

# MINI PROJECT REPORT ON

WebGL

*Submitted in partial fulfillment of the requirements for the award of the degree of*

**Bachelor of Technology**

**In**

**Computer Science And Engineering**

*By*

**Ramiz Balayil**

**Vivek Gupta**

**Rohith Vijay**



**FEDERAL INSTITUTE OF SCIENCE AND TECHNOLOGY  
(FISAT)**

**ANGAMALY-683577, ERNAKULAM (DIST)**

*Affiliated to*

**MAHATMA GANDHI UNIVERSITY**

**Kottayam-686560**

**May 2015**

**FEDERAL INSTITUTE OF SCIENCE AND TECHNOLOGY (FISAT)**

Mookkannor(P.O), Angamaly-683577



**CERTIFICATE**

This is to certify that project report titled **WebGL** is a bonafide work carried out by Ramiz Balayil(12003792), Vivek Gupta(12003826), Rohith Vijay(12003800) in partial fulfilment for the award of Bachelor of Technology in Computer Science and Engineering from Mahatma Gandhi University, Kottayam, Kerala during the academic year 2014-2015.

**Staff In-charge**

**Head of the Department**

**Place:**

**Date:**

**Internal Examiner**

**External Examiner**

## ABSTRACT

In this project we have implemented the classic game of concentration in a 3D environment using WebGL. Memory games are games that require players to match similar elements. Some of the popular games in this genre are Dominoes, board games such as mahjong solitaire and card games such as Concentration or Rummy. In this modification, we have introduced 8 crates that jump up and down in an ocean and changes the color of its texture on mouse click. The aim is to consequently click on crates that reveal matching colors and this will lead to the vanishing of the crates. The crates would turn back to its default texture on a mismatch. The player has to remove all the crates from the ocean. A score is maintained based on the number of mouse clicks that has been input.

## ACKNOWLEDGEMENT

This project was supported by Mr. Pankaj Kumar from the Computer Science Department of Federal Institute of Science And Technology. We thank our classmates who provided insight and expertise that greatly assisted the project.

We would also like to show our gratitude to Mrs. Reshmi R and Mrs. Nisha Rajeev for sharing their pearls of wisdom with us during the course of this project and for their comments on an earlier version of the game, although any errors are our own and should not tarnish the reputations of these esteemed persons.

Ramiz Balayil, Rohith Vijay, Vivek Gupta

# CONTENTS

List of Figures . . . . .	i
List of Tables . . . . .	ii
<b>1 Introduction</b>	<b>1</b>
1.1 Overview . . . . .	1
1.2 Objective . . . . .	1
<b>2 Related Works</b>	<b>2</b>
2.1 Background Study . . . . .	2
2.2 Existing System . . . . .	6
<b>3 Design</b>	<b>7</b>
3.1 Introduction . . . . .	7
3.2 System Architecture . . . . .	7
3.3 Logical Design . . . . .	7
3.3.1 Algorithm . . . . .	8
3.3.2 Flow Chart . . . . .	8
3.3.3 Data Flow Diagram . . . . .	9
3.4 System Requirement . . . . .	10
3.4.1 Hardware Requirement . . . . .	10
3.4.2 Software Requirement . . . . .	10
<b>4 Testing</b>	<b>12</b>
4.1 Introduction . . . . .	12
4.2 White-box Testing . . . . .	12
4.3 Unit Testing . . . . .	12
4.4 Integration Testing . . . . .	13
<b>5 Implementation</b>	<b>14</b>
5.1 Introduction . . . . .	14
5.2 Implementation Procedure . . . . .	14
5.3 Area of implementation . . . . .	15
5.3.1 Three.js . . . . .	15
5.3.2 An HTML page . . . . .	15
<b>6 Conclusion</b>	<b>16</b>

<b>APPENDIX</b>	<b>17</b>
<b>A Source Code</b>	<b>18</b>
A.1 Adding the wooden crate . . . . .	18
A.2 Introducing Mouse Click . . . . .	18
A.3 Implementing random colors . . . . .	20
A.4 Keeping Score . . . . .	20
<b>B Screenshots</b>	<b>22</b>
B.1 Crates - A 3D WebGL game . . . . .	22

## LIST OF FIGURES

2.1	Memory Game . . . . .	6
3.1	Flow Chart . . . . .	8
3.2	Data Flow Diagram . . . . .	9
3.3	Data Flow Diagram . . . . .	9
3.4	Data Flow Diagram . . . . .	10
B.1	Crates - Welcome Screen . . . . .	22
B.2	The game in play . . . . .	23

## LIST OF TABLES



# Chapter 1

## Introduction

WebGL is the new standard for 3D graphics on the Web. With WebGL, developers can harness the full power of the computer's graphics rendering hardware using only JavaScript, a web browser, and a standard web technology stack. WebGL is part of the HTML5 family of technologies. While not in the official specification, it is shipped with most browsers that support HTML5. Like Web Workers, Web Sockets, and other technologies outside the official W3C recommendations, WebGL is an essential component in an emerging suite that is transforming the modern browser into a first-class application platform.

