**MINI PROJECT REPORT**

ON

**Compilation Of 3 Games**

MINI PROJECT LAB (KCS-354)

Submitted by-

Manan Bansal

(2000970130061)

Hardik Srivastava

(2000970130051)

Under the supervision of

Mr. Nilotpal Pathak



Galgotias College of Engineering and Technology

Greater Noida

**Index**

|  |  |  |
| --- | --- | --- |
| **S**  **No.** | **Content** | **Page No.** |
| **1** | **Acknowledgement** | **3** |
| **2** | **Introduction** | **4** |
| **3** | **Literature Review** | **5** |
| **4** | **Description** | **6** |
| **5** | **Methodology of the Research Work** | **7** |
| **6** | **Code and Results** | **8-24** |
| **7** | **List of References** | **25** |

**Acknowledgement**

We want to extend special thanks to our mini project coordinator Mr. Nilotpal Pathak for the timely advice and the valuable guidance during the designing and implementation of this project work.

We also want to express our sincere thanks and gratitude to Dr. SK Singh head of the department of Information Technology branch for providing us with the facilities and for all the encouragement and support.

Finally, We would like to express our sincere thanks towards all the staff members in the department of Information Technology for their support and coordination.

Manan Bansal

(2000970130061)

Hardik Srivastava

(2000970130051)

**Introduction**

In todays era full of stress and work , people have forgot their good old games and this is the time to bring those memories back with this game zone. This project is designed in such a way so that people who faces the problem of stress and work can get out of this Real World for sometime and emerge in the world of Games. So The objective of this project is to make various games using interesting web technologies such as Hyper Text Markup Language (HTML) , cascading stylesheet (CSS) and java script (JS) . The project is based on real world implementation of these technologies . The games will be executed in the form of a Game Zone .The interesting and challenging games include :

• Snake Game

• Colour Chosing Game

• Tic Tac Toe

**Literature Review**

The Early Years The First recognized example of a game machine was unveiled by Dr.Edward uhler Condon at the New York World’s fair in 1940 . The Game ,based On the ancient mathematical game of Nim ,was played by about 50,000 people, with the computer winning more than 90 percent of the games. There are various Games made since the evolution of tech and arcade as well . In the earlier times , to play the Games kids used to go to arcade parlor far away from their home and also costs them a lot of money but with the expansion in the technology and rapid advancement in modern tools have bring all this to a whole new complete level. These new technologies have brought up some really great games of the decades such as GTA series , FAR CRY series and many more , but in this hype of new games we have lost our GOOD OLD GAMES like snake game , Tic Tac Toe etc.

**Description**

The best way to learn any programming language is through hands-on projects.The Snake Game ,Colour Chosing Game and Tic Tac Toe is a simple game you can make using the basics of HyperText Markup Language (HTML) , Cascading Style Sheets (CSS) and JavaScript (JS) These Games are a classic video game from the late 70s.The basic goal is to navigate a snake and eat as many apples as possible without touching the walls or the snake’s body. The colour game is supposed to be a interesting game in which the correct colour shown is to be chosen and points are given on the basis of answer being right or wrong. The goal of TIC TAC TOE is to make 3 in a line together ,in this game both players places their O or X turn by turn trying to make 3 same together either rowise or columnwise or diagonally. Today ,we’ll show you step-by -step how to create these type of amazing games using HyperText Markup Language (HTML) , Cascading Style Sheets (CSS) and JavaScript (JS).To succeed in this tutorial ,you should have a basic understanding of HTML,CSS and JS.

**Methodology of research work and Major tools Required**

Currently, a full appreciation of how problem gaming impacts the daily lives of gamers is lacking. This study aims to gain a more holistic understanding of the activities in the daily lives of problem gamers This is a remake of the Snake video game which was first released during the mid 1970s in arcades. Whenever we think of iconic games that we’ve played as a child, most of us would remember games like NFS II SE, or Roadrash or SEGA’s Virtua Cop 2. However, people who didn’t have access to PC games were fixated on the games that were available on their parent’s phones.And one game that was loved by many was the classic Snake game, often seen on the black and white Nokia phones like the Nokia 3310, 3315 and several other models that followed.

