Car Collision Game

REPORT

SUBMITTED IN PARTIAL FULFILMENT OF THE
REQUIREMENT FOR
SUMMER INDUSTRIAL TRAINING
PRACTICAL TRAINING - 1(PT-CSE-329G)

SUBMITTED BY

NAME: **Ashish Goyal**
COLLEGE ROLL: **206010**
REGN NO: **2012191075**

Department of Computer Science Engineering

St. Andrews Institute of Technology & Management

GURGAON, INDIA

TO WHOM IT MAY CONCERN

I hereby certify that **Ashish Goyal** Roll No. **206010** of St Andrew's Institute of Technology and management Gurugram. He worked on **CAR COLLISION GAME** project during the training under the supervision of **MRS RITIKA**. During his tenure with us I found him sincere and hardworking.

Wishing him great success in the future.

Signature of the Student

Signature of the HOD **Signature of the Mentor (S)**

ACKNOWLEDGEMENT

I am highly grateful to the **Dr. Jugnesh Kumar**, Director, St. Andrews's Institute of Technology and Management, Gurugram for providing this opportunity to carry out the summer Internship training.

The constant guidance and encouragement received from Dean, SAITM, Gurugram, has been of great help in carrying out the project work and is acknowledged with reverential thanks.

I consider myself lucky enough to get such a good project. This project would add as an asset to my academic profile.

I would like to express my thankfulness to my project guide, **Mrs. RITIKA** for her constant motivation and valuable help through the project work.
Finally, I would like to thank my friends for their co-operation to complete this project.

ASHISH GOYAL
ROLL NUMBER: 206010
ST. ANDREWS INSTITUTE OF TECHNOLOGY & MANAGEMENT

ST. ANDREWS INSTITUTE OF TECHNOLOGY &MANAGEMENT GURGAON AUG-DEC, 2022

BONAFIDE CERTIFICATE

This is to certify that the project entitled **CAR COLLISION GAME** is the bonafide record of project work done by **Mr. ASHISH GOYAL**, CRN: **206010** of B. Tech during the year 2022

Signature of the DIRECTOR **Signature of the HOD**

DECLARATION

I affirm that the project work titled **CAR COLLISION GAME** being submitted in partial fulfillmentfor the award of the degree of B. TECH is the original work carried out by me. It has not formed the part of any other project work submitted for award of any degree or diploma, either in this or any other university.

Signature of the candidate
ASHISH
CRN: 206010
BRANCH: B. TECH CSE 5TH SEM

I certify that the declaration made above by the candidate is true.

Signature of the Guide
MRS.RITIKA
ASST. PROFF
SAITM

TABLE CONTENT

Chapter No.	Title	Page no.
1	Introduction	7
2	Technology Overview	8
3	Concepts	14
4	Flow Chart	17
5	Coding	19
6	Testing	29
7	Output Screen	30
8	Conclusion & Future Scope	33
9	Bibliography	34

CHAPTER -1

INTRODUCTION

Introduction:

I have built a car collision game which uses JavaScript concepts for its working and HTML, CSS are used for its structure and designing.

I have created this game with a clean user interface that can be used by users easily. I will start by creating the HTML layout first that defines the structure of the game, make it look good by styling using CSS and then write the game logic for all the functions in JavaScript.

I have used Play/pause button for better interaction. I am also calculating the live scores for the session. It is a typical car collision game which I are playing from our old times in the keypad phone.

In order to play this game, you must click on the start button then play button. You can play this game with the arrow keys and touch pad for moving left, right. When your car collides with another car coming from opposite direction, you will lose. And the score and restart button displayed on the screen.

CHAPTER – 2

TECHNOLOGY OVERVIEW

This application consists the following technologies:

- HTML
- CSS
- JavaScript

1. HTML

HTML is the **language in which most Ibsites are written**. HTML is used to create pages and make them functional.

- HTML stands for Hyper Text Markup Language
- HTML is the standard markup language for creating Ib pagesHTML
- describes the structure of a Ib page
- HTML consists of a series of elements
- HTML elements tell the browser how to display the content

Syntax:

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>
<h1>My First Heading</h1>
<p>My first paragraph </p>
</body>
</html>
```

Syntax Explained:

- The <!DOCTYPE html> declaration defines that this document is an HTML5document
- The <html> element is the root element of an HTML page

- The <head>element contains meta information about the HTML page
- The <title> element specifies a title for the HTML page (which is shown in the browser's title bar or in the page's tab)
- The <body> element defines the document's body, and is a container for all the visible contents, such as headings, paragraphs, images, hyperlinks, tables, lists, etc.
- The <h1> element defines a large heading
- The <p> element defines a paragraph

The History of HTML

HTML was first created by Tim Berners-Lee, Robert Cailliau, and others starting in **1989**.
.

In HTML, Hypertext means that the document contains **links that allow the reader to jump to other places** in the document or to another document altogether. The latest version which I use is known as HTML5.

A **Markup Language** is a way that computers speak to each other to control how text is processed and presented. To do this HTML uses two things: tags and **attributes**.

Tags and Attributes

Tags and attributes are the basis of HTML.

They work together but perform different functions. They are explained below: -

HTML Tags

Tags are used to **mark up the start of an HTML element** and they are usually enclosed in angle brackets. An example of a tag is: `<h1>`.

Most tags must be opened `<h1>` and closed `</h1>` in order to function.

There are some tags which does not have a closing tag E.g.-
, <hr> tag etc.

HTML Attributes

Attributes contain **additional pieces of information**. Attributes take the form of an opening tag and additional info is **placed inside**.

An example of an attribute is:

```
<img src="myhouse.jpg" alt="A photo of my house.">
```

In this instance, the image source (src) and the alt text (alt) are attributes of the <img> tag.

HTML TAGS USED IN OUR PROJECT: -

- I used <link rel="stylesheet" href="css/style.css"> tag for adding CSS in our html file.

- I used <script src="js/script.js"></script> tag at the end of the body tag for adding JavaScript.

- I have also used tag many times in our html code.

2. CSS

- CSS stands for Cascading Style Sheets
- CSS describes how HTML elements are to be displayed on screen, paper, or in other media.
- CSS saves a lot of work. It can control the layout of multiple Ib pages all at once
- External stylesheets are stored in CSS files

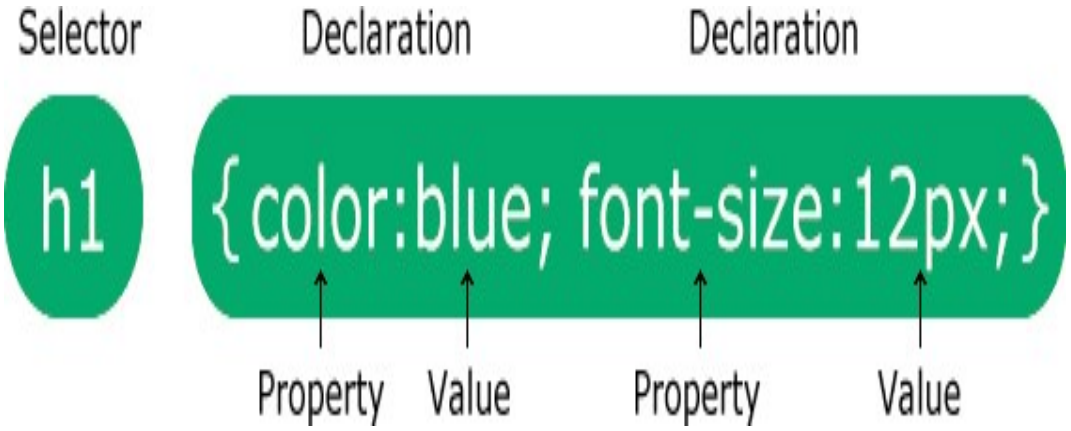
CSS Example

```
body {
  background-color: #ff359b;
}

h1 {
  color: white;
  text-align: center;
}

p {
  font: verdana;
  font-size:10px;
}
```

CSS Syntax



- The selector points to the HTML element you want to style.

- The declaration block contains one or more declarations separated by semicolons. Each declaration includes a CSS property name and a value, separated by a colon.

- Multiple CSS declarations are separated with semicolons, and declaration blocks are surrounded by curly braces.

CSS USED IN OUR PROJECT: -

- Here I used '**background: url("../img/bg.jpg");**' property for adding image in the background.
- I used '**background: linear-gradient (rgb(170, 236, 170), rgb(236, 236, 167));**' for adding color gradient.
- Also I used **Google font** for different type of font styling.
- Also I used **Grid Properties** for moving the snake in different positions and also used **Flex Properties** for designing.

3.JAVASCRIPT

- JavaScript is commonly used for **creating Ib pages**.
- It allows us to add dynamic behavior to the Ibpage and add special effects to the Ib page.
- On Ibsites, it is mainly used for validation purposes.
- JavaScript helps us to execute complex actions and enables the interactionof Ibsites with visitors.

JavaScript often abbreviated **JS**, is a programming language that is one of the core technologies of the World Wide Ib, alongside HTML and CSS. Most of the Ibsites use JavaScript on the client side for Ib page . All major Ib browsers have adedicated JavaScript engine to execute the code on the user's device. Eg-V8 engine.

JavaScript is a high-level, often just-in-time compiled language that conforms to the ECMAScript standard. It has dynamic typing, prototype-based object-orientation, and first-class functions. It is multi-paradigm, supporting event-driven, functional, and imperative programming styles.

It has application programming interfaces (APIs) for working with text, dates, regularexpressions, standard data structures, and the Document Object Model (DOM).

JavaScript engines Ire originally used only in Ib browsers, but are now core components of some servers and a variety of applications. The most popular runtimesystem for this usage is Node.js.

Importance of JavaScript

- JavaScript, helps us in developing front-end as Ill as back-end software using different JavaScript based frameworks like jQuery, Node.JS etc.
- JavaScript usage has now extended to mobile app development, desktop app development, and game development. This opens many opportunities for us as JavaScript Programmer.
- Due to high demand, there is lots of job growth and high pay for those who know JavaScript.
- Great thing about JavaScript is that I have lots of frameworks and Libraries already developed which can be used directly in our software development to reduce our time

SYNTAX:

```
<html>
  <body>
    <script language = "javascript" type = "text/javascript">
      <!--
        document.write("Hello World!")
      //-->
    </script>
  </body>
</html>
```

There are many useful **JavaScript frameworks** and libraries available:

- Angular
- React
- jQuery
- Vue.js
- Ext.js
- Node.js

It is impossible to give a complete list of all the available JavaScript frameworks and libraries. The JavaScript world is just too large and too much new is happening.

Applications of JavaScript Programming:

As mentioned before, **JavaScript** is one of the most widely used **programming languages** (Front-end as Ill as Back-end). It has its presence in almost every area of software development. I am going to list few of them here:

- **Client-side validation** - This is important to verify any user input before submitting it to the server and JavaScript plays an important role in validating thoseinputs at front-end itself.
- **Manipulating HTML Pages** - JavaScript helps in manipulating HTML page on the

fly. This helps in adding and deleting any HTML tag very easily using JavaScript and modify your HTML to change its look and feel based on different devices and requirements.

- **User Notifications** - You can use JavaScript to raise dynamic pop-ups on the Ibpages to give different types of notifications to your Ibsite visitors.

- **Back-end Data Loading** - JavaScript provides Ajax library which helps in loading back-end data while you are doing some other processing. This really gives an amazing experience to your Ibsite visitors.

- **Presentations** - JavaScript also provides the facility of creating presentations which gives Ibsite look and feel. JavaScript provides Reveals and Bespoke libraries to build a Ib-based slide presentation.

- **Server Applications** - Node JS is built on Chrome's JavaScript runtime for building fast and scalable network applications. This is an event-based library which helps in developing very sophisticated server applications including Ib Servers.

JAVASCRIPT USED IN OUR PROJECT: -

In my project I have used different logics for running the car which were written in JavaScript.

- I have used different datatypes like – ***let, var, const.***
- I have used **DOM** (Document Object Model) properties for fetching the data.
- I have created many **functions** like setupImages, setupEnemy , moveLeft, moveRight , stopGame & toggleStop.
- I have used Objects for the left and right direction of the car left--**vector(10,0) ,right--vector(-10,0).**
- I have used Event Listeners for adding different functionalities like – '**keydown**',etc.
- I have used **this** keyword and alert method for displaying the message .

CHAPTER – 3

CONCEPTS

This game reinforces JavaScript concepts for running the game. In this game, You have a car and its work is not to collide the vehicle coming from different direction.
To play the game, I must follow these steps:-

- Firstly, you will see the display screen consisting of three buttons start, help and exit.
- To start the game, I need to press START BUTTON and then PLAY BUTTON.
- If you want to see the instruction of the game press the HELP BUTTON.
- If you want to exit the game press the EXIT Button.
- To move the car, use “left arrow” for left, “right arrow” for right.
- The aim of the game is to avoid the car coming from different direction.
- You get scored according to the avoiding the car.
- There is no concept of lives, once your car collides to another vehicle, the game overs here.
- At the end you’ll see the GAME OVER!!! Message and the RESTART BUTTON.

EVENTS AND ITS OUTCOMES: -

S.NO.	EVENTS	OUTCOME
1.	CLICK ON START BUTTON	START THE GAME
2.	CLICK ON HELP BUTTON	GET THE GAME INSTRUCTION
3.	CLICK ON EXIT BUTTON	EXIT FROM THE GAME
4.	CLICK ON LEFT ARROW BUTTON OR KEYBOARD LEFT ARROW	CAR WILL MOVE IN LEFT DIRECTION
5.	CLICK ON RIGHT ARROW BUTTON OR KEYBOARD RIGHT ARROW	CAR WILL MOVE IN RIGHT DIRECTION
6.	CLICK ON PAUSE BUTTON	TO STOP AND START THE GAME
7.	CLICK ON RESTART BUTTON	TO RESTART THE GAME

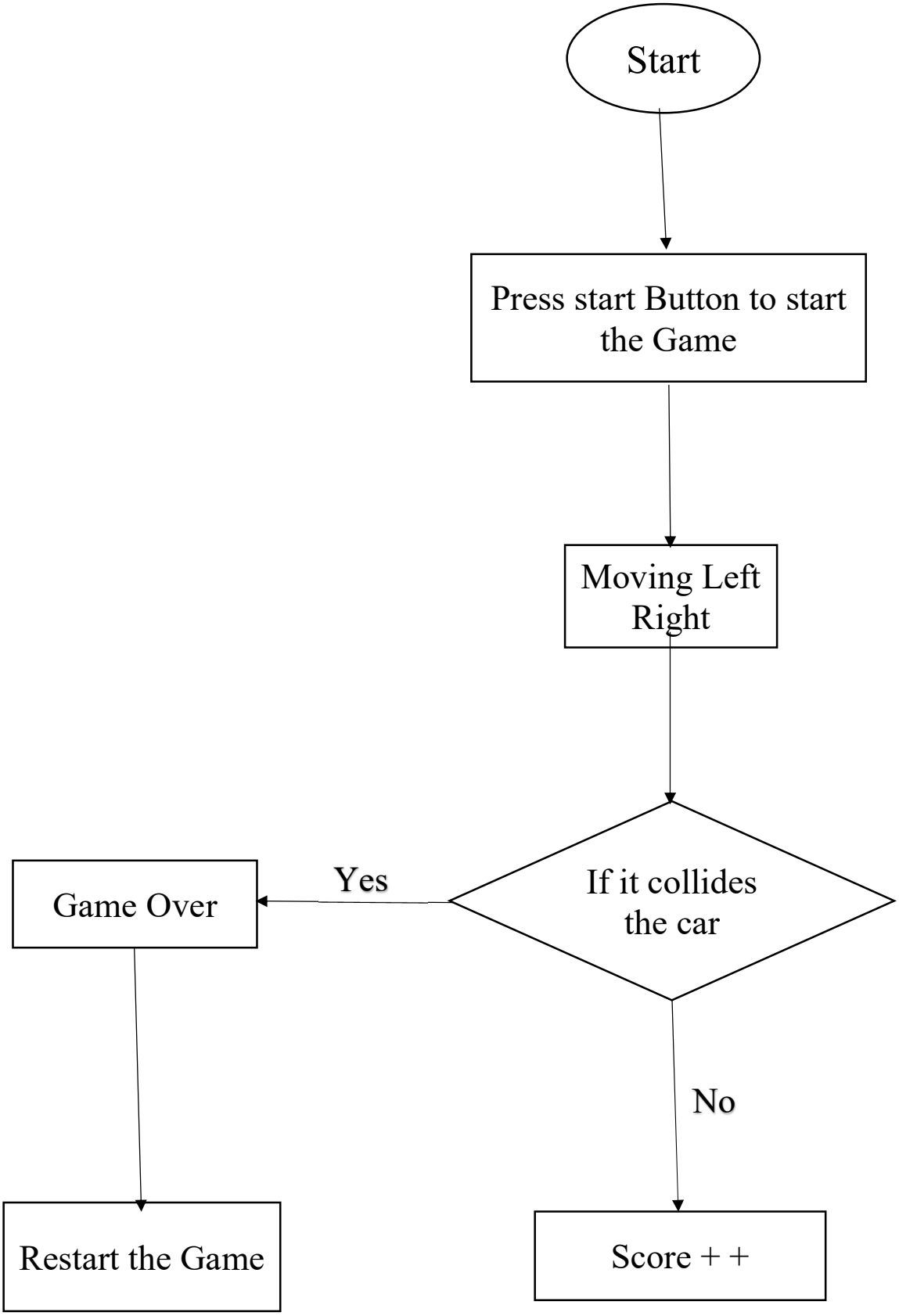
CHAPTER 4

FLOW CHART

A flow chart is a picture of the separate steps of a process in sequential order.

It is a generic tool that can be adopted for a wide variety of purposes and can be used to describe various processes, such as – A manufacturing process, an administrative or service process, or a project plan.

Below represented flow chart defines how our game runs as Ill as the functionality of the game.


Start
Press start Button to start the Game
Moving Left Right
If it collides the car
Yes
Game Over
Restart the Game
No
Score + +

CHAPTER – 5

CODING

HTML CODE: -

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,height=device-height,initial-scale=1.0">
    <title>Game</title>
    <link rel="stylesheet"  text="text/css" href="style.css">
    <script>alert("Bumping Car!!! You can also play this game with keyboard. Do you want to play this game so press OK");</script>
</head>
<body onload="Load();">
        <center><div id="board">
        <label>Score:</label>
        <label id="updatescore">0</label>
        <button onclick="togglestop(this);">&amp;#10074;&amp;#10074;</button>
    </div>
<canvas id="mycanvas"></canvas>
    <div id="controls">
        <button onclick="moveLeft();">&amp;larr;</button>
        <button onclick="moveRight();">&amp;rarr;</button>
    </div></center>
<!--Screens-->
 <section>
 <div class="color"></div>
   <div class="color"></div>
```

```
    <div class="color"></div>
</section>
<div class="screen" id="Waitscreen" >
    <label>Wait Loading images</label>
</div>
<div class="screen" id="startingscreen">
    <button onclick="startGame();" class="funkybutton">Play</button><br></div>
    <div class="screen" id="gameoverscreen" style="display: none;">
        <label id="finalscore">0</label><br>
        <button onclick="restartGame();" class="funkybutton">Restart</button><br></div>
</body>
<script  src="script.js" text="text/JavaScript"></script>
</html>
```