### 1.1 Overview

In this project we have implemented a memory game in a 3D environment using WebGL. The game consists of several crates that jump up and down in an ocean. In the same way that the classic game of concentration is played, every crate once clicked changes the color of its texture. The wooden texture of the crate changes to any of the four colors: red, blue, yellow and green. There is a total of 8 crates and the four colors are assigned randomly to the crates at the start of the game. The player has to click on the crates to reveal the color. If a matching color is found on the consequent click the crate vanishes, otherwise the crates change to its default texture. A score is kept based on the number of clicks that the player has input.

### 1.2 Objective

The basic objectives of the project are:

- To develop a WebGL game.
- To develop a webpage to run the game.
- To provide a very good game playing experience.

# Chapter 2

## Related Works

### 2.1 Background Study

WebGL is basically a low-level JavaScript interface to the OpenGL ES 2.0 feature set (using OpenGL/ES 2.0 or DirectX behind the scenes). A common misconception is that WebGL is for 3D only, it is actually not true. By leveraging low-level primitives you can either recreate higher level constructs for 3D content, like three.js does, or simply 2D primitives to power 2D. Two triangles form a quad, which a texture can be applied to, and there you have it, a 2D image GPU accelerated. WebGL will not only help gaming, it will also be useful to power interactive experimentations or parts of a digital marketing websites. WebGL elements can be mixed with other HTML elements and composited with other parts of the page or page background. WebGL programs consist of control code written in JavaScript and shader code that is executed on a computers Graphics Processing Unit (GPU). WebGL is designed and maintained by the non-profit Khronos Group. WebGL offers Tight integration with HTML content, including layered compositing, interaction with other HTML elements etc.

Three.js allows the creation of GPU-accelerated 3D animations using the JavaScript language as part of a website without relying on proprietary browser plugins. Writing programs in WebGL is a hefty task and this is the reason we use already defined libraries like Three.js. Three.js includes the following features:

- Renderers: Canvas, SVG and WebGL.
- Effects: Anaglyph, cross-eyed and parallax barrier.
- Scenes: add and remove objects at run-time; fog
- Cameras: perspective and orthographic; controllers: trackball, FPS, path and more
- Animation: armatures, forward kinematics, inverse kinematics, morph and keyframe

- Lights: ambient, direction, point and spot lights; shadows: cast and receive
- Materials: Lambert, Phong, smooth shading, textures and more
- Shaders: access to full OpenGL Shading Language (GLSL) capabilities: lens flare, depth pass and extensive post-processing library
- Objects: meshes, particles, sprites, lines, ribbons, bones and more - all with Level of detail
- Geometry: plane, cube, sphere, torus, 3D text and more; modifiers: lathe, extrude and tube
- Data loaders: binary, image, JSON and scene 2
- Utilities: full set of time and 3D math functions including frustum, matrix, quaternion, UVs and more
- Export and import: utilities to create Three.js-compatible JSON files from within: Blender, openCTM, FBX, Max, and OBJ
- Support: API documentation is under construction, public forum and wiki in full operation
- Examples: Over 150 files of coding examples plus fonts, models, textures, sounds and other support files
- Debugging: Stats.js, WebGL Inspector, Three.js Inspector

Three.js runs in all browsers supported by WebGL. HTML5 is a markup language for structuring and presenting content for the World Wide Web and a core technology of the Internet. Its core aims have been to improve the language with support for the latest multimedia while keeping it easily readable by humans and consistently understood by computers and devices (web browsers, parsers, etc.). In particular, HTML5 adds many new syntactic features. These include the new `<video>`, `<audio>` and `<canvas>` elements, as well as the integration of scalable vector graphics (SVG) content (that replaces the uses of generic `<object>` tags) and MathML for mathematical formulas. These features are designed to make it easy to include and handle multimedia and graphical content on the web without having to resort to proprietary plugins and APIs. Other new elements, such as `<section>`, `<article>`, `<header>` and `<nav>`, are designed to enrich the semantic content of documents. The APIs and Document Object Model (DOM) are no longer afterthoughts, but are fundamental parts of the HTML5 specification.

