

브위님 오목 질문!!



```
gameTimerCount = 0,  
gameTimerPower = false,  
gameTimerShield = 3,
```

[illegible]

```

gameTurn = 1,

gamePosX = 0,
gamePosY = 0,

gameStartTime = "",

gamePlayerList = [],
gamePlayerStone = [],

gameTimerCount = 0,
gameTimerPower = false,
gameTimerShield = 3,

roomName = "",
roomCreat = false,
roomTimerCount = 0,
roomTimerPower = false;
},

restart : function()
{
Game.setMapClear();

gamePlayerStone = [0, 0];

gameFirst = true;

gamePosX = 0;
gamePosY = 0;
gameTurn = 1;

gameTimerCount = 0;
},

/* ----- Timer ----- */

startGameTimer : function()
{
gameTimerPower = true;

new Thread
({
run : function() { try
{
while (gameTimerPower)
{
Thread.sleep(1000);

if (gameTimerCount >= GAME_TIMER_OUT)
{
Game.setNextTurn();

Bot.replyRoom
(

```

```

"시간이 초과되어 자동으로 턴이 넘어갑니다.\n" +
"현재 플레이어 : " + Game.getNowPlayer() + "(" + Game.getNowStone() + ")"
);
}
else
{
gameTimerCount ++;

((gameTimerCount - GAME_TIMER_OUT) == 20) ? Bot.replyRoom("20초 남았습니다.") :
((gameTimerCount - GAME_TIMER_OUT) == 10) ? Bot.replyRoom("10초 남았습니다.") : null;
}
}
}
catch(e)
{
Bot.replyRoom
(
"TIMER ERROR!\n\n" +
"Error Code : " + e
);
}
}).start();
},

startRoomTimer : function()
{
roomTimerPower = true;

new Thread
({
run : function() { try
{
while (roomTimerPower)
{
Thread.sleep(1000);

if ((roomTimerCount >= ROOM_TIMER_OUT) && (!gamePower))
{
Bot.replyRoom("60초가 지나 자동으로 방을 삭제합니다.");
Game.off(); break;
}
else
{
roomTimerCount ++;

((roomTimerCount - ROOM_TIMER_OUT) == 30) ? Bot.replyRoom("30초 후 방이 삭제됩니다.") :
((roomTimerCount - ROOM_TIMER_OUT) == 10) ? Bot.replyRoom("10초 후 방이 삭제됩니다.") : null;
}
}
}
catch(e)
{
Bot.reply
(
"ROOM TIMER ERROR!\n\n" +

