```
/*
    File name: CasinoCraps.js
    Author: Hyewon Choi
    Date created: July 31, 2018
    Version: 0.1
    Copyright: 1.0
    Purpose: Playing Casino Craps game
    Description: It is a dice game. At the beginning of the game, the player rolls a
    pair of dice and computes the total. If the total is 2, 3, or 12 (called
    "craps"), the player loses. If the total is 7 or 11 (called a "natural"), the
    player wins.
    If the total is any other number, that number becomes the "point". From here,
    the player keeps rolling the dice until (a) the point comes up again, in which
    case the player wins, or (b) a 7 appears, in which case the player loses. The
    numbers 2,3,11, and 12 no longer have special significance after the first rolls.
*/

// an Array includes six icons of six value of a dice.
var dicesArray = [
    ["<i class='fas fa-dice-one'></i>"],
    ["<i class='fas fa-dice-two'></i>"],
    ["<i class='fas fa-dice-three'></i>"],
    ["<i class='fas fa-dice-four'></i>"],
    ["<i class='fas fa-dice-five'></i>"],
    ["<i class='fas fa-dice-six'></i>"]
];
var firstDice; // a value of first dice. 0-5 represent each of 1-6.
var secondDice; // a value of second dice. 0-5 represent each of 1-6.
var sumOfDices= 0; // sum of values of first dice and second dice
var firstTryYN= true; // if this try is first try of a game or not.
var gameOver= false;  // if a round can be continued.
var point; //a point which a player made at first try and a player should make to
win after first try.

// rollDices(): set dices and figure sum of them out.
function rollDices() {
    if(!gameOver) {
        sumOfDices= 0;
        var rndNum= Math.floor(Math.random()*6);
        sumOfDices += (rndNum+1);
        firstDice= rndNum;
        //firstDice= dicesArray[rndNum];

        rndNum= Math.floor(Math.random()*6);
        sumOfDices += (rndNum+1);
        secondDice= rndNum;
        //secondDice= dicesArray[rndNum];

        drawDice(1, firstDice);
        drawDice(2, secondDice);

        calResult();
        }
}

// calResult(): calculate the result of a try.
function calResult() {
    var messageBox= document.getElementById("messageBox");

    if(firstTryYN) {
        // first try : 2,3,12 -> Lose / 7,11 -> win / other numbers -> point.
        firstTryYN= !firstTryYN;
        if(sumOfDices == 2 || sumOfDices == 3 || sumOfDices == 12) {
            gameOver= true;
            messageBox.innerHTML= "<p>That's craps. You LOSE!</p>";
        } else if(sumOfDices == 7 || sumOfDices == 11) {
            gameOver= true;
            messageBox.innerHTML= "<p>That's a natural. You WIN!</p>";
        }
        else {
            point= sumOfDices;
            messageBox.innerHTML= "<p>Rolling dices: "+(firstDice+1)+" + "+(secondDice+1)+" = "+sumOfDices+"</p>";
            messageBox.innerHTML += "<p>Your point is "+point+"</p>";
        }
    }
    else {
        // 7 -> Lose / point -> Win / others -> roll again
        if(sumOfDices == 7) {
            gameOver= true;
            messageBox.innerHTML += "<p>Rolling dices: "+(firstDice+1)+" + "+(secondDice+1)+" = "+sumOfDices+"</p>";
            messageBox.innerHTML += "<p>That's a 7. You LOSE!</p>";
        } else if(sumOfDices == point) {
            gameOver= true;
            messageBox.innerHTML += "<p>Rolling dices: "+(firstDice+1)+" + "+(secondDice+1)+" = "+sumOfDices+"</p>";
            messageBox.innerHTML += "<p>You made your point. You WIN!</p>";
        } else {
            messageBox.innerHTML += "<p>Rolling dices: "+(firstDice+1)+" + "+(secondDice+1)+" = "+sumOfDices+"</p>";
        }
    }
}

// drawDice(diceNum, diceToBeDrawn): draw dices depends on what number each dice has.