HTML5 also defines in some detail the required processing for invalid documents so that syntax errors will be treated uniformly by all conforming browsers and other user agents. It was

specially designed to deliver rich content without the need for additional plugins. The HTML5 version delivers everything from animation to graphics, music to movies, and can also be used to build complicated web applications. HTML5 is also cross-platform. It is designed to work whether you are using a PC, or a Tablet, a Smartphone, or a Smart TV.

Features:

- HTML5 introduces elements and attributes that reflect typical usage on modern websites.
- In addition to specifying markup, HTML5 specifies scripting application programming interfaces (APIs) that can be used with JavaScript.
- The HTML5 specification gives detailed rules for lexing and parsing, with the intent that different compliant browsers will produce the same result in the case of incorrect syntax.
- The `<video>` and `<audio>` elements for media playback.
- Support for local storage.
- New content-specific elements, like `<article>`, `<footer>`, `<header>`, `<nav>`, `<section>`.
- New form controls, like calendar, date, time, email, url, search.
- The `<canvas>` element for 2D drawing.
- HTML 5 will enable web designers to use cleaner, neater code, like remove most div tags and replace them with semantic HTML 5 elements.
- HTML5 enables designer to use more fancier forms.
- HTML5 has more greater consistency than the previous versions.
- Improved Accessibility.
- Offline Application cache.
- Geolocation support

Netscape invented JavaScript, and JavaScript was first used in Netscape browsers. However, Netscape is working with Ecma International the European association for standardizing information and communication systems (ECMA was formerly an acronym for the European Computer Manufacturers Association) to deliver a standardized, international programming language based on core JavaScript. This standardized version of JavaScript, called ECMAScript, behaves the same way in all applications that support the standard.

JavaScript is loosely based on Java and it is built into all the major modern browsers. A javascript language is a lightweight programming language. JavaScript code can be inserted into any HTML page, and it can be executed by all types of web browsers. JavaScript (JS) is a dynamic computer programming language. It is most commonly used as part of web browsers, whose implementations allow client-side scripts to interact with the user, control the browser, communicate asynchronously, and alter the document content that is displayed. It is also being used in server-side programming, game development and the creation of desktop and mobile applications. JavaScript is a prototype-based scripting language with dynamic typing and has first-class functions. Its syntax was influenced by C. JavaScript copies many names and naming conventions from Java, but the two languages are otherwise unrelated and have very different semantics. The key design principles within JavaScript are taken from the Self and Scheme programming languages. It is a multi-paradigm language, supporting object-oriented, imperative, and functional programming styles. JavaScript follows most Java expression syntax, naming conventions and basic control-flow constructs which was the reason why it was renamed from LiveScript to JavaScript. Features of javascript:

- JavaScript is a lightweight, interpreted programming language
- Designed for creating network-centric applications
- Complementary to and integrated with Java
- Complementary to and integrated with HTML
- Open and cross-platform

First, the JavaScript code gets a 3D context from an HTML5 canvas element. Then it registers a set of shaders, which are written in GLSL (OpenGL Shading Language) and essentially resemble C code. The rest of the process is very modular. You need to get vertex data and any other information you intend to use (such as vertex colors, texture coordinates, and so forth) down to the graphics pipeline using uniforms and attributes which are defined in the shader, but the exact layout and naming of this information is very much up to the developer. JavaScript sets up the initial data structures and sends them to the WebGL API, which sends them to either ANGLE or OpenGL ES, which ultimately sends it off to the graphics hardware.