```

```
ROOM_TIMER_ERROR +
```

```
"Error : " + e
```

```
);
```

```
}
```

```
)).start();
```

```
},
```

```
stopGameTimer : function()
```

```
{
```

```
gameTimerCount = 0;
```

```
gameTimerPower = false;
```

```
},
```

```
stopRoomTimer : function()
```

```
{
```

```
roomTimerCount = 0;
```

```
roomTimerPower = false;
```

```
},
```

```
/* ----- */
```

```
getMapText : function()
```

```
{
```

```
let posText = ["①","②","③","④","⑤","⑥","⑦","⑧","⑨","⑩","⑪","⑫","⑬","⑭","⑮"],
```

```
vertical = [], horizontal = [], num = 0, i, posX, posY, mapText = "";
```

```
for (i = 0 ; i < gameMapSize ; i ++)
```

```
{
```

```
vertical[i] = posText[i];
```

```
horizontal[i] = posText[i];
```

```
}
```

```
horizontal = "○" + horizontal + "₩n";
```

```
mapText = "" + horizontal;
```

```
for (posX = 0 ; posX < gameMapSize ; posX ++)
```

```
{
```

```
for (posY = 0 ; posY < gameMapSize ; posY ++)
```

```
{
```

```
if (!posY) { mapText += vertical[num ++]; }
```

```
}
```

```
mapText += gameMap[posX];
```

```
if (posX != (gameMapSize - 1)) { mapText += "₩n"; }
```

```
}
```

```
return "₩n₩n" +
```

```
mapText.replace(/./g,"").replace(/0/g,"┌").replace(/1/g,"●").replace(/2/g,"○").replace(/3/g,"┐").replace(/4/g,"└").replace(/5/g,"┬").replace(/6/g,"┴");
```

```
},
```

```
/* ----- */
```

```
printEventMap : function()
```

```
{
```

```

var time = Math.floor((new Date() - gameStartTime) / 1000),
minute = Math.floor(time / 60), second = time % 60,

minuteText = ((String(minute).length > 1) ? minute : '0' + minute) + "분 ",
secondText = ((String(second).length > 1) ? second : '0' + second) + "초",

timeText = (minute == 0) ? secondText + " " : " " + minuteText + secondText,
ballText = (gamePlayerStone[0] == 10 && gamePlayerStone[1] == 9) ? ("흑 " + gamePlayerStone[0] + " 백 " + '0' +
gamePlayerStone[1]) :
((gamePlayerStone[1] > 9) ? ("흑 " + gamePlayerStone[0] + " 백 " + gamePlayerStone[1]) : ("흑 " +
gamePlayerStone[0] + " 백 " + gamePlayerStone[1])),

messageText = "현재 순서 : " + Game.getNowPlayer() + " (" + Game.getNowStone() + ")\\n\\n" + "[ " + timeText +
" : " + ballText + " ]";

Bot.replyRoom
(
messageText + Game.getMapText().substring(1) + "\\n\\n" +
"좌표 ( " + gamePosX + " , " + gamePosY + " ) 에 돌을 두었습니다."
);
},

printWinEvent : function()
{
Bot.replyRoom(Game.getMapText().substring(2, Game.getMapText().length));
Bot.replyRoom("승자는 " + Game.getNextStone() + " 을 두신 [ " + Game.getNextPlayer() + " ] 님 입니다!");
Game.off();
},

/* ----- */

checkStone : function(posX, posY)
{
if (gameFirst)
{
if (posX == 8 && posY == 8)
{
gameFirst = false;
return true;
}
else
{
Bot.replyRoom("첫 수는 가운데에만 둘 수 있습니다. \\n\\n좌표 : ( 8 , 8 )");
return false;
}
}
else if (!posX || !posY)
{
Bot.replyRoom("값을 입력해주세요.");
return false;
}
else if (isNaN(posX) || isNaN(posY))
{
Bot.replyRoom("숫자만 입력해주세요.");
return false;
}
}

```

```

return raise;
}
else if (1 > posX || posX > 15 || 1 > posY || posY > 15)
{
Bot.replyRoom("입력 범위를 초과했습니다\n" + "1 ~ 15 사이의 값을 입력해주세요.");
return false;
}
else if (gameMap[posY - 1][posX - 1] == 1 || gameMap[posY - 1][posX - 1] == 2)
{
Bot.replyRoom("이미 돌이 있습니다.");
return false;
}
else { return true; }
},

checkWin : function()
{
var score = 0, posX, posY;

for (posX = 2 ; posX < (gameMapSize - 2) ; posX++)
{
for (posY = 0 ; posY < gameMapSize ; posY++)
{
if (gameMap[posY][posX - 2] == 1 && gameMap[posY][posX - 1] == 1 && gameMap[posY][posX] == 1 &&
gameMap[posY][posX + 1] == 1 && gameMap[posY][posX + 2] == 1) { score++; }
else if (gameMap[posY][posX - 2] == 2 && gameMap[posY][posX - 1] == 2 && gameMap[posY][posX] == 2 &&
gameMap[posY][posX + 1] == 2 && gameMap[posY][posX + 2] == 2) { score++; }
else if (gameMap[posX - 2][posY] == 1 && gameMap[posX - 1][posY] == 1 && gameMap[posX][posY] == 1 &&
gameMap[posX + 1][posY] == 1 && gameMap[posX + 2][posY] == 1) { score++; }
else if (gameMap[posX - 2][posY] == 2 && gameMap[posX - 1][posY] == 2 && gameMap[posX][posY] == 2 &&
gameMap[posX + 1][posY] == 2 && gameMap[posX + 2][posY] == 2) { score++; }
}
}
for (posX = 2 ; posX < (gameMapSize - 2) ; posX++)
{
for (posY = 2 ; posY < (gameMapSize - 2) ; posY++)
{
if (gameMap[posY - 2][posX - 2] == 1 && gameMap[posY - 1][posX - 1] == 1 && gameMap[posY][posX] == 1 &&
gameMap[posY + 1][posX + 1] == 1 && gameMap[posY + 2][posX + 2] == 1) { score++; }
else if (gameMap[posY - 2][posX - 2] == 2 && gameMap[posY - 1][posX - 1] == 2 && gameMap[posY][posX] == 2
&& gameMap[posY + 1][posX + 1] == 2 && gameMap[posY + 2][posX + 2] == 2) { score++; }
else if (gameMap[posY + 2][posX - 2] == 1 && gameMap[posY + 1][posX - 1] == 1 && gameMap[posY][posX] ==
1 && gameMap[posY - 1][posX + 1] == 1 && gameMap[posY - 2][posX + 2] == 1) { score++; }
else if (gameMap[posY + 2][posX - 2] == 2 && gameMap[posY + 1][posX - 1] == 2 && gameMap[posY][posX] ==
2 && gameMap[posY - 1][posX + 1] == 2 && gameMap[posY - 2][posX + 2] == 2) { score++; }
}
}
return (score > 0) ? true : false;
},

/* ----- */

setRoomCreat : function(room)
{
roomName = room;