The best way to learn any programming language is through hands-on projects.The Snake Game ,Colour Chosing Game and Tic Tac Toe is a simple game you can make using the basics of HyperText Markup Language (HTML) , Cascading Style Sheets (CSS) and JavaScript (JS) These Games are a classic video game from the late 70s.The basic goal is to navigate a snake and eat as many apples as possible without touching the walls or the snake’s body. The colour game is supposed to be a interesting game in which the correct colour shown is to be chosen and points are given on the basis of answer being right or wrong. The goal of TIC TAC TOE is to make 3 in a line together ,in this game both players places their O or X turn by turn trying to make 3 same together either rowise or columnwise or diagonally. Today ,we’ll show you step-by -step how to create these type of amazing games using HyperText Markup Language (HTML) , Cascading Style Sheets (CSS) and JavaScript (JS).To succeed in this tutorial ,you should have a basic understanding of HTML,CSS and JS.

**Code**

**Snake Game:-**

**Html**

<html>

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Document</title>

<link rel="stylesheet" href="style.css">

</head>

<body>

<div class="body">

<div id="board">

</div>

<div id="score">

score: 0

</div>

<div id="Hiscore">

Hiscore: 0

</div>

</div>

<script src="java.js"></script>

</body></html>

**Css**

@import url('https://fonts.googleapis.com/css2?family=Klee+One:wght@600&display=swap');

\*{

margin:0;

padding:0;

}

.body{

background:url("https://th.bing.com/th/id/R.fca4f1b0a1f7a1d1e05ce9d62a9b44af?rik=91C1IDmMX%2bYaJw&riu=http%3a%2f%2fwallpoper.com%2fimages%2f00%2f33%2f99%2f90%2fouter-space\_00339990.jpg&ehk=YNkGNqUjAeU7ULYRKKuxQbfXyUaCbEMMJYIOZJD1O8o%3d&risl=&pid=ImgRaw&r=0");

min-width:100vw;

min-height:100vh;

background-size:100vw 100vh;

background-repeat:no-repeat;

display:flex;

justify-content:center;

align-items:center;

}

#board

{

background:black;

width: 70vmin;

height: 70vmin;

border: 2px solid white;

display:grid;

grid-template-rows:repeat(18,1fr);

grid-template-columns:repeat(18,1fr);

}

.snake{

background-color:lightgreen;

}

.head{

background-color:green;

}

.food{

background-color:chocolate;

border-radius:3px;

}

#score{

position:absolute;

top: 20px;

color:white;

right:100px;

font-size: 30px;

font-family: 'Klee One', cursive;

}

#hiscore{

position:absolute;

top: 50px;

color:white;

right:100px;

font-size: 30px;

font-family: 'Klee One', cursive;

}

**JS**

let indir={x:0,y:0};

let speed=10;

let lastPaintTime=0;

let snake=[{x: 13,y: 15}];

let food={x:6,y:7};

let scorejs=0;

let hiscore=0;

function main(ctime){

window.requestAnimationFrame(main);

if((ctime-lastPaintTime)/1000<(1/speed)){

return;

}

lastPaintTime=ctime;

gameengine();

}

function gameengine(){

if(iscollide(snake)){

indir={x:0,y:0};

scorejs=0;

score.innerHTML="Score : 0";

alert("Game Over");

snake=[{x: 13,y: 15}];

}

if(snake[0].x==food.x && snake[0].y==food.y)

{

let a=2;

let b=16;

snake.unshift({x:snake[0].x+indir.x,y:snake[0].y+indir.y});

food={x:Math.round(a+(b-a)\*Math.random()),y:Math.round(a+(b-a)\*Math.random())};

scorejs+=10;

if(scorejs>hiscore)

{

hiscore=scorejs;