CSS CODE: -

```
@import
url('https://fonts.googleapis.com/css?family=Poppins:200,300,400,500,600,700,800,900&displ
ay=swap');
*{
    margin: 0;
    padding: 0;
    box-sizing: border-box;
    font-family: 'Poppins',sans-serif;
}
body{
    overflow: hidden;
}
/* game css */
 #mycanvas{    background: #000;}
#controls{    position: relative ; bottom: 4px;}
#controls button{
    height: 30px;
    font-size: 30px;
    line-height: 0px;
    color: #fff;
    width: 49.5%;
    cursor: pointer;
    background-color: #008000;
    border: 0;
}
#controls button:hover{    background-color: #003E00;    }
#controls button:last-child{ float: right;}
#board{
    height: 30px;
```

```
    width: 100%;
    font-size: 30px;
}
#board button{
    float: right;
    height: 30px;
    width: 30px;
}
.screen{
    position: absolute;
    font-size: 40px;
    color:#fff;
    top: 0px;
    left: 0px;
    height: 100%;
    width:100%;
    background-color: rgba(0,0,0,0.8);
}
.funkybutton{
    position: absolute;
    top: 48%;
    left: 10%;
    color:#fff;
    width: 80%;
    font-size: 40px;
    height: 50px;
    border-style: none;
    border-radius: 20px;
    cursor: pointer;
    background-color: #008000;
```

```
}
.funkybutton:hover{    background-color: #003E00;    }
#finalscore{    position: absolute;
    left: 10%;
    top: 20%;}
#Waitscreen{z-index: 10;}
#Waitscreen label{
    position: absolute;
    top:40%;
    left: 10%;
    color: #fff;
}
```