```

```

roomCreat = true;

Game.startRoomTimer();
},

setPlayerAdd : function(player)
{
gamePlayerList.push(player);
gamePlayerStone.push(0);
},

setMapStone : function(posX, posY)
{
gamePosX = posX;
gamePosY = posY;

gamePlayerStone[gameTurn - 1] ++;
gameMap[posY - 1][posX - 1] = ((gameTurn == 1) ? 1 : 2);

Game.setNextTurn();
},

setNextTurn : function()
{
gameTimerCount = 0;
gameTurn = (gameTurn == 1) ? 2 : 1;

Game.checkWin() ? Game.printWinEvent() : Game.printEventMap();
},

main : function(room, message, sender)
{
let input = message.split(" ")[0] == "!",
command = message.charAt(0) == ".";

(command) ? Game.command(room, message, sender) : null;

if (gamePower && input)
{
let posX = Number(message.split(" ")[1]),
posY = Number(message.split(" ")[2]), num = 0;

// gamePlayerList.length = 2
for (num = 0 ; num < 2 ; num ++)
{
if (sender == gamePlayerList[num] && (gameTurn - 1) == num)
{
if (Game.checkStone(posX, posY))
{
Game.setMapStone(posX, posY);
}
}
}
}
}
,

```



```

},

getNowPlayer : () => { return gamePlayerList[gameTurn - 1]; },
getNowStone : () => { return ((gameTurn == 1) ? "흑" : "백"); },

getNextPlayer : () => { return gamePlayerList[(gameTurn == 1) ? 1 : 0]; },
getNextStone : function() { return ((gameTurn == 1) ? "백" : "흑"); },

command : function(room, message, sender)
{
var command = message.charAt(0),
input = message.substr(1).split(" ")[0],

printMessage = function(type)
{
switch (type)
{
case 1 : Bot.replyRoom("이미 생성된 방이 있습니다."); break;
case 2 : Bot.reply("생성된 방이 없습니다."); break;
case 3 : Bot.reply("오목 게임이 생성된 채팅방에서만 입력이 가능합니다."); break;
case 4 : Bot.replyRoom("이미 오목 게임이 진행중입니다."); break;
case 5 : Bot.replyRoom("진행중인 게임이 없습니다."); break;
case 6 : Bot.replyRoom("2명 이상이 참가해야 게임 시작이 가능합니다."); break;
case 7 : Bot.replyRoom("방장만 입력이 가능합니다."); break;
case 8 : Bot.replyRoom("중복 참여로 참가가 거부되었습니다."); break;
case 11 : Bot.replyRoom("게임에 참가중인 상태가 아닙니다."); break;
case 12 : Bot.replyRoom("인원이 초과되었습니다.");
}
},

roomManager = gamePlayerList[0],
roomCheck = (roomName == room),
managerCheck = (roomManager == sender),
playerCheck = (gamePlayerList.indexOf(sender) != -1);

if (command == ".") { if (input == "도움말")
{
Bot.reply
(
"< 오목 도움말 >\n\n" +
"● 게임시작 방법 : 생성 > 참가 (자동시작)\n" +
"● 게임 방법 : ! x좌표 y좌표\n" +
"● .판 / .맵 / .종료 / .재시작"
);
}
else if (input == "생성") { if (!roomCreat)
{
Game.setPlayerAdd(sender);
Game.setRoomCreat(room);

Bot.replyRoom
(
"[ " + sender + " ] 님이 오목 게임을 생성하셨습니다.\n\n" +
"게임 참가를 원하시면 ₩.참가₩를 입력 해 주세요."
);
}
}
}