Hiscore.innerHTML="Hi-Score : "+ hiscore;

}

score.innerHTML="Score : "+ scorejs;

}

for(let i=snake.length-2;i>=0;i--)

{

snake[i+1]={...snake[i]};

}

snake[0].x+=indir.x;

snake[0].y+=indir.y;

board.innerHTML="";

snake.forEach((e,index)=>{

snakeElement=document.createElement('div');

snakeElement.style.gridRowStart=e.y;

snakeElement.style.gridColumnStart=e.x;

if(index==0)

snakeElement.classList.add('head');

else

snakeElement.classList.add('snake');

board.appendChild(snakeElement);

});

foodElement=document.createElement('div');

foodElement.style.gridRowStart=food.y;

foodElement.style.gridColumnStart=food.x;

foodElement.classList.add('food');

board.appendChild(foodElement);

}

function iscollide(snake){

for(let index=1;index<snake.length;index++)

{

if(snake[index].x===snake[0].x && snake[index].y===snake[0].y)

return true;

}

if(snake[0].x>=18 || snake[0].y>=18 || snake[0].y<=0 ||snake[0].x<=0)

return true;

}

window.requestAnimationFrame(main);

window.addEventListener('keydown',e =>{

indir={x:0,y:1};

switch(e.key){

case "ArrowUp" :

indir.x=0;

indir.y=-1;

break;

case "ArrowDown" :

indir.x=0;

indir.y=1;

break;

case "ArrowLeft" :

indir.x=-1;

indir.y=0;

break;

case "ArrowRight" :

indir.x=1;

indir.y=0;

break;

default:

break;

}

});

**Color Choosing Game:-**

**Html**

<html><head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Colour Game</title>

<link rel="stylesheet" href="style.css">

</head>

<body>

<div class="scoreboard">

<div class="board">

<h4> Timer </h4>

<span id="timer">5</span>

</div>

<div class="board" id="scorecard">

<h4> Score</h4>

<span id="score">0</span>

</div>

<div class="board">

<h4>Round</h4>

<span id="round">0</span>/20

</div>

</div>

<div class="banner"></div>

<div class="choices">

<div class="row">

<div class="circle" id="opt1"></div>

<div class="circle" id="opt2"></div>

<div class="circle" id="opt3"></div>

</div>

<div class="row">

<div class="circle" id="opt4"></div>

<div class="circle" id="opt5"></div>

<div class="circle" id="opt6"></div>

</div>

</div>

<script src="index.js"></script>

</body></html>

**Css**

body {

margin: 5px;

font-family: Verdana, sans-serif;

}

.scoreboard{

display:flex;

flex-direction:row;

margin-top:10px;

justify-content:center;

}

.board {

background-color: lightgray;

font-size: 15px;

height: 100px;

width: 300px;

margin-left: 10px;

margin-bottom:20px;

text-align: center;

justify-content:center;

border-radius: 10px;

}

span{

color:red;

}

.banner {

height: 200px;

width: 100%;

margin: 0;

padding: 0;

background-color: rgb(255,255,255);

}

.choices {

margin-top: 30px;

}

.row {

display: flex;

flex-direction: row;

justify-content: center;

}

.circle {

height: 100px;

width: 100px;

border-radius: 50%;

margin: 15px;

border: 5px solid black;

background-color: rgb(255,255,255);

}

**Js**

function gamecolour(){

let optionsArr=document.getElementsByClassName('circle');

let colourArr=[];

for(let i=0;i<optionsArr.length;i++)

{

r=Math.floor(Math.random()\*256);

g=Math.floor(Math.random()\*256);

b=Math.floor(Math.random()\*256);

optionsArr[i].style.backgroundColor="rgb(" + r+","+g+","+b + ")";

colourArr[i]="rgb(" + r+","+g+","+b + ")";

correct = Math.floor(Math.random() \* 5);

document.getElementsByClassName('banner')[0].style.backgroundColor = colourArr[correct];