JS CODE: -

```
var canvas,ctx,w,h,enemy=[],player,tmpenm = [],cols = [],clk,ty=0,gamOver = false,score =
0,scoreLabel,paused=false;
var yposs = [-300,-600,-400,-500,-250,-450];
var imglocs=[
   "https://image.ibb.co/hV2Bha/Ambulance.png",
   "https://image.ibb.co/gqVxNa/Audi.png",
   "https://image.ibb.co/dm58TF/Black_viper.png",
   "https://image.ibb.co/mup2oF/Mini_truck.png",
   "https://image.ibb.co/cuTv8F/Mini_van.png",
   "https://image.ibb.co/jOXNoF/Police.png",
   "https://image.ibb.co/cp4rha/taxi.png",
   "https://image.ibb.co/fv1F8F/truck.png"
];
   var scl = 0;
function Vector(x,y){
   this.x = x;    this.y = y;
   this.add = function(vec){this.x+=vec.x;this.y+=vec.y;}
}//Vector class
function constrain(n,max,min){return Math.max(min,Math.min(n,max));}// get Value in range
function Load(){
   w = window.innerWidth;
   if(w > 400) w = 400;
   h = window.innerHeight-30-30;
   if(h > 600) h = 640-60;
   for(var i=0;i<3;i++) cols.push(i*w/3+10);
   canvas = document.getElementById("mycanvas");
   scoreLabel = document.getElementById("updatescore");
   ctx = canvas.getContext("2d");
   window.addEventListener("keydown",KeyDown,true);
   canvas.height = h;    canvas.width  = w; // set canvas height
   ctx.strokeStyle = "#ffffff";// line color
   ctx.lineWidth = 10;// line width
   ctx.setLineDash([h/5,10]);// set Dashed Line
   scl = h/580;
   if(scl>1)sc = 1;
   //ctx.scale(scl,scl);
```

```
    var carImg = new Image();
    carImg.src = "https://image.ibb.co/gOuP2a/Car.png";
   // carImg.src= "Car.png";
    carImg.onload = function(){
        player = new vechcile(carImg);
        player.width = carImg.width;
        player.height = carImg.height;
        player.pos = new Vector(0,(h-player.height*scl));
        setupImages();
    }
}
function setupImages(){
    for(var i=0;i<imglocs.length;i++){
        var tmpimg = new Image();
        tmpimg.src = imglocs[i];
        if(i==imglocs.length-1)tmpimg.onload =
function(){document.getElementById("Waitscreen").style ="display:none";}
        var tmp = new vechcile(tmpimg);
        tmp.width = tmpimg.width;
        tmp.height = tmpimg.height;
        tmp.vel = new Vector(0,2);
        enemy.push(tmp);
        setupEnemy();
    }
}
function vechcile(img){
    this.pos = new Vector(0,0);
    this.image = img;
    this.vel = new Vector(0,0);
    this.width = img.width;
    this.height = img.height;
    this.show = function(){
            if(this.height == 0) this.height = 214;
            if(this.width == 0) this.width = 96;

            this.pos.x = constrain(this.pos.x,w-(this.width*scl),0);   // Check position within a
range
            ctx.drawImage(this.image,this.pos.x,this.pos.y,this.width*scl,this.height*scl);//
```

```
drawImage image at selected position
            }
   this.update = function(){   this.pos.add(this.vel);     }
   this.end = function(){      return (this.pos.y > h);    }
}
function line(sx,sy,ex,ey){
   ctx.beginPath();
   ctx.moveTo(sx,sy);
   ctx.lineTo(ex,ey);
   ctx.stroke();
}
// setup each enemy on selected index
function setup1Enemy(i){
   var ypos = yposs[Math.floor((yposs.length-1)*Math.random())];
   var xpos = cols[Math.floor(3*Math.random())];
   if(i==0&&tmpenm[i]!=undefined && xpos == tmpenm[1].pos.x) setup1Enemy(i);
   else if(i>0 && tmpenm[i]!=undefined &&  xpos == tmpenm[i-1].pos.x) setup1Enemy(i);
   else {
      var index = Math.floor((enemy.length-1)*Math.random());
      enemy[index].pos = new Vector(xpos,ypos);
      tmpenm[i] = enemy[index];
      ypos -= (2*player.height+100);
   }
}
//setup all enemy
function setupEnemy(){  for(var i=0;i<2;i++)    setup1Enemy(i); }
function loop(){
   ctx.clearRect(0,0,w/scl,h/scl); // clear canvas
   ty += 5;    if(ty > 100) ty=0; // update ty for line animation
   line(w/3,ty,w/3,h);
   line(2*w/3,ty-10,2*w/3,h);
   for(var i=0;i<tmpenm.length;i++){
      tmpenm[i].show();
      tmpenm[i].update();
      if(tmpenm[i].end()) {setup1Enemy(i); updatescore(); }
      if(checkColision(player,tmpenm[i])) {
         gamOver = true;
         stopGame();
```

```
            gameOver();
        }
    }
    player.show();
    if(!gamOver)clk = window.requestAnimationFrame(loop);
}

function checkColision(obj1,obj2){ // Check Colision BetIen two object
    return ((obj1.pos.x+obj1.width*scl >= obj2.pos.x)  &&  (obj2.pos.x+obj2.width*scl >=
obj1.pos.x) && (obj1.pos.y+obj1.height*scl >= obj2.pos.y) && (obj2.pos.y+obj2.height*scl
>= obj1.pos.y));
}
function updatescore(){ score++; scoreLabel.innerText = score;} // update Score to score label
function moveLeft(){   if(!paused) player.pos.add(new Vector(-10,0)); } // Move Left
function moveRight(){  if(!paused)  player.pos.add(new Vector(10,0)); } // Move right
function KeyDown(e){ // KeyDown on keyboard
    var key = e.keyCode || e.charCode;
            if(key == 39)      moveRight();
        else if(key == 37)      moveLeft();
}
function stopGame(){    window.cancelAnimationFrame(clk); } //stop the game
function togglestop(e){ // Pause Button
    if(e.innerText == "▮▮") {           e.innerText = "▶";       stopGame();  paused = true;}
    else if(e.innerText == "▶"){       e.innerText = "▮▮";       loop();      paused = false;}
}
//start the game
function startGame(){
    document.getElementById("startingscreen").style = "display:none;"; loop();
}
function gameOver(){
    document.getElementById("finalscore").innerText = "GameOver!!!!!!\n Score:"+score;
    document.getElementById("gameoverscreen").style = "display:block;";
}
function restartGame(){
    score = 0;scoreLabel.innerText = score;
    Load();    gamOver = false;
    document.getElementById("gameoverscreen").style = "display:none;";
    document.getElementById("startingscreen").style = "display:block;";
```