## 2.2 Existing System

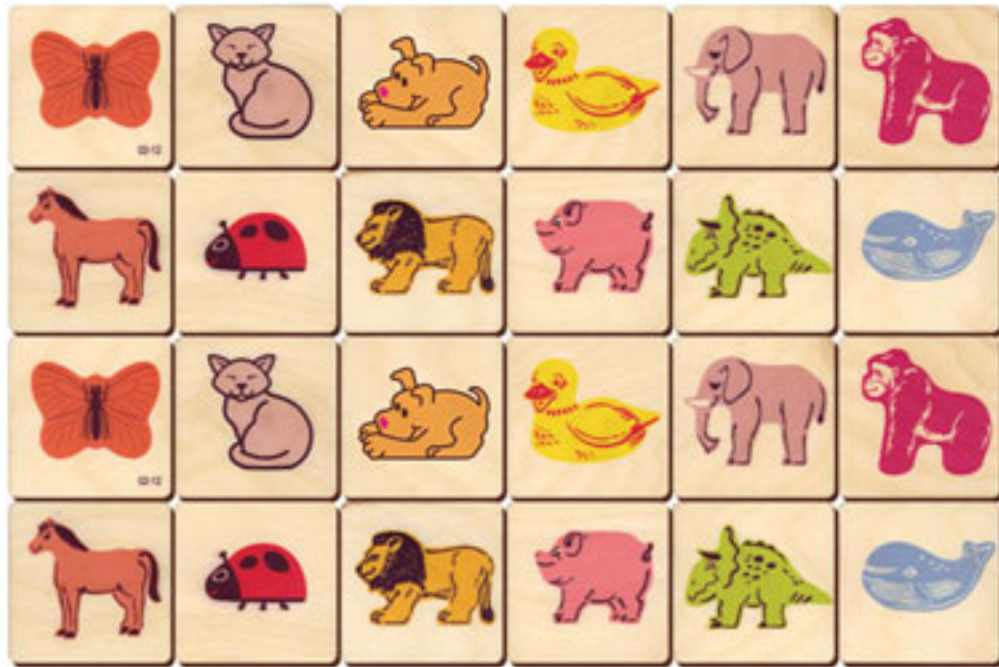


Figure 2.1. A Memory Game

# Chapter 3

## Design

### 3.1 Introduction

Design is the first step in the development phase for any engineered product or system. It may be defined as: “The process of applying different techniques and principles for the purpose of defining a device, a process or a system in sufficient detail to permit its physical realization.” Detail design focus on refinements to architectural representation that lead to detailed data structure and algorithm representation for software. The data design transforms the information domain model created during analysis into the datastructures that will be required to implement the software.

### 3.2 System Architecture

The JavaScript library which we have used for this project is three.js.

Three.js is a lightweight cross-browser JavaScript library/API used to create and display animated 3D computer graphics on a Web browser. Three.js uses WebGL. Three.js allows the creation of GPU-accelerated 3D animations using the JavaScript language as part of a website without relying on proprietary browser plugins. This is possible thanks to the advent of WebGL.

High-level libraries such as Three.js or GLGE, SceneJS, PhiloGL or a number of other libraries make it possible to author complex 3D computer animations that display in the browser without the effort required for a traditional standalone application or a plugin.

### 3.3 Logical Design

In the logical design, description of the inputs, outputs, databases and procedures are given a format that meets the requirements.

### 3.3.1 Algorithm

1. Start.
2. Load the game environment.
3. The user can play the game, by clicking on the object.
4. If two similar objects are found, they are removed from the scene.  
Otherwise, they go back to their original state.
5. Score is updated as per the mouse clicks.
6. The game is continued till all the objects disappear.
7. Stop