}

}

function timer(){

let timeleft = Math.floor(document.getElementById('timer').textContent);

timeleft -= 1;

document.getElementById('timer').textContent = timeleft;

let round = Math.floor(document.getElementById('round').textContent);

if (timeleft == 0) {

gamecolour();

document.getElementById('timer').textContent = 5;

round += 1;

}

document.getElementById('round').textContent = round;

}

let options = document.getElementsByClassName('circle');

for(let i=0;i<options.length;i++)

{

options[i].addEventListener('click',(e) =>{

target=e.target;

correct = document.getElementsByClassName('banner')[0].style.backgroundColor;

choosed = target.style.backgroundColor;

let round = Math.floor(document.getElementById('round').textContent);

let score = Math.floor(document.getElementById('score').textContent);

if (correct == choosed) {

gamecolour();

document.getElementById('timer').textContent = 5;

round += 1;

score += 30;

}

else {

score -= 10;

document.getElementById('scorecard').style.backgroundColor = 'rgb(255, 92, 51)';

setTimeout(() =>{

document.getElementById('scorecard').style.backgroundColor = 'lightgray';

}, 500);

}

document.getElementById('round').textContent = round;

document.getElementById('score').textContent = score;

if(round==21)

{

round=0;

document.getElementById('score').textContent=0;

document.getElementById('round').textContent=0;

alert("Time's Up");

}

});

}

let game = setInterval(timer, 1000);

**Tic Tac Toe:-**

#include<iostream>

#include<stdlib.h>

#include<conio.h>

#include<time.h>

using namespace std;

int count=0,win=0,lose=0;

char a[9]=" ";

int pinput()

{

int place;

cout<<"\n enter the place";

cin>>place;

a[place-1]='X';

}

int cplace,cvalue;

int randomising()

{

srand(time(0));

cplace=(rand()%8);

cvalue=(rand()%2);

count=0;

}

void check()

{

if(a[cplace]=='X' || a[cplace]=='O')

randomising();

else

{

a[cplace]='O';

count=1;

}

}

void endgame()

{

if(a[0]==a[1] && a[1]==a[2] && a[0]=='X')

win=1;

else if(a[0]==a[3] && a[3]==a[6] && a[0]=='X')

win=1;

else if(a[0]==a[4] && a[4]==a[8] && a[0]=='X')

win=1;

else if(a[1]==a[4] && a[4]==a[7] && a[1]=='X')

win=1;

else if(a[2]==a[5] && a[5]==a[8] && a[2]=='X')

win=1;

else if(a[2]==a[4] && a[4]==a[6] && a[2]=='X')

win=1;

else if(a[3]==a[4] && a[4]==a[5] && a[3]=='X')

win=1;

else if(a[6]==a[7] && a[7]==a[8] && a[6]=='X')

win=1;

if(a[0]==a[1] && a[1]==a[2] && a[0]=='O')

lose=1;

else if(a[0]==a[3] && a[3]==a[6] && a[0]=='O')

lose=1;

else if(a[0]==a[4] && a[4]==a[8] && a[0]=='O')

lose=1;

else if(a[1]==a[4] && a[4]==a[7] && a[1]=='O')

lose=1;

else if(a[2]==a[5] && a[5]==a[8] && a[2]=='O')

lose=1;

else if(a[2]==a[4] && a[4]==a[6] && a[2]=='O')

lose=1;

else if(a[3]==a[4] && a[4]==a[5] && a[3]=='O')

lose=1;

else if(a[6]==a[7] && a[7]==a[8] && a[6]=='O')

lose=1;

}

int main()

{

int turn=0;

cout<<" TIC TAC TOE\n\n\n";

for(int i=0;i<=4;i++)

{

randomising();

do

{

check();