```

```

} else printMessage(1); }
else if (input == "참가") { if (roomCreat) { if (roomCheck) { if (!gamePower) { if(!playerCheck) { if
(gamePlayerList.length < 2)
{
Game.setPlayerAdd(sender);

Bot.replyRoom
(
"[ " + sender + " ] 님이 오목에 참가하셨습니다.\n\n" +
"5초 뒤 게임이 시작됩니다."
);

Game.start();

Bot.replyRoom
(
"게임을 시작합니다!\n\n" +
"< 플레이어 목록 >\n" +
"흑 : " + gamePlayerList[0] + "님\n" +
"백 : " + gamePlayerList[1] + "님\n\n" +
Game.getMapText().substring(2, Game.getMapText().length)
);
} else printMessage(12); } else printMessage(8); } else printMessage(4); } else printMessage(3); } else printMessage(2);
}
else if (input == "판" || input == "맵") { if (roomCreat)
{
Bot.replyRoom(Game.getMapText().substring(2, Game.getMapText().length));
} else printMessage(2); }
else if (input == "종료" || input == "재시작") { if (roomCheck) { if (managerCheck) { if (gamePower)
{
if (input == "종료")
{
Bot.replyRoom("게임이 종료됩니다.");
Game.off();
}
else if (input == "재시작")
{
Bot.replyRoom("현재 인원으로 게임을 재시작합니다.");
Game.restart();
}
} else printMessage(5); } else printMessage(7); } else printMessage(3); }}
},
};

function response(room, message, sender, isGroupChat, replier) { try
{
Bot.replyRoom = (chatting) => { Api.replyRoom(roomName, chatting); };
Bot.reply = (chatting) => { replier.reply(chatting); };

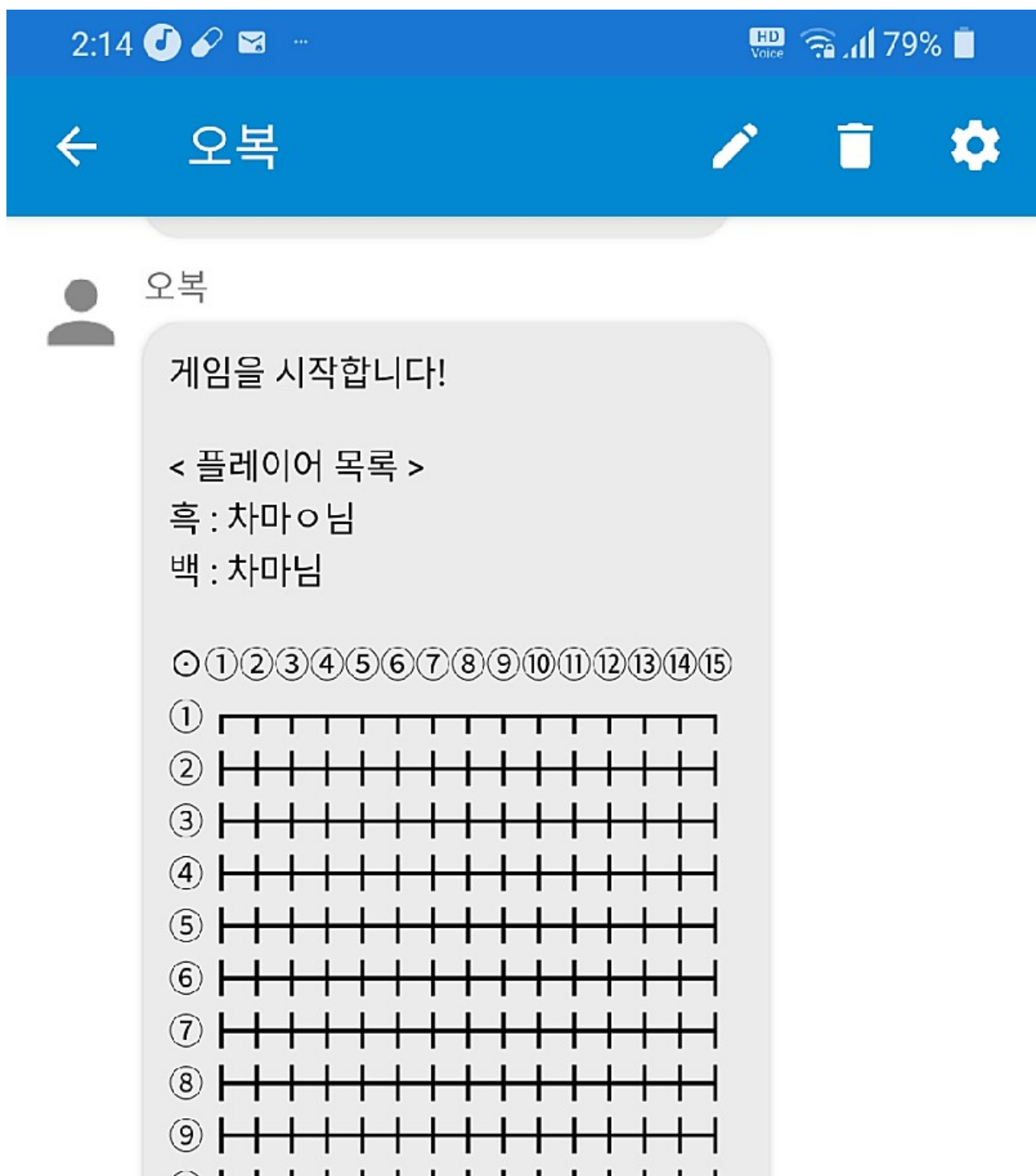
Game.main(room, message, sender);
}
catch(e)
{
Bot.replyRoom
,

