

# **FOOP - Final Report**

**Team 12:**

B03902017 曹峻寧

B03902048 林義聖

B03902049 陳力宇

B03902087 周景軒

# Relation Between Classes

## 1.Directory Structure:

```
foop15_final10/
├── Makefile
├── README.md
├── bin
├── config
│   └── *.json
├── image
│   ├── beer.png
│   ├── dice.png
│   └── dice_logo.png
├── lib
│   └── json-20151123.jar
└── src
    ├── liardice
    │   ├── ChatRobot.java
    │   ├── ConfigShuffler.java
    │   ├── Dice.java
    │   ├── GameHub.java
    │   ├── GameWindow.java
    │   ├── Main.java
    │   ├── Robot.java
    │   ├── RobotSender.java
    │   └── message
    │       ├── BidMessage.java
    │       ├── CatchMessage.java
    │       ├── ChatMessage.java
    │       ├── ContinueMessage.java
    │       ├── GameStatus.java
    │       ├── ReadyMessage.java
    │       └── RejectedMessage.java
    └── netgame
        └── common
            ├── Client.java
            ├── DisconnectMessage.java
            ├── ForwardedMessage.java
            ├── Hub.java
            ├── ResetSignal.java
            └── StatusMessage.java
```

9 directories, 21 java files(6 of them are from package "netgame.common")

**\*\*\*\*\* We refer to "<http://math.hws.edu/javanotes/c12/s5.html>" and uses its package: "netgame.common"**

## 2.Implementation and Inheritance:

Implementation:

All classes in "liardice/message/" and Dice implements `java.io.Serializable`

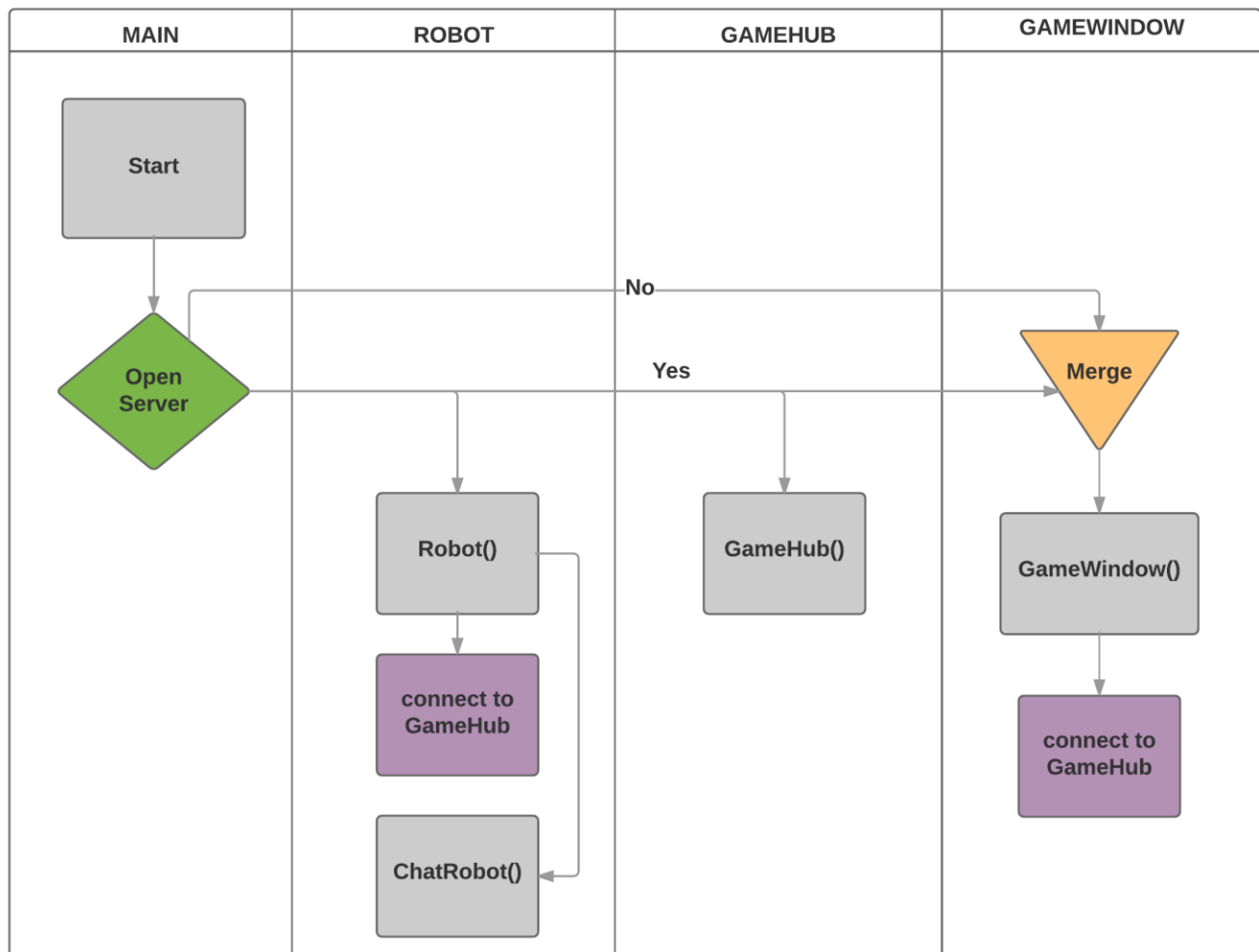
Inheritance:

Class `GameClient`(in `GameWindow`) and `Robot` extends `netgame.common.Client`

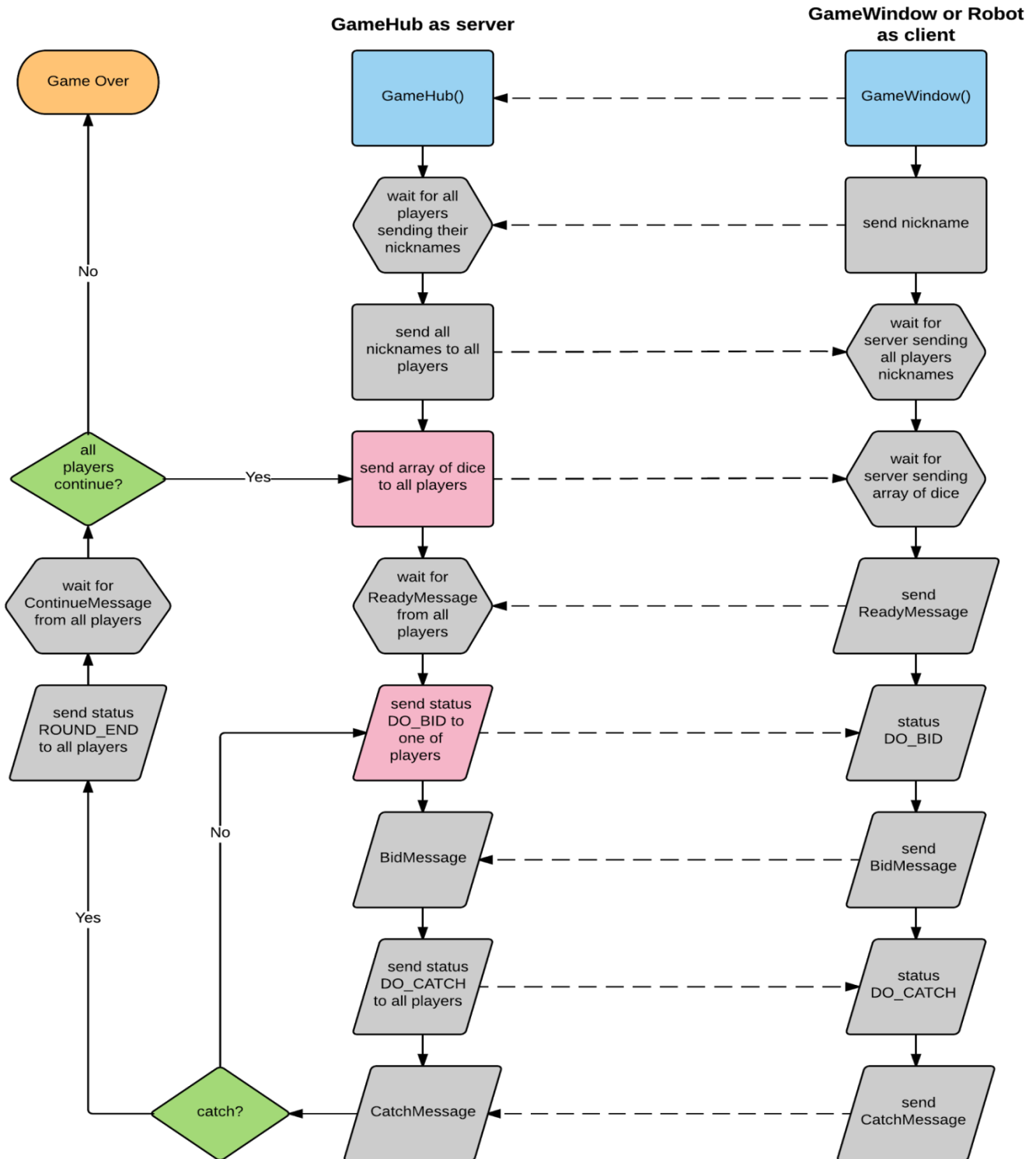
Class `GameHub` extends `netgame.common.Hub`