}while(count==0);

cout<<"\t\t\t | |\n\t\t\t "<<a[0]<<" | "<<a[1]<<" | "<<a[2]<<"\n\t\t\t | |\n \t\t\t\_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_\n";

cout<<"\t\t\t | |\n\t\t\t "<<a[3]<<" | "<<a[4]<<" | "<<a[5]<<"\n\t\t\t | |\n \t\t\t\_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_\n";

cout<<"\t\t\t | |\n\t\t\t "<<a[6]<<" | "<<a[7]<<" | "<<a[8]<<"\n\t\t\t | |\n\t\t\t | |\n\n";

endgame();

if(lose==1)

{

cout<<"\n\n you lost comp won";

break;

}

else if(win==1)

{

cout<<"\n\n you won comp lost";

break;

}

turn++;

if(turn==9 && win==0 && lose==0)

{

cout<<"\n\n IT IS A TIE";

break;

}

pinput();

cout<<"\t\t\t | |\n\t\t\t "<<a[0]<<" | "<<a[1]<<" | "<<a[2]<<"\n\t\t\t | |\n \t\t\t\_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_\n";

cout<<"\t\t\t | |\n\t\t\t "<<a[3]<<" | "<<a[4]<<" | "<<a[5]<<"\n\t\t\t | |\n \t\t\t\_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_\n";

cout<<"\t\t\t | |\n\t\t\t "<<a[6]<<" | "<<a[7]<<" | "<<a[8]<<"\n\t\t\t | |\n\t\t\t | |\n\n";

endgame();

if(lose==1)

{

cout<<"\n\n you lost comp won";

break;

}

else if(win==1)

{

cout<<"\n\n you won comp lost";

break;

}

turn++;

if(turn==9 && win==0 && lose==0)

{

cout<<"\n\n IT IS A TIE";

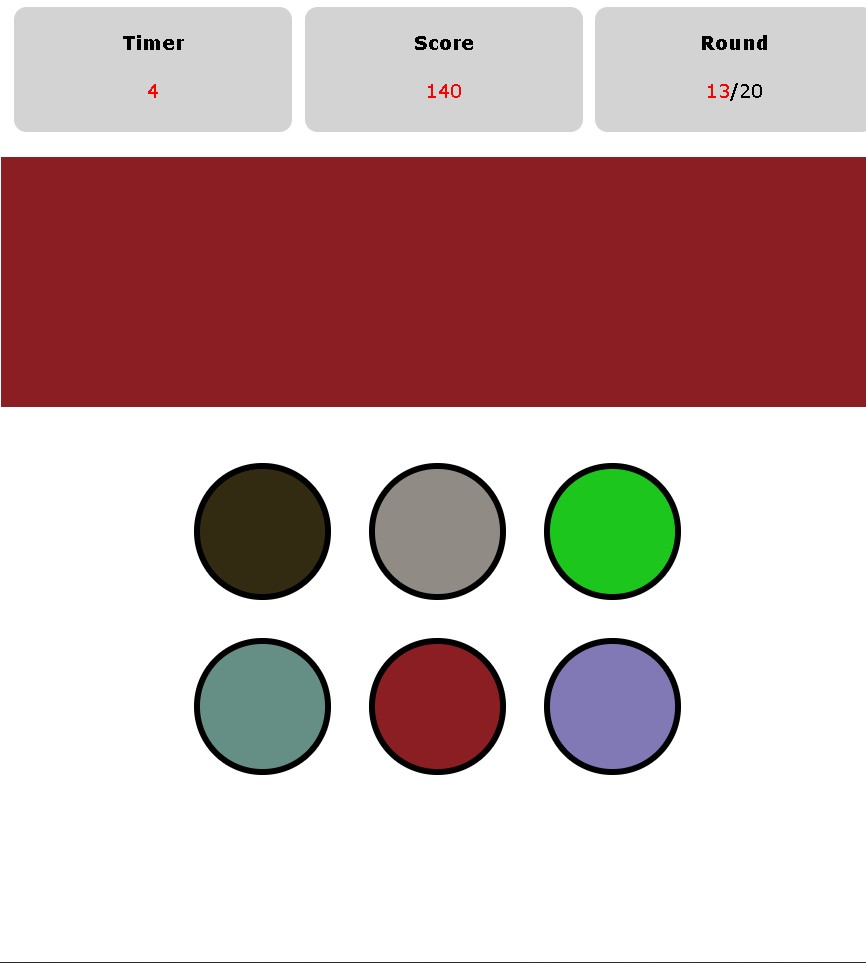
break;

