One Line Javascript Games
Luke Wood

Project Description

This past summer while teaching kids the basics of Java programming, I noticed a common theme; students being enrolled in the course solely to learn how to create a video game. This is often challenging for young students and in order to make this slightly more friendly and give quick results to students I started this project. This library allows users to implement fully coded and fully enclosed Javascript games in their websites using a single line of code.

Implementation

In order to reduce conflicts with other libraries or Javascript files, every game in the library is contained within a single function. The user then calls this function and the game will then be created as a child of the element the user passes as a parameter.

Code

Currently, there are only two games implemented in this library: Snake and Tic Tac Toe. Below is the Javascript for both.

Snake.js

```
function initSnake(parent, style)
{
        var snake;
        var foodArr = [];
        var count = 0;
        var currentInter;
        var repaintInter;
        var gameIsOver = true;
        window.addEventListener('keydown',function(e)
        {
    var key = e.keyCode ? e.keyCode : e.which;
        //UP
    if (key == 87)
```

```
        {
                snake.setDir(0);
    }
        //RIGHT
        if(key == 68)
        {
                snake.setDir(1);
        }
        //LEFT
        if(key == 65)
        {
                snake.setDir(3);
        }
        //DOWN
        else if (key == 83)
        {
                snake.setDir(2);
    }
        else if (key == 13)
        {
                spaceHit();
        }
        else if (key == 32)
        {
                spaceHit();
        }
        });
        var parentDiv = document.getElementById(parent);
        var snakeCanvas = document.createElement("canvas");
        snakeCanvas.setAttribute("width", "500px");
        snakeCanvas.setAttribute("height", "500px");

        var ctx = snakeCanvas.getContext("2d");
        ctx.fillStyle = "#FF0000";
        ctx.font="30px Georgia";
        ctx.fillText("SNAKE",200,200);
        ctx.fillText("WASD to control", 160,250);
        ctx.fillText("Space to Start", 160,300);
        var snakeContainer = document.createElement("div");
        if(!style)
        {
        snakeContainer.setAttribute("style","width: 520px; height: 520px;background-color:
black;padding: 10px;");
        snakeCanvas.setAttribute("style","border: solid white 2px;");
        }
        snakeContainer.appendChild(snakeCanvas);
        parentDiv.appendChild(snakeContainer);
        function tick()
        {
                if(count > 10)
                {
                foodArr.push(new food());
                count = 0;
                }
                count++;
                snake.tick();
                checkCollisions();
        }
        function startGame()
        {
                gameIsOver = false;
                snake = new snakeConstructor();
                currentInter = setInterval(tick, 200);
                repaintInter = setInterval(repaint,50);
        }
        function checkCollisions()
        {
                snake.checkColl(foodArr);
        }
        function repaint()
        {
                ctx.fillStyle = "#000000";
                ctx.fillRect(0,0,500,500);
                ctx.fillStyle="#FF0000";
                for(var i = 0; i < foodArr.length; i++)
                {
                        ctx.fillRect(foodArr[i].x *25,foodArr[i].y * 25,25,25);
                }
                snake.draw(ctx);
        }
        function food()
        {
                this.x = Math.floor(Math.random() * 20);
                this.y = Math.floor(Math.random() * 20);
        }
        function gameOver()
        {
                window.clearInterval(currentInter);
                window.clearInterval(repaintInter);
                foodArr = [];
                repaint();
                ctx.fillStyle = "#FF0000";
                ctx.font="30px Georgia";
                ctx.fillText("Game Over",170,200);
                ctx.fillText("Score: "+ snake.length, 170,250);
                ctx.fillText("Space or Enter to Restart", 120, 300);
                gameIsOver = true;
        }
        function spaceHit()
        {
                if(gameIsOver)
                {
                startGame();
                }
        }
        function snakeConstructor()
        {
                var direction = 1;
                var x = 0;
                var y = 0;
                var pastLocs = [];
                this.length = 0;
                var speed = 1;
                var hasGrown = false;
                this.checkColl = function(food)
                {
                        for(var i = 0; i < food.length; i++)
                        {
                        if(x == food[i].x && y == food[i].y)
                        {
```

```
            food.splice(i, 1);
            this.length = this.length+1;
            hasGrown = true;
            }
            }
            for(var j = 0; j < this.length;j++)
            {
                if(x == pastLocs[j].x && y == pastLocs[j].y)
                {
                    gameOver();
                }
            }
        }
        this.tick = function()
        {
            if(hasGrown)
            {
                pastLocs.push(new pastLoc(x,y));
                hasGrown = false;
            }
            for(var i = 0; i < this.length; i++)
            {
                if(i != pastLocs.length-1)
                {
                    pastLocs[i] = pastLocs[i+1];
                }
                else
                {
                    pastLocs[i] = new pastLoc(x,y);
                }
            }
            if(direction == 1)
            {
            if(x < 19)
            {
            x += speed;
            }
            else{gameOver();}
            }
            else if(direction == 2)
            {
            if(y < 19)
            {
            y += speed;
            }else{gameOver();}
            }
            else if(direction == 3)
            {
            if(x > 0)
            {
            x -= speed;
            }else{gameOver();}
            }
            else if(direction == 0)
            {
            if(y >0)
            {
            y -= speed;
            }else{gameOver();}
            }
```

```
                }
                this.setDir = function(i)
                {
                        direction = i;
                }
                this.draw = function(ctx)
                {
                        ctx.fillStyle = "#ffff00";
                        ctx.fillRect(x*25,y*25,25,25);
                        for(var i = 0; i < this.length; i++)
                        {
                                ctx.fillRect(pastLocs[i].x*25,pastLocs[i].y*25,25,25);
                        }
                }
                function pastLoc(nx,ny)
                {
                        this.x = nx;
                        this.y = ny;
                }
        }
        return snakeContainer;
}
```