function drawDice(diceNum, diceToBeDrawn) {
    var dice;
    if(diceNum == 1)
        dice= document.getElementById("firstDice");
    else
        dice= document.getElementById("secondDice");

    dice.innerHTML= dicesArray[diceToBeDrawn];
    if(sumOfDices == 10 || sumOfDices == 12) {
        if(document.getElementById("sumOfDices"))
            document.getElementById("sumOfDices").id="sumOfDices10n12";
        document.getElementById("sumOfDices10n12").innerText= sumOfDices;
    }
    else {
        if(!document.getElementById("sumOfDices"))
            document.getElementById("sumOfDices10n12").id="sumOfDices";
        document.getElementById("sumOfDices").innerText= sumOfDices;
    }
}

// startNewGame(): initialize variables and messages.
function startNewGame() {
    firstTryYN= true;
    gameOver= false;
    document.getElementById("firstDice").innerHTML="";
    document.getElementById("secondDice").innerHTML="";
    document.getElementById("sumOfDices").innerHTML="";
    document.getElementById("messageBox").innerHTML="<p>Game start! Roll the dices.</p>";
}

/*
    File name: TicTacToe.js
    Author: Hyewon Choi
    Date created: July 31, 2018
    Version: 0.1
    Copyright: 1.0
    Purpose: Playing Tic Tac Toe game
    Description: It is a game for two players, X and Y,
    who take turns marking a character the space in a 3*3 gameboard.
    The player who succeeds in placing three of their marks in a horizontal, vertical, or diagonal row wins the game.
*/

var player= "X"; // which does player have a turn to play
var gameOver= false; // if a round can be continued.
var cntClick= 0; // count valid clicks until 9 to figure out if a game ends as draw

// clickBoard(i,j): actions when a cell of the game board is clicked.
function clickBoard(i,j) {
    var boardValue= document.getElementById("board"+i+j);
    if(gameOver)
    {
        return true;
    }
    else if(boardValue.innerText == "Y" || boardValue.innerText == "X") {
        // when a player clicks a cell already clicked.
        document.getElementById("message").innerText= "Choose another";
        return true;
    }

    // if it was turn of a player "X"
    if(player == "X") {
        player= "Y";
        cntClick++;
        boardValue.innerText= "X";
        boardValue.style.background= "red";
        document.getElementById("message").innerText= "Player Y go!";
    } else { // if it was turn of a player "Y"
        player= "X";
        cntClick++;
        boardValue.innerText= "Y";
        boardValue.style.background= "blue";
        document.getElementById("message").innerText= "Player X go!";
    }
    checkWin();

    // if number of valid clicks is 9 but game haven't over.
    if(cntClick == 9 && !gameOver)
    {
        document.getElementById("message").innerText= "Draw!";
        gameOver= true;
        return true;
    }
}

// checkWin(): if a player take 3 consecutive cells vertically/horizontally/diagonally
function checkWin() {
    for(var i=0;i<3;i++) {
        if((document.getElementById("board"+i+0).innerText.trim() != "T")&&
                (document.getElementById("board"+i+0).innerText ==
                 document.getElementById("board"+i+1).innerText) &&
                (document.getElementById("board"+i+0).innerText ==
                 document.getElementById("board"+i+2).innerText)) ||

             ((document.getElementById("board"+0+i).innerText.trim() != "T")&&
                (document.getElementById("board"+0+i).innerText ==
                 document.getElementById("board"+1+i).innerText) &&
                (document.getElementById("board"+0+i).innerText ==
                 document.getElementById("board"+2+i).innerText))) {
            document.getElementById("message").innerText= (player=="X")?"Player Y wins!":"Player X wins!";
            gameOver= true;
            return true;
        }
    }

    if(((document.getElementById("board00").innerText==
             document.getElementById("board11").innerText) &&
            (document.getElementById("board00").innerText==
             document.getElementById("board22").innerText)) ||

         ((document.getElementById("board02").innerText==
             document.getElementById("board11").innerText) &&
            (document.getElementById("board02").innerText==
             document.getElementById("board20").innerText))) {
        document.getElementById("message").innerText= (player=="X")?"Player Y wins!":"Player X wins!";
        gameOver=true;
        return true;
    }
}

// startNewGame(): initialize the game board and variables.
function startNewGame() {
    document.getElementById("board00").innerText= "T";
    document.getElementById("board01").innerText= "I";
    document.getElementById("board02").innerText= "C";
    document.getElementById("board10").innerText= "T";
    document.getElementById("board11").innerText= "A";
    document.getElementById("board12").innerText= "C";
    document.getElementById("board20").innerText= "T";
    document.getElementById("board21").innerText= "O";
    document.getElementById("board22").innerText= "E";

    for(var i=0;i<9;i++)
        document.getElementsByClassName("gameBoard")[i].style.backgroundColor= "white";

    document.getElementById("message").innerText= "Player X go!";
    player= "X";
    gameOver=false;
    cntClick=0;
}

/*
    File name: PickAPair.js
    Author: Hyewon Choi
    Date created: July 31, 2018
    Version: 0.1
    Copyright: 1.0
    Purpose: Playing Pick A Pair game
    Description: It is a game requires players to match same elements.