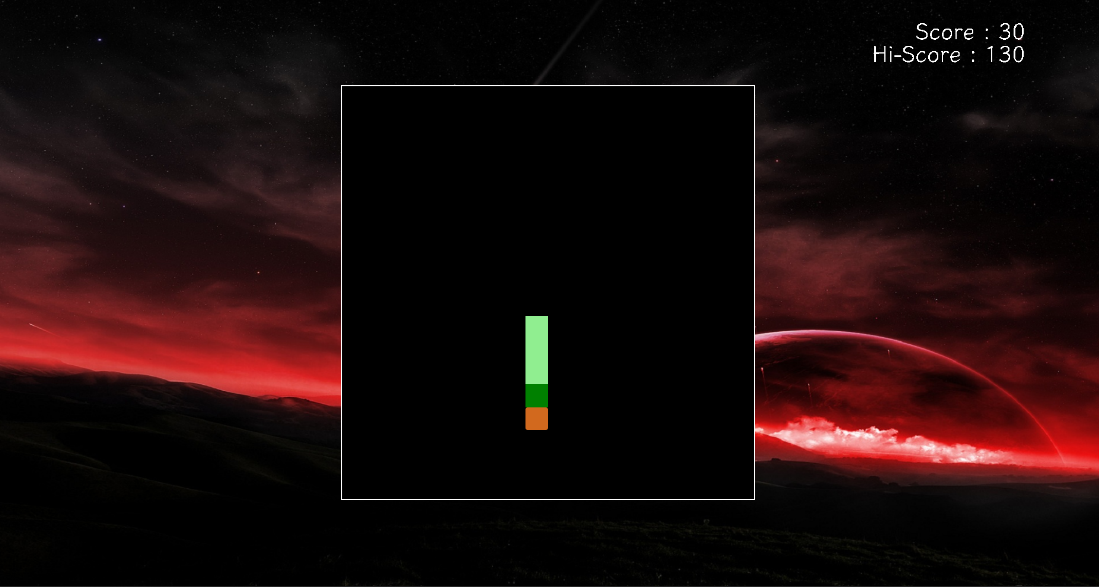
}

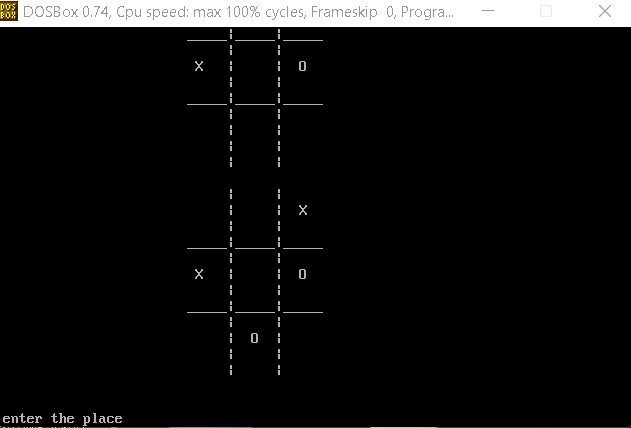
}

getch();

}







**List of References**

* W3schools Online Web Tutorials <https://www.w3schools.com/>
* The use of games and play to achieve the real world goals - <http://gamification-research.org/2015/06/the-use-of-games-and-play-to-achieve-real-world-goals/>
* Worms Zone .io - Hungry Snake – Apps on Google Play https://play.google.com › store › apps › details
* A Game Development Requirement to make 2-D Games - <https://www.researchgate.net/publication/347835330_A_game_development_environment_to_make_2D_games>
* D Math Primer for Graphics and Game Development • Author Name :- Fletcher Dunn