TicTacToe.js

```
//NAME OF PARENT DIV
function initTicTacToe(parentContainerName, style)
{
        var currentTurn = "X";
        var xVictories = 0;
        var oVictories = 0;
        var xWins;
        var oWins;
        var lastWinnerDisplay;
        var lastWinner = "";
        var ticVic = false;
        var currentTurnDisplay = "X";
        var ticTacToeGame = document.createElement("div");
        ticTacToeGame.setAttribute("class","ticTacToeBoard");
        var leftSection = document.createElement("div");
        leftSection.setAttribute("style", "display:inline-block;float:left;");
        var rightSection = document.createElement("div");
        var scoreBoard = createScoreBoard(style);
        rightSection.appendChild(scoreBoard);
        rightSection.setAttribute("style", "display:inline-block;float:right;");
        if(!style)
        {
                ticTacToeGame.setAttribute("style","border-radius: 10px; border: solid 2px
black;display:inline-block;padding: 20px;margin: 20px; background: white;");
        }
        var parentContainer = document.getElementById(parentContainerName);
        var pieces = createTicTacToeBoard(style);
        var clearButton = document.createElement("div");
        clearButton.setAttribute("id","ticTacToeClearButton");
        if(!style)
        {
                clearButton.setAttribute("style","margin-top: 10px;width: 200px;
    height: 30px; font-family: Rockwell; font-size: 20px; background-color: #D0D0D0; cursor:
 pointer;");
        }
        clearButton.addEventListener("click", clearBoard);
        clearButton.innerHTML="Clear";

        for(var i = 0; i < 9; i++)
        {
        (function(t){
        pieces[t].addEventListener("click",function(){takeTurn(t);});
        })(i)
            if(i%3 == 0)
            {
                leftSection.appendChild(document.createElement("br"));
            }
            leftSection.appendChild(pieces[i]);
        }
        leftSection.appendChild(document.createElement("br"));
        leftSection.appendChild(clearButton);
        ticTacToeGame.appendChild(leftSection);
        ticTacToeGame.appendChild(rightSection);
        parentContainer.appendChild(ticTacToeGame);

        function createTicTacToeBoard(style)
        {
        var arr = [];
        for(var i = 0; i < 9; i++)
        {

            var temp = document.createElement("div");
               temp.setAttribute("class", "ticTacToeBoardPiece");
               if(!style)
               {
                temp.setAttribute("style","width: 50px; height: 50px;
 background-color: white; border: solid 2px black;border-radius: 4px; margin: 5px; display:
 inline-block;  vertical-align:top;");
               }
               arr.push(temp);
        }
        return arr;
        }
        function takeTurn(i)
        {
               if(!ticVic)
               {
               if(pieces[i].innerHTML ==="")
               {
               pieces[i].innerHTML = currentTurn;
               if(currentTurn === "X"){currentTurn = "O"}
               else{currentTurn = "X"}
               checkVictory();
               }
               }
        }
        function clearBoard()
        {
               ticVic = false;
               for(var i = 0; i < 9; i++)
               {
```

```
                        pieces[i].innerHTML = "";
                }
        }

        function createScoreBoard(a)
        {
                var scoreBoardDiv = document.createElement("div");
                scoreBoardDiv.setAttribute("id", "ticTactoeScoreboard");
                xWins = document.createElement("div");
                oWins = document.createElement("div");
                xWins.setAttribute("class", "ticTacToeWinsDisplay");
                oWins.setAttribute("class", "ticTacToeWinsDisplay");

                currentTurnDisplay = document.createElement("div");
                currentTurnDisplay.setAttribute("id", "ticTacToeCurrentTurnDisplay");
                var header = document.createElement("div");
                lastWinnerDisplay = document.createElement("div");