    It shows whole board for 10 secs at first.
*/

// symbols of 18 pairs are used for games.
var groupOfSymbols=
["\★","\☆","\♥","\♡","\♠","\♤","\♣","\♧","\♨","\♬","\☎","\♀","\♂","\€","\Ω","\∞","\☞","\☜",
"\★","\☆","\♥","\♡","\♠","\♤","\♣","\♧","\♨","\♬","\☎","\♀","\♂","\€","\Ω","\∞","\☞","\☜"];
// two dimensional array 6*6
var gameBoard= new Array(6);
// 0: no box haven't been clicked
// 1: one has been clicked  2: state that a player cannot click
var clickNum= 0;
var firstClickedBox; // store information of a box first clicked
var succeededPairs= 0; // totally 18pairs
var memorizingTimeCounterArray= new Array(); // event ids for memorizing time
var playTimeCounter; // an event id for counting play time
var playedSecs= 0, playedMins= 0; // played secs, mins
var isPossibleToStartNewGame= true; // it is not possible to start new game while showing all of board for 10 secs


// clickBoard(i,j): the action when a box on the gameboard is clicked
function clickBoard(i,j) {
    var aBox= document.getElementById("board"+i+j);

    // if a box is already open is clicked
    if(aBox.innerText != "") {
        document.getElementById("message").innerText= "Choose another";
        return true;
    }

    // choose first one in a try
    if(clickNum == 0) {
        clickNum++;
        firstClickedBox= [i,j,gameBoard[i][j]];
        toggleBox(i,j);
    }
    // choose second one in a try
    else if(clickNum==1) {
        clickNum++;
        toggleBox(i,j);
        // if two are not matched
        if(gameBoard[i][j] != firstClickedBox[2])
        {
            setTimeout(function(){
                                toggleBox(i,j);
                                toggleBox(firstClickedBox[0],firstClickedBox[1]);
                                clickNum= 0;},700);
        } else {
            succeededPairs++;
            setTimeout(function(){clickNum= 0;},100);
        }
    }

    // if a player succeed to match 18 pairs
    if(succeededPairs == 18)
    {
        clearInterval(playTimeCounter);
        document.getElementById("message").innerText += " You SUCCEEDED!";
    }
}

// toggleBox(i,j): flip a box on the game board
function toggleBox(i,j) {
    var aBox= document.getElementById("board"+i+j);
    if(aBox.innerText=="")
        aBox.innerText= gameBoard[i][j];
    else
        aBox.innerText= "";
}

// toggleBoard(): flip over the game board
function toggleBoard() {
    for(var i=0;i<6;i++) {
        for(var j=0;j<6;j++) {
            toggleBox(i,j);
        }
    }
}

// resetEvents(): remove events used in last game
function resetEvents() {
    clearInterval(playTimeCounter);
    for(var i=0;i<=10;i++)
        clearTimeout(memorizingTimeCounterArray[i]);

    playedSecs= 0;
    playedMins= 0;
}

// startNewGame(): load a new round
function startNewGame() {
    if(isPossibleToStartNewGame){
        isPossibleToStartNewGame= false;
        document.getElementById("newGames").innerText=" ";
        resetEvents();
        resetBoard();
        toggleBoard();

        // count 10secs down and start game
        for(var i=0;i<=10;i++)
            memorizingTimeCounterArray[i]= count10Secs(1000*i);
        setTimeout(function(){
            toggleBoard();
            document.getElementById("message").innerText= "GO!";
        }, 11000);
i
        // count play time
        setTimeout(function(){
            clickNum= 0;
            succeededPairs= 0;
            playTimeCounter= setInterval(printCounter,1000);
            document.getElementsByClassName("active")[0].style.visibility="visible";
            document.getElementById("newGames").innerText="New GAME";
            isPossibleToStartNewGame= true;
        }, 12000);
    }
}

// count10Secs(millisecs): print secs left per every 1 secs
function count10Secs(millisecs){