## 3.Communication Between Classes:

About Main:



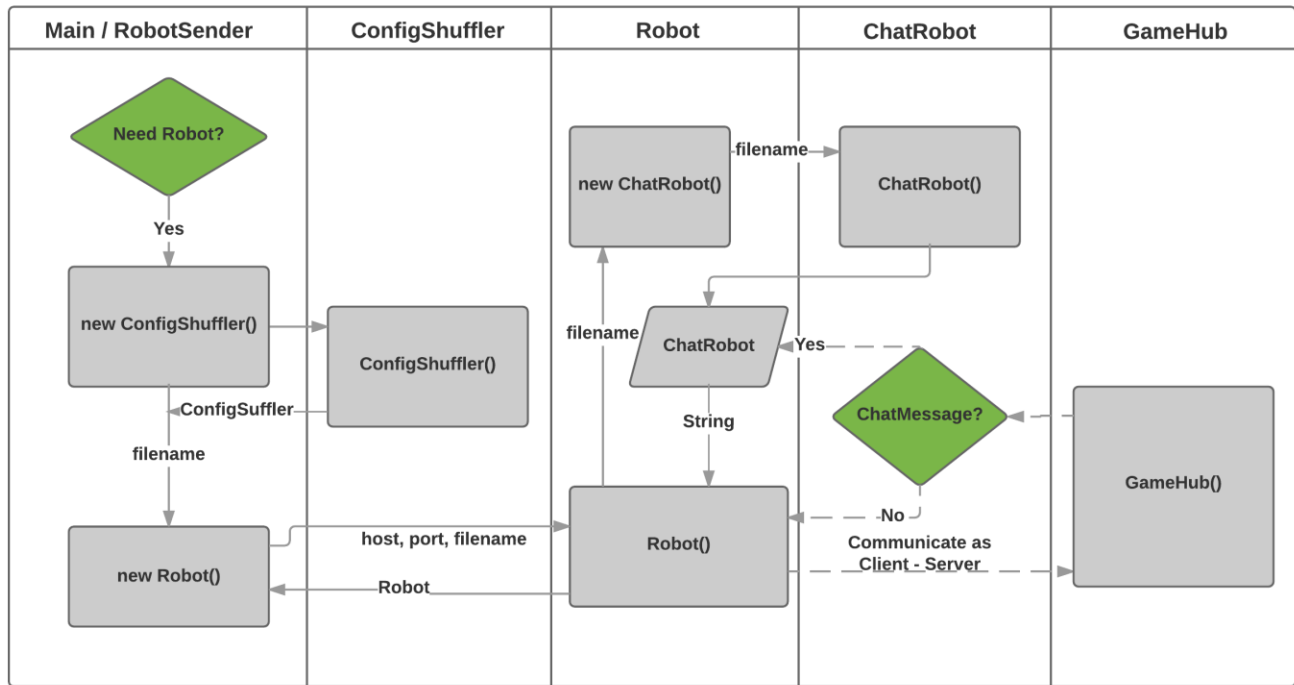
# GameHub and Client:



## Note:

There are also ChatMessage (for players to chat, GameHub just sends it to all players whenever it receives a ChatMessage) and RejectedMessage (GameHub sends it to a player trying to connect a full room.)

# About Robot :



# Advantages & Disadvantages

## Advantages :

### 1. 完成度 :

GUI 設計清楚完整，提供簡潔但明瞭的遊戲指示，讓新手也能一次上手。各種事件、例外狀況皆有做處理（基本上玩不壞）。遊戲豐富度足夠，可支援單機遊戲、自創多個 AI 陪玩，也可建伺服器、網路多人連線遊戲。AI 種類豐富、設計人性化大幅提升遊戲趣味度，甚至可以自製 config file 創造不同性格、不同聊天機制的專屬 AI。