### 3.3.2 Flow Chart

Flow Chart

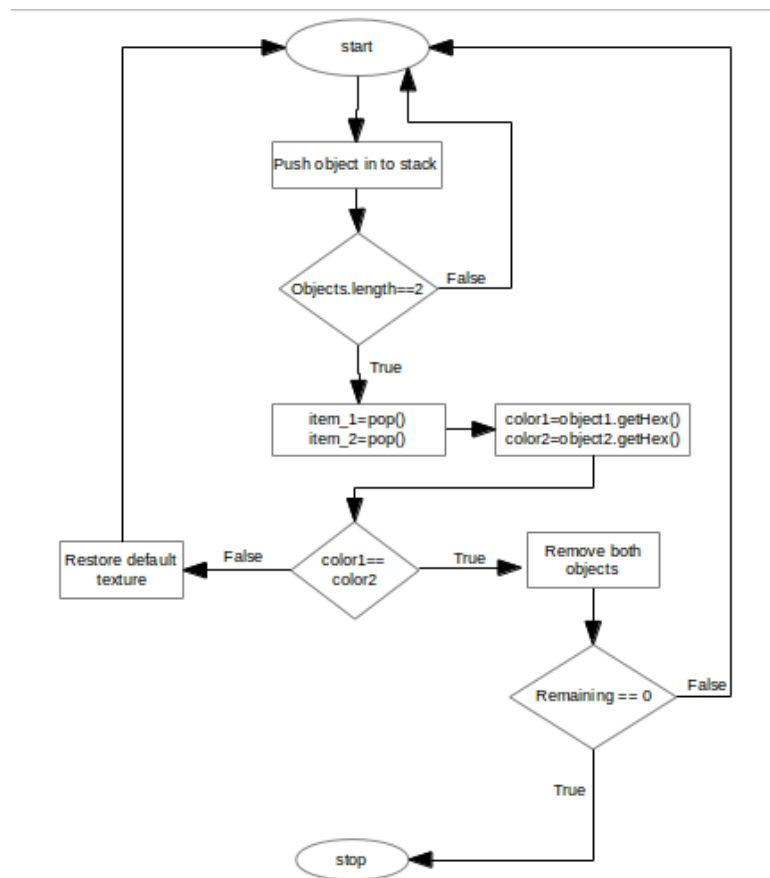


Figure 3.1. Flow Chart



### 3.3.3 Data Flow Diagram

Data Flow Diagram-Level 0

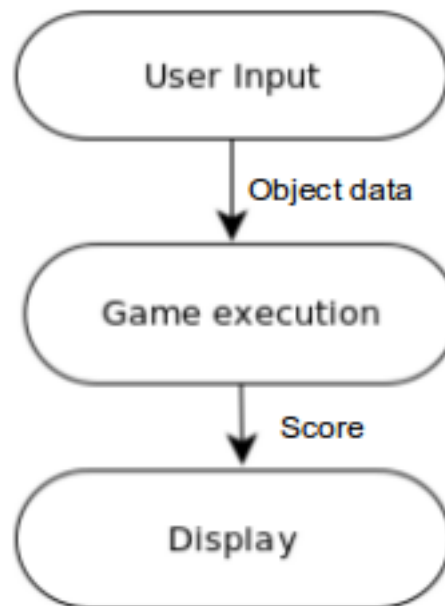


Figure 3.2. Level 0

Data Flow Diagram-Level 1

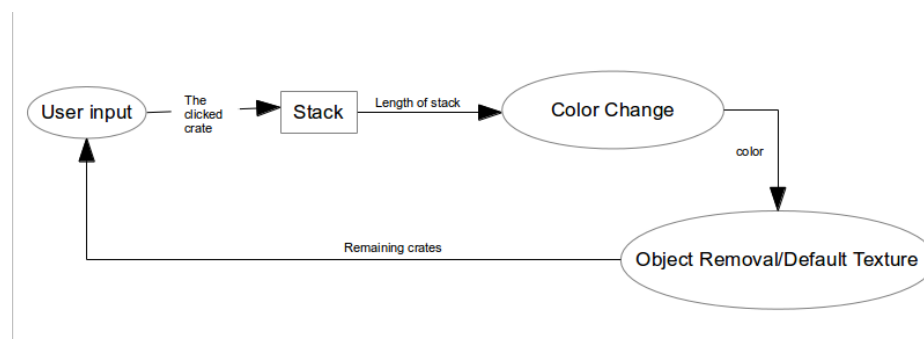


Figure 3.3. Level 1

Data Flow Diagram-Level 2

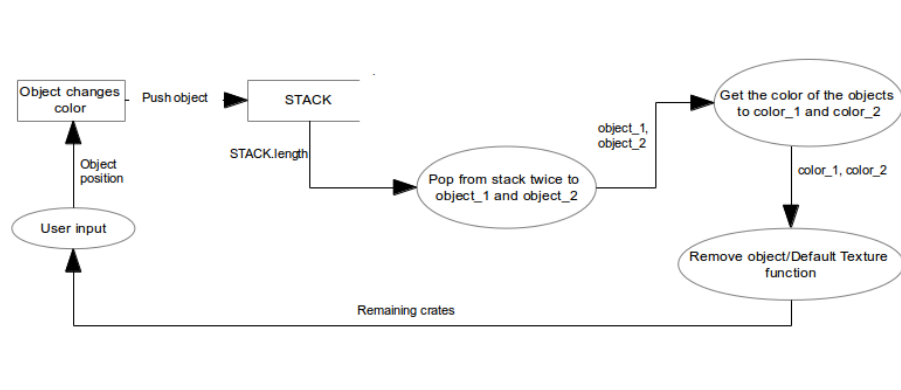


Figure 3.4. Level 2

## 3.4 System Requirement

Operating System: LINUX (Debian) An operating system is a program that acts as an interface between a user and a hard disk. The OS manages computer hardware and system resources, such as memory and hard disk. It also manages the interaction between users and the computers by accepting input from the user, interpreting the input and generating the output. The OS forms the base on which any application software is developed

### 3.4.1 Hardware Requirement

Processor: PENTIUM IV or above

RAM: 1 