                lastWinnerDisplay.setAttribute("id","ticTacToeLastWinnerDisplay");
                lastWinnerDisplay.innerHTML = "Last Winner: ";
                header.setAttribute("id","ticTacToeScoreboardHeader");
                header.innerHTML = "Scoreboard";
                if(!a)
                {
                        header.setAttribute("style","font-family: Rockwell; font-size:
20px;font-weight: bold;");
                        scoreBoardDiv.setAttribute("style","display:inline-block; padding:
10px; margin:10px; border: solid black 2px; border-radius: 5px;");
                        xWins.setAttribute("style","font-family: Rockwell; font-size:
15px;font-weight: bold;");
                        oWins.setAttribute("style","font-family: Rockwell; font-size:
15px;font-weight: bold;");
                        currentTurnDisplay.setAttribute("style","font-family:Courier,
monospace; font-size: 15px;font-weight: bolder;");
                        lastWinnerDisplay.setAttribute("style","font-family:Courier,
monospace; font-size: 15px;font-weight: bolder;");
                }
                updateWins();
                scoreBoardDiv.appendChild(header);
                scoreBoardDiv.appendChild(xWins);
                scoreBoardDiv.appendChild(oWins);
                scoreBoardDiv.appendChild(document.createElement("br"));
                scoreBoardDiv.appendChild(currentTurnDisplay);
                scoreBoardDiv.appendChild(document.createElement("br"));
                scoreBoardDiv.appendChild(lastWinnerDisplay);

                return scoreBoardDiv;
        }
        function updateWins()
        {
                var xText = "X Wins: "+ xVictories;
                var yText = "Y Wins: "+ oVictories;
                xWins.innerHTML = xText;
                oWins.innerHTML = yText;
                currentTurnDisplay.innerHTML = "Current Turn: "+ currentTurn;
        }
        function checkVictory()
        {
        if(!ticVic)
        {
        if(pieces[0].innerHTML === pieces[4].innerHTML && pieces[0].innerHTML ===
pieces[8].innerHTML&& !(pieces[0].innerHTML === ""))
        {
        victory(pieces[0].innerHTML);
        }
        if(pieces[0].innerHTML === pieces[1].innerHTML && pieces[0].innerHTML ===
pieces[2].innerHTML&& !(pieces[0].innerHTML === ""))
        {
        victory(pieces[0].innerHTML);
        }
        if(pieces[0].innerHTML === pieces[3].innerHTML && pieces[0].innerHTML ===
pieces[6].innerHTML&& !(pieces[0].innerHTML === ""))
        {
        victory(pieces[1].innerHTML);
        }
        if(pieces[1].innerHTML === pieces[4].innerHTML && pieces[1].innerHTML ===
pieces[7].innerHTML && !(pieces[1].innerHTML === ""))
        {
        victory(pieces[1].innerHTML);
        }
        if(pieces[2].innerHTML === pieces[4].innerHTML && pieces[2].innerHTML ===
pieces[6].innerHTML&& !(pieces[2].innerHTML === ""))
        {
        victory(pieces[2].innerHTML);
        }
        if(pieces[2].innerHTML === pieces[5].innerHTML && pieces[2].innerHTML ===
pieces[8].innerHTML&& !(pieces[2].innerHTML === ""))
        {
        victory(pieces[2].innerHTML);
        }
        if(pieces[3].innerHTML === pieces[4].innerHTML && pieces[3].innerHTML ===
pieces[5].innerHTML&& !(pieces[3].innerHTML === ""))
        {
        victory(pieces[3].innerHTML);
        }
        if(pieces[6].innerHTML === pieces[7].innerHTML && pieces[6].innerHTML ===
pieces[8].innerHTML&& !(pieces[6].innerHTML === ""))
        {
        victory(pieces[6].innerHTML);
        }
                }
        updateWins();
        }
function victory(a)
{
        if(!ticVic)
        {
        ticVic = true;
        if(a === "X"){xVictories++; lastWinner = "X";}
        else if(a ==="O"){oVictories++;lastWinner = "O";}
        }
        lastWinnerDisplay.innerHTML = "Last Winner: "+ lastWinner;
}
        return ticTacToeGame;
}
```

Concluding Remarks

This project really makes great use of the scoping mechanics in Javascript and allows for minimal conflict between Onelinejsgames and other libraries.