### 2. 架構 :

架構清楚、Class 功能明確。容易修改、擴充。例如 AI、乃至聊天 AI 都是後來加上的設計，但幾乎不用更動原先撰寫的部分。

## Disadvantages :

### 1. 相容性 :

針對 OS X 設計介面，在 Windows / Linux 上無法呈現最佳視覺效果。此外，視窗大小為絕對值，因此若螢幕解析度較低，遊戲視窗將超過螢幕範圍。

### 2. 聊天室字色 :

當 AI 較多時，聊天室訊息量會比較多，可能會使同樣印在其中的遊戲訊息被忽略，雖然遊戲視窗本身也都有提示訊息，不致影響流暢度。不過不能將遊戲訊息和普通聊天訊息標成不同顏色，算是一個小遺憾。

# How To Play

## 一、遊戲流程：

### 1. 輸入暱稱並選擇開新遊戲或連接已存在的遊戲

#### 開新遊戲

填寫真人玩家數量（真人玩家數量需介於 1-6 之間）

填寫機器人玩家數量（總玩家數量需介於 2-10 之間）

填寫要使用的通訊埠（預設為 42857）

#### 連接已存在的遊戲

填寫要連接的網址或 IP（長度需小於 30）

填寫要連接的通訊埠（預設為 42857）



The screenshot shows a window titled "Liar's Dice" with the following content:

**LIAR'S DICE!**

To lie or not to lie.

Please type your nickname (length<11):

☒ Start a new game.

Number of player (1~6):

Number of AI (0~6):  (Total players < 10)

Listen on port:

☐ Connect to existing game.

Host URL:

Port Number:

Buttons: Cancel, OK



The screenshot shows a window titled "Liar's Dice" with the following content:

**LIAR'S DICE!**

To lie or not to lie.

Please type your nickname (length<11):

☐ Start a new game.

Number of player (1~6):

Number of AI (0~6):  (Total players < 10)

Listen on port:

☒ Connect to existing game.

Host URL:

Port Number:

Buttons: Cancel, OK

## 2. 等待玩家到齊後開始遊戲

3. 每回合開始時，每位玩家會獲得該回合的骰子，之後每位玩家依序叫  $X$  個  $Y$ ，即是預料所有參加者中至少有  $X$  個點數為  $Y$  的骰子

4. 在每位玩家叫  $X$  個  $Y$  後，其他玩家可以決定是否要抓：

若有人喊抓：

只有第一個喊抓的玩家能抓。抓就是代表認為叫的玩家在吹牛，若果真如此，則叫的玩家輸，進入下回合；否則，抓的玩家為輸，進入下回合。

若無人喊抓：

則換下一位玩家喊。

## 二、備註：

### a. 遊戲規則

1. 除了每回合第一個叫的玩家可以叫任何的  $X$  個  $Y$  ( $X > 0$ ,  $0 < Y < 7$ ) 外，其他的玩家叫的  $X$  個  $Y$  都必須比前一個玩家「大」。意即假設前一位玩家叫  $X$  個  $Y$ ，而輪到的玩家要叫  $A$  個  $B$ ，則必須符合  $A > X$  或 ( $A = X$  且  $B > Y$ )，才是合法的。

2. 當有玩家要抓時，伺服器會計算骰子總數。如果該回合沒有任何玩家叫過  $X$  個 1，則 1 可以代表任何數字，否則，1 就只能代表 1 自己。當伺服器計算完骰子總數後，會判斷該回合的輸家。

3. 每一回合結束後，伺服器會詢問所有玩家是否繼續遊戲，如果有任何玩家不想繼續，則遊戲結束。遊戲中，如果有任何玩家關閉遊戲，則所有玩家的遊戲都會結束。



## b. 遊戲內容

1. 在遊戲進行中，可以利用聊天室與其他玩家對話，藉此表達自己的心情，或擾亂其他玩家的思考。
2. 點擊左上方的骰子，可以使骰子由小到大排序，點擊右上方的啤酒，可以使骰子隨意排列。
3. 發配骰子時，骰子會依序閃動一陣後才得到真實值。
4. 遊戲中，可以在視窗上方點擊某玩家的名字，即可使其發言不出現在自己的聊天室，再點一次則恢復。

# Responsibility

陳力宇、周景軒負責 GUI 設計與調教和 Client 端

曹峻寧、林義聖負責 Server 端、Protocol 訂定和 AI 設計

	曹峻寧	林義聖	陳力宇	周景軒
Main			v	v
GameWindow			v	v
GUI Design			v	v
GUI Effector			v	
GameHub	v	v		
Robot	v	v		
ChatRobot		v	v	
RobotSender			v	
ConfigShuffler		v		
message/ *	v			
GameStatus	v			
AI Config File	v	v	v	v
Report	v	v		v