```

```
(
"Error : " + e.name + "\n\n" +
"Content : " + e.message + "\n\n" +
"Line : " + e.lineNumber
);
}}
```

```
function onStartCompile()
{
Game.stopRoomTimer();
Game.stopGameTimer();
};
```

여기서 시작도 되고
다 되는데 돌이 안 놓아짐....



- ⑩ | | | | | | | | | | | | | | | | | |
- ⑪ | | | | | | | | | | | | | | | | | |
- ⑫ | | | | | | | | | | | | | | | | | |
- ⑬ | | | | | | | | | | | | | | | | | |
- ⑭ | | | | | | | | | | | | | | | | | |
- ⑮ | | | | ...



전체보기

차마○

! 8,8

차마○

! 8 8



어케해야 하죠?
소스는 다 읽어 봄

! 8 8

이렇게함



맛있는 연어 스테이크님의 게시글 더보기 >

❤ 좋아요 1 💬 댓글 10

🔗 공유 | 신고

댓글 등록순 최신순 🔍

댓글알림 ☐



맛있는 연어 스테이크 작성자

이게 아닌가요?

2020.04.13. 14:16 답글쓰기





카톡우앙

소스가 수정된 게 아니라면 기본 둘 두는 명령어는 ! X Y 입니다.

2020.04.13. 14:54 답글쓰기



맛있는 연어 스테이크 작성자

네 저도 그건...

2020.04.13. 14:56 답글쓰기



카톡우앙

맛있는 연어 스테이크 그러면 소스 재다운로드하세요.

2020.04.13. 14:58 답글쓰기



맛있는 연어 스테이크 작성자

카톡우앙 넵

2020.04.13. 15:12 답글쓰기



한국인

소스는 지우고 링크로 바꿔주세요

2020.04.13. 17:02 답글쓰기



맛있는 연어 스테이크 작성자

카카오톡 링크만 남길까요 —?

2020.04.13. 17:16 답글쓰기



한국인

맛있는 연어 스테이크 ?

2020.04.13. 17:34 답글쓰기



한국인

맛있는 연어 스테이크 소스링크요

2020.04.13. 17:35 답글쓰기



맛있는 연어 스테이크 작성자

한국인 넵

2020.04.13. 18:38 답글쓰기

Hibot

댓글을 남겨보세요



등록

글쓰기

답글

목록

▲ TOP


브위님 오목 질문!! 🐼 [2]

맛있는 연어 스테이크 2020.04.14.

브위님 오목 질문!! 🐼 [10]

맛있는 연어 스테이크 2020.04.13.

이 카페 인기글




봇을쓰는데 메세지가 두번

예전에 어떤분이 카톡봇을 이용하여 카페에 게시글을 올렸었는데

리페

♡0 💬1



태양, 달 정보 구현 완료



링크를 포함한 글 홍보시 링크 분석(?)하기

elif eval 질문

♡0 💬2

Kiri

♡0 💬5

path질문

젤리파덕

♡0 💬2



틱택토 (Player vs Player)

도미 doami2005

반가워요

♡0 💬5

천방지축하연

♡0 💬4

오픈이발 괴롭히기(메봇R)

ERROR

♡0 💬7