    setTimeout(function(){
        document.getElementById("message").innerText= ""+(10-Number(millisecs)/1000);
    },millisecs);
}

// resetBoard(): arrange symbols on the gameboard randomly
function resetBoard() {
    for(var i=0;i<6;i++) {
        gameBoard[i] = new Array();
        for(var j=0;j<6;j++) {
            var rdmNum= Math.floor(Math.random()*(36-(i*6)-j));
            gameBoard[i][j]= groupOfSymbols[rdmNum];
            var tmp= groupOfSymbols[36-(i*6)-j-1];
            groupOfSymbols[36-(i*6)-j-1]= groupOfSymbols[rdmNum];
            groupOfSymbols[rdmNum]= tmp;
            document.getElementById("board"+i+j).innerText= "";
        }
    }
    clickNum= 2;
}

// printCounter(): count play time
function printCounter() {
    playedSecs++;
    if(playedSecs>=60)
    {
            playedSecs= 0;
            playedMins++;
    }
    document.getElementById("message").innerText=playedMins+":"+playedSecs;
}


/*
    File name: HangMan.js
    Author: Hyewon Choi
    Date created: July 31, 2018
    Version: 0.1
    Copyright: 1.0
    Purpose: Playing Hang Man game
    Description: It is a guessing game.
    A player should guess a word within 10 lives.
*/

// words is showed up in this game
var
wordsArray=['abruptly','absurd','abyss','affix','askew','avenue','awkward','axiom','azure','bagpipes','bandwagon','banjo','bayou','beekeeper','bikini','blitz','blizzard','boggle','bookworm','boxcar','boxful','buckaroo','buffalo','buffoon','buxom','buzzard','buzzing','buzzwords','caliph','cobweb','cockiness','croquet','crypt','curacao','cycle'
```

```
,'daiquiri','dirndl','disavow','dizzying','duplex','dwarves','embezzle','equip','espionage','euouae','exodus','faking','fishhook','fixable','fjord','flapjack','flopping','fluffiness','flyby','foxglove','frazzled','frizzled','fuchsia','funny','gabby','galaxy','galvanize','gazebo','giaour','gizmo','glowworm','glyph','gnarly','gnostic','gossip','grogginess','haiku','haphazard','hyphen','iatrogenic','icebox','injury','ivory','ivy','jackpot','jaundice','jawbreaker','jaywalk','jazziest','jazzy','jelly','jigsaw','jinx','jiujitsu','jockey','jogging','joking','jovial','joyful','juicy','jukebox','jumbo','kayak','kazoo','keyhole','khaki','kilobyte','kiosk','kitsch','kiwifruit','klutz','knapsack','larynx','lengths','lucky','luxury','lymph','marquis','matrix','megahertz','microwave','mnemonic','mystify','naphtha','nightclub','nowadays','numbskull','nymph','onyx','ovary','oxidize','oxygen','pajama','peekaboo','phlegm','pixel','pizazz','pneumonia','polka','pshaw','psyche','puppy','puzzling','quartz','queue','quips','quixotic','quiz','quizzes','quorum','razzmatazz','rhubarb','rhythm','rickshaw','schnapps','scratch','shiv','snazzy','sphinx','spritz','squawk','staff','strength','strengths','stretch','stronghold','stymied','subway','swivel','syndrome','thriftless','thumbscrew','topaz','transcript','transgress','transplant','triphthong','twelfth','twelfths','unknown','unworthy','unzip','uptown','vaporize','vixen','vodka','voodoo','vortex','voyeurism','walkway','waltz','wave','wavy','waxy','wellspring','wheezy','whiskey','whizzing','whomever','wimpy','witchcraft','wizard','woozy','wristwatch','wyvern','xylophone','yachtsman','yippee','yoked','youthful','yummy','zephyr','zigzag','zigzagging','zilch','zipper','zodiac','zombie'];
var rndWord; // a word is decided randomly among wordsAppay
var levelOfHangMan= 0; // how many steps are proceeded(how many times a player chose wrong an alphabet)
var succeedCnt= 0; // how many times a player chose correct an alphabet.
var canvas; // a canvas which a hangman will be drawn on
var ctx; // context of a canvas
var gameOver= false; // if a round can be continued.
var remainedLives= 10; // total number of lives in a round is 10.