LINK FOR THE GAME APPLICATION: -

https://ashish3281.github.io/CAR-COLLISION-GAME/

LINK FOR THE GITHUB REPOSITORY: -

https://github.com/ashish3281/CAR- COLLISION-GAME

CHAPTER – 6

TESTING

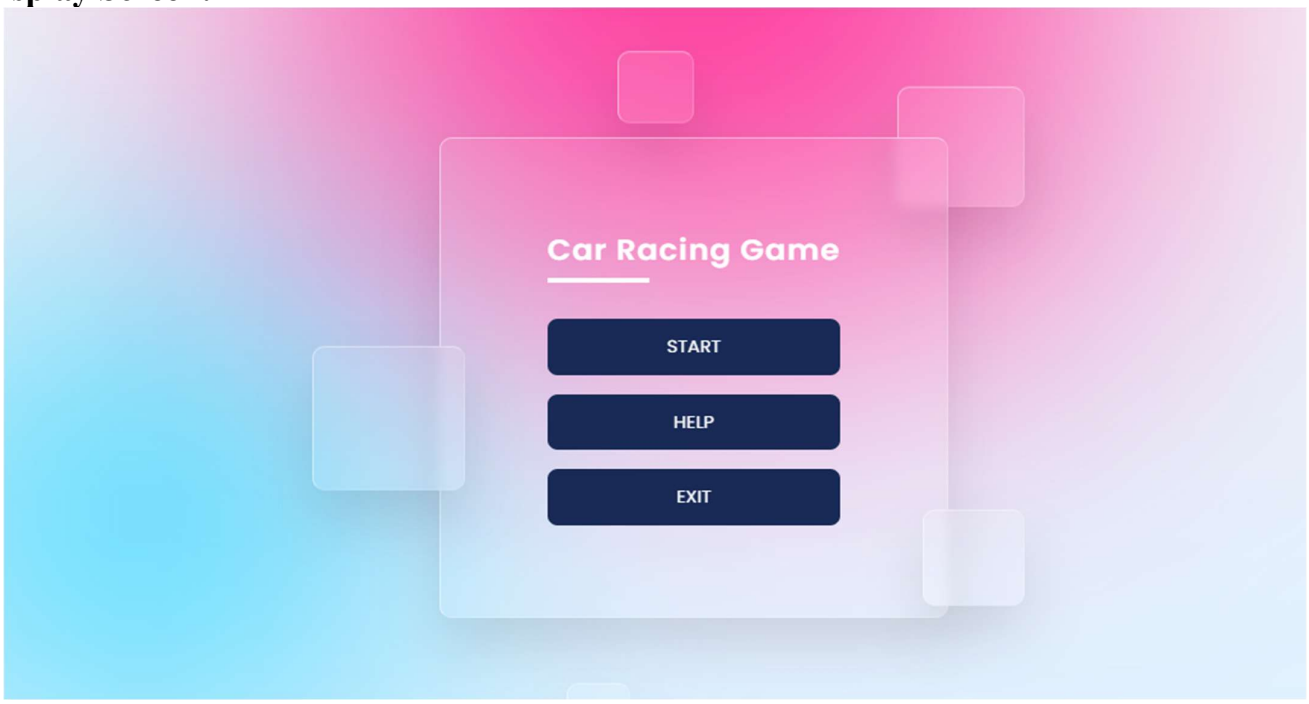
Testing is the major control measure used during software development. Its basic function is to detect errors in the software. During requirement analysis and design, the output is a document that is usually textual and no executable. After the coding phase, computer programs are available that can be executed for testing purpose. This implies that testing not only must uncover errors introduced during coding, but also errors introduced during previous phase. Thus, the goal of testing is to uncover the requirements, design, and coding errors in the programs. Each execution these test will be documented in a test indent report document.

S.NO	PROCEDURE	SUCCESS	OUTCOME
1	Click on start button and display game screen	Game Screen will display	PASS
2	Click on help button and display help screen	Help Screen will display	PASS
3	Click on exit button to exit the screen	Exit the screen	PASS
4	Click on restart button to restart the game	Restart the game	PASS

