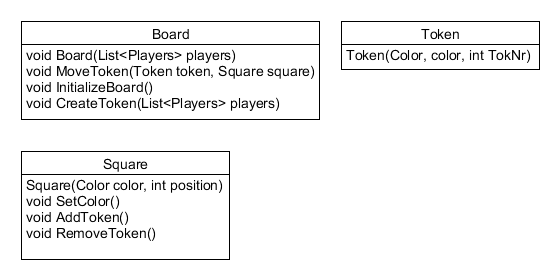
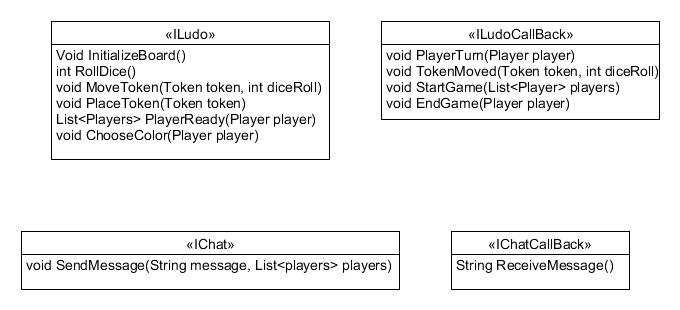
Handles the notifying the other players of the chat messages.

String ReceiveMessage(string msg, string receiver): Returns the message written by other players.

Void SendNames(List<string> names); Returns a list of Names to the other players

# Class diagram

Note: We were unsure whether we needed to have a class diagram for the overall project, of for just the WCF components



# Sequence diagrams