// setGuessingWord(): decide a word randomly which a player should guess
function setGuessingWord() {
    rndWord= wordsArray[Math.floor(Math.random()*wordsArray.length)];
    var wordUl= document.getElementById("wordUl");
    var letters;

    while (wordUl.firstChild) {
        wordUl.removeChild(wordUl.firstChild);
    }

    for(var i=0;i<rndWord.length;i++) {
        letters= document.createElement('li');
        letters.innerText= '_';
        wordUl.appendChild(letters);
    }
}

// addEventToAlphabets(): add actions after each alphabets is clicked to them
function addEventToAlphabets() {
    var alphabetsLi= document.querySelectorAll(".alphabetsUl>li");

    for(var i=0;i<alphabetsLi.length;i++) {
        alphabetsLi[i].addEventListener("click",function(){
            if(!gameOver) {
                // clicked alphabets disappear
                this.setAttribute('id','clickedAlphabets');
                clickAlphabet(this.innerText);
            }
            // case : a player succeed to guess a word
            if(succeedCnt==rndWord.length) {
                document.getElementById("message").innerText= "YOU ARE SURVIVED!";
                gameOver= true;
            }
            // case : a player fail to guess a word within 10 turns.
            if(levelOfHangMan==10) {
                document.getElementById("message").innerText= "YOU KICKED THE BUCKET.";
                gameOver= true;
            }
        })
    }
```

```
}

// clickAlphabet(alphabet): check if the clicked alphabet is a part of guessing word
function clickAlphabet(alphabet) {
    var letters= document.querySelectorAll("#wordUl>li");
    var cnt= 0;

    for(var i=0;i<rndWord.length;i++) {
        if(rndWord.charAt(i).toUpperCase() == alphabet) {
            letters[i].innerText= alphabet;
            cnt++;
            succeedCnt++;
        }
    }
    // if the alphabet is not included in the word, draw next step of hangman
    if(cnt==0) {
        drawHangMan();
        document.getElementById("message").innerText= "You have "+(--remainedLives)+" lives.";
    }
}

// drawHangMan(): draw hangman on canvas
function drawHangMan() {
    levelOfHangMan++;
    switch(levelOfHangMan) {
        case 1:
            drawLine(15,147,285,147);
            break;
        case 2:
            drawLine(40,147,40,10);
            break;
        case 3:
            drawLine(30,15,175,15);
            break;
        case 4:
            drawLine(150,15,150,25);
            break;
        case 5:
            ctx.moveTo(150,40);
            ctx.arc(150,40,15,0,Math.PI*2,true);
            ctx.fill();
            ctx.stroke();
            break;
        case 6:
            drawLine(150,55,150,85);
            break;
        case 7:
            drawLine(150,60,110,85);
            break;
        case 8:
            drawLine(150,60,190,85);
            break;
        case 9:
            drawLine(150,85,120,125);
            break;
        case 10:
            drawLine(150,85,180,125);
            break;
    }
}

/*
    drawLine(pathFromX, pathFromY, pathToX, pathToY):
            draw line from (pathFromX, pathFromY) to (pathToX, pathToY)
*/
function drawLine(pathFromX, pathFromY, pathToX, pathToY) {
    ctx.moveTo(pathFromX, pathFromY);
    ctx.lineTo(pathToX, pathToY);
    ctx.stroke();
}
```

```
// startNewGame(): initialize variables, attributes and messages.
function startNewGame() {
    setGuessingWord();
    levelOfHangMan= 0;
  succeedCnt= 0;
    remainedLives= 10;
    gameOver= false;
    canvas= document.getElementById("hangManCanvas");
  ctx= canvas.getContext("2d");
    ctx.clearRect(0, 0, canvas.width, canvas.height);
  ctx.beginPath();
  ctx.strokeStype= "black";
  ctx.lineWidth= 2;

    var alphabetsLi= document.querySelectorAll(".alphabetsUl>li");
    for(var i=0;i<alphabetsLi.length;i++)
        alphabetsLi[i].removeAttribute('id');
    document.getElementById("message").innerText= "You have 10 lives.";
}

// onLoad(): it is run when the game html page is loaded
function onLoad() {
    startNewGame();
    addEventToAlphabets();
}
```