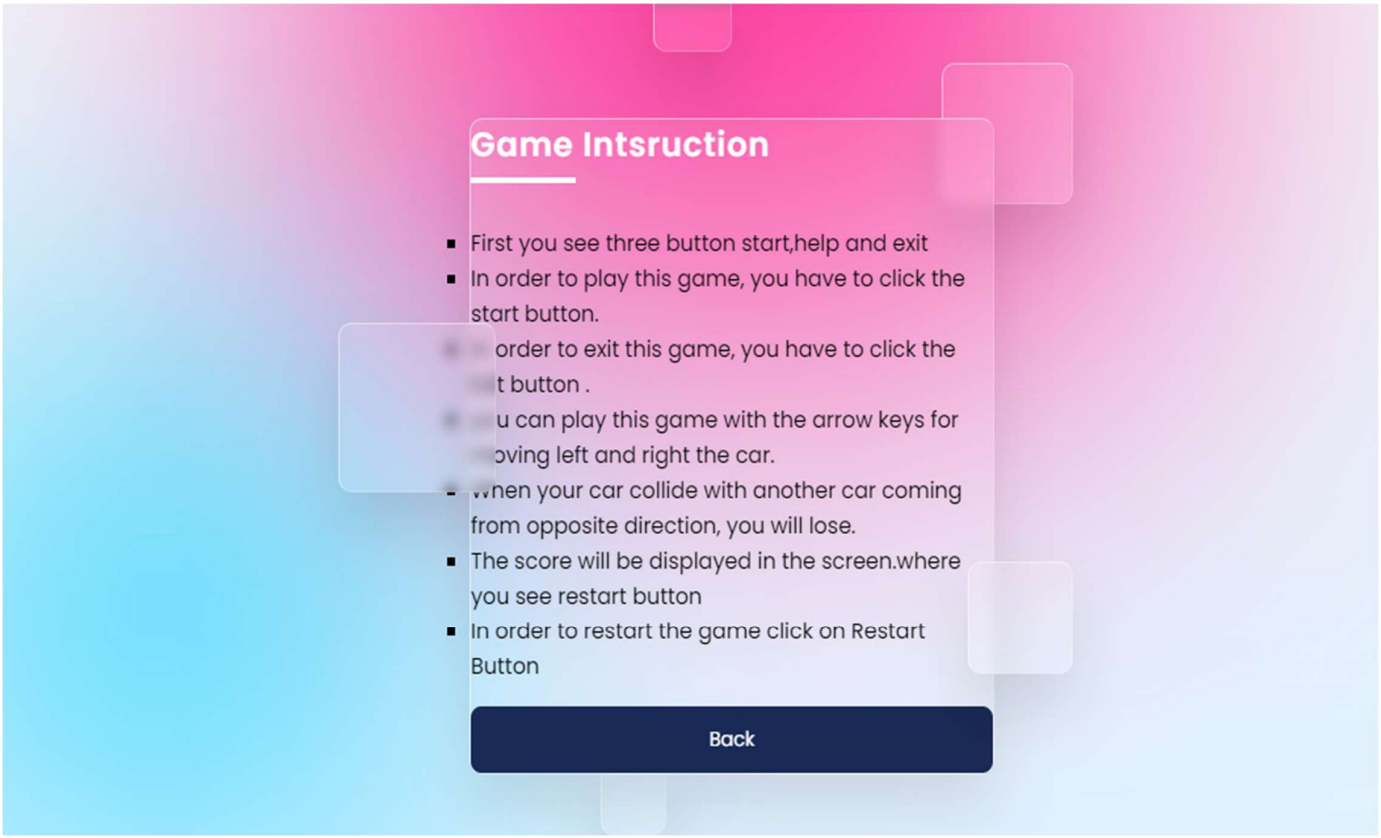
CHAPTER 7

OUTPUT SCREEN

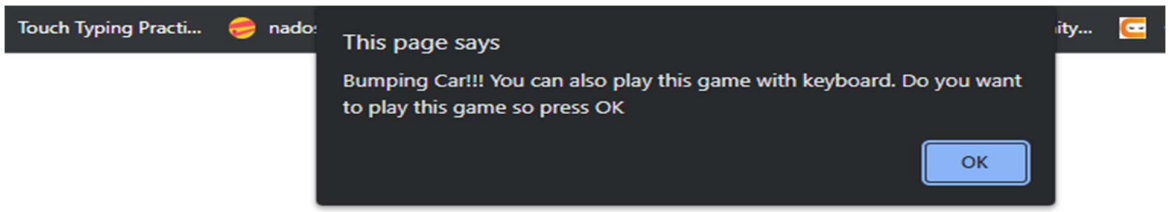
Display Screen:

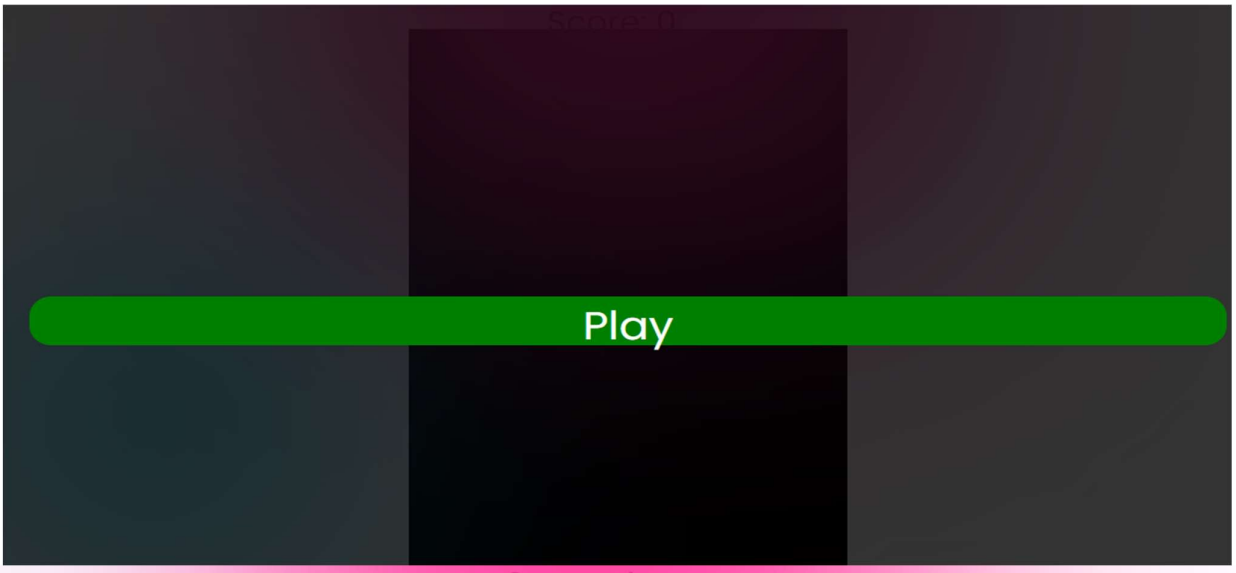


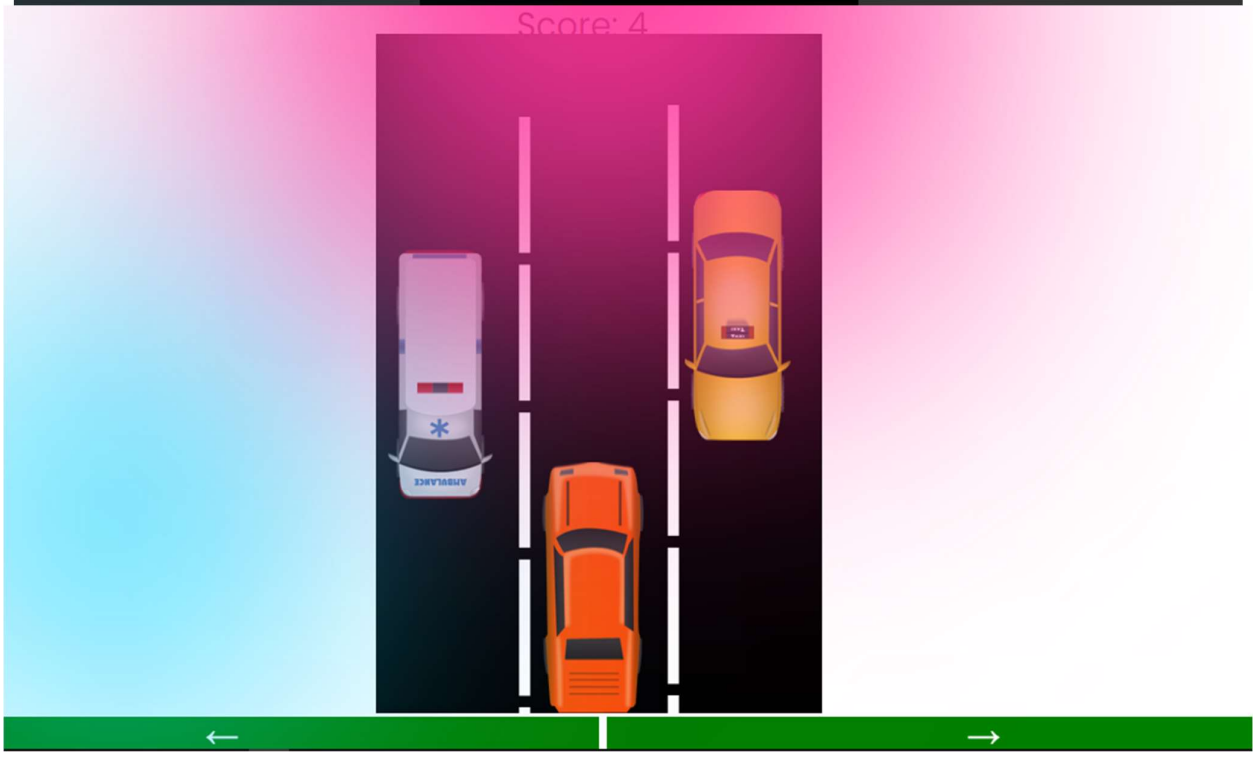
Help Screen:

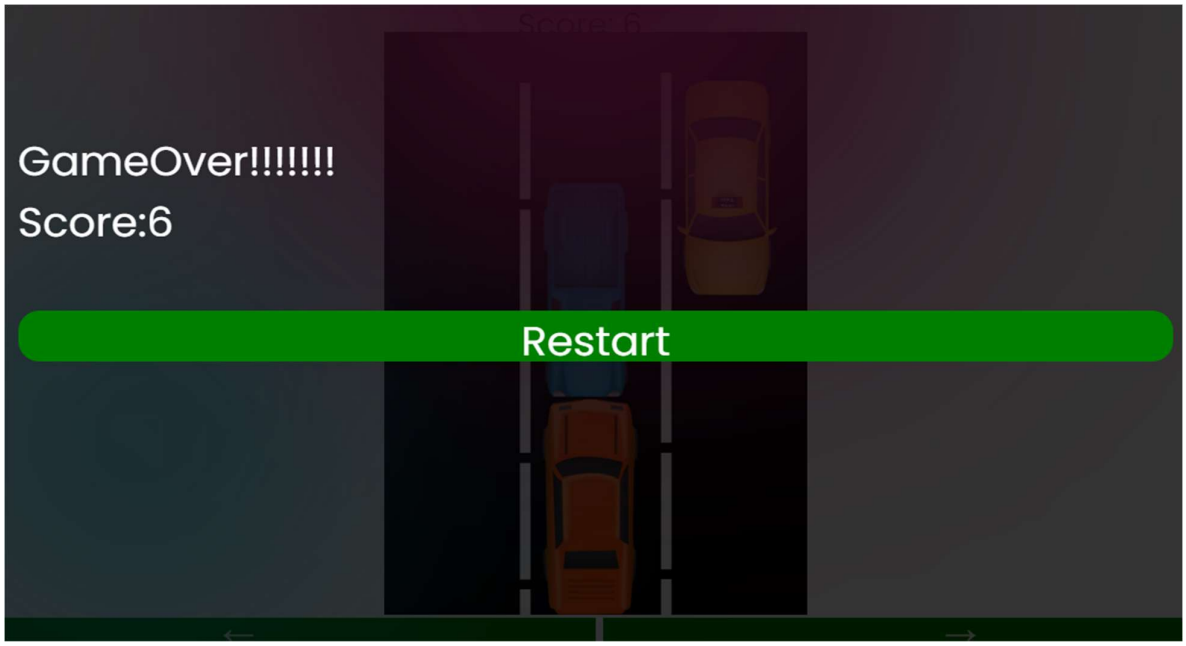


Game Screen:


Touch Typing Practi...
This page says
Bumping Car!!! You can also play this game with keyboard. Do you want to play this game so press OK
OK


Play


Score: 4
←
→


GameOver!!!!!!!
Score:6
Restart

CHAPTER -8

CONCLUSION & FUTURE SCOPE

CONCLUSION:

The project named **CAR COLLISION GAME** has been studied successfully above. All the requirements as well as working of the project along with the technology stack used, has been explained carefully keep in mind each and every detail.

I have completed our knowledge of frontend web Designing after I finished the project and obtained more experience of website designing. The techniques (HTML, CSS, JAVASCRIPT) used for navigation of the car could be generalized to apply to various other problems of interest.

The work presented in this paper provides innumerable opportunities for further investigation into the evaluation of task.

FUTURE SCOPE:

My Project will be able to implement in future after making some changes and modifications as I make our project at a very low level. So, the modifications that can be done in our project are:
It can be made with good graphics.
And I can add more options like adding player's profiles, multiple levels and also, I can add multiplayeroption.

CHAPTER -9
BIBLIOGRAPY

Book Reference:

- Learning How to Design Web Pages: "HTML AND CSS" - DESIGN AND BUILD WEBSITES, BY JON DUCKETT"
- React Quickly: Painless Web Apps with React-By Azat Mardan.

Website References:

- Simplilearn.com-

  https://simiplilearn.com

- Other Resources

  https://www.w3schools.com/

  https://www.geeksforgeeks.org/

- MDN Web docs

  https://developer.mozilla.org/en-US/