| 소스 질문 | >

어떤걸 수정해야 오목판 크기를 설정할수있죠?

김지훈 챗봇 입문자 1:1 채팅
2020.01.10. 02:16 조회 143

댓글 4 URL 복사

질문할 어플 (새자봇, 젤브봇, 메신저봇, 닥토봇, 기타)
=> 메신저봇

질문할 소스

```
=> const scriptName = "오목";
function response(room, msg, sender, isGroupChat, replier, imageDB, packageName) {
importPackage(java.lang);
importPackage(java.io);
const GAME_TIMER_OUT = 30;
const ROOM_TIMER_OUT = 60;
let gameMap = [
[' ┌',3,3,3,3,3,3,3,3,3,3,3,3,3,'┐ '],
  [6,0,0,0,0,0,0,0,0,0,0,0,0,0,4],
  [6,0,0,0,0,0,0,0,0,0,0,0,0,0,4],
  [6,0,0,0,0,0,0,0,0,0,0,0,0,0,4],
  [6,0,0,0,0,0,0,0,0,0,0,0,0,0,4],
  [6,0,0,0,0,0,0,0,0,0,0,0,0,0,4],
  [6,0,0,0,0,0,0,0,0,0,0,0,0,0,4],
  [6,0,0,0,0,0,0,0,0,0,0,0,0,0,4],
  [6,0,0,0,0,0,0,0,0,0,0,0,0,0,4],
  [6,0,0,0,0,0,0,0,0,0,0,0,0,0,4],
  [6,0,0,0,0,0,0,0,0,0,0,0,0,0,4],
  [6,0,0,0,0,0,0,0,0,0,0,0,0,0,4],
  [6,0,0,0,0,0,0,0,0,0,0,0,0,0,4],
  [6,0,0,0,0,0,0,0,0,0,0,0,0,0,4],
[' └',5,5,5,5,5,5,5,5,5,5,5,5,5,'┘ ']];
gameMapSize = 15,
gamePower = false,
gameFirst = true,
gameTurn = 1,

gamePosX = 0,
gamePosY = 0,

gameStartTime = "",

gamePlayerList = [],
gamePlayerStone = [],

gameTimerCount = 0,
gameTimerPower = false,
gameTimerShield = 3,
roomName = "",
roomCreat = false,
roomTimerCount = 0,
roomTimerPower = false;

const Bot = {};
const Game =
{
setMapClear : function()
{
// initialization
gameMap = [
        [' ┌',3,3,3,3,3,3,3,3,3,3,3,3,3,'┐ '],
          [6,0,0,0,0,0,0,0,0,0,0,0,0,0,4],
          [6,0,0,0,0,0,0,0,0,0,0,0,0,0,4],
          [6,0,0,0,0,0,0,0,0,0,0,0,0,0,4],
          [6,0,0,0,0,0,0,0,0,0,0,0,0,0,4],
          [6,0,0,0,0,0,0,0,0,0,0,0,0,0,4],
          [6,0,0,0,0,0,0,0,0,0,0,0,0,0,4],
          [6,0,0,0,0,0,0,0,0,0,0,0,0,0,4],
          [6,0,0,0,0,0,0,0,0,0,0,0,0,0,4],
          [6,0,0,0,0,0,0,0,0,0,0,0,0,0,4],
          [6,0,0,0,0,0,0,0,0,0,0,0,0,0,4],
          [6,0,0,0,0,0,0,0,0,0,0,0,0,0,4],
          [6,0,0,0,0,0,0,0,0,0,0,0,0,0,4],
          [6,0,0,0,0,0,0,0,0,0,0,0,0,0,4],
        [' └',5,5,5,5,5,5,5,5,5,5,5,5,5,'┘ ']];
},
start : function()
{
Game.stopRoomTimer();
Thread.sleep(5000);
gameStartTime = new Date();
Game.startGameTimer();
},
off : function()
{
Game.stopRoomTimer();
Game.stopGameTimer();
Game.setMapClear();
gameMapSize = 15,
gamePower = false,
gameFirst = true,
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" +
"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,"┤ ").repl
ace(/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;
}
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 생성하였습니다.₩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();
};

}
```

질문내용
=> 어떤 부분을 수정해야 오목판 크기를 설정할수있죠?

(말머리 등록 해주세요!)

김지훈님의 게시글 더보기 >

좋아요 **0** 댓글 **4** 공유 신고

댓글 **등록순** 최신순 댓글알림

카톡우앙
무슨 소리죠?
2020.01.10. 04:26 답글쓰기

AlphaDo
누가 봐도 오목판처럼 생긴 이차원 배열을 수정하시면 될거긴 한데
알고리즘에 따라 오류가 나던지, 아니던지가 되겠네요
2020.01.10. 12:13 답글쓰기

AlphaDo
그 밑에 맵 사이즈도..
2020.01.10. 12:14 답글쓰기

산하
ㅇㅅㅇ
2020.01.10. 12:43 답글쓰기

Hibot
댓글을 남겨보세요
등록

글쓰기 답글 목록 ▲ TOP

'| 소스 질문 |' 게시판 글

이 게시판 새글 구독하기

소스질문!! [4]	q지구p	2020.01.10.
[해결됨] 일치갯수 확인 [8]	프로그	2020.01.10.
어떤걸 수정해야 오목판 크기를 설정할수있죠? [4]	김지훈	2020.01.10.
끝말잇기 소스 리로드가 안되고있어요ㅠㅠ [2]	김지훈	2020.01.09.
[질문] 정규식에서 괄호의 역할은? [6]	AlphaDo	2020.01.09.

1 2 3

전체보기

이 카페 인기글

공식 카카오톡 옾챗

Lunar

♡0 💬9

path질문

젤리파덕

♡0 💬2

가르치기 리로드..

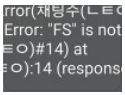
Milk2

♡0 💬9

반가워요.

천방지축하연

♡0 💬4



```
rror(채팅수(ㄴㅌㅇ
Error: "FS" is not
ㅌㅇ)#14) at
ㅌㅇ):14 (respons
```

채팅순위봇이 갑자기 안되네요..ㅠ

안녕하세요

KHP

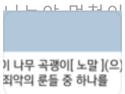
♡0 💬6

kakao.py에 대해서

undefined

♡0 💬13

나는야 멍청이



이 나무 곡괭이[노말](으)
최악의 룬들 중 하나를

5000분의 1을 얻은 사람

메신저봇 삭제버튼 이렇게 안될까요?

1 2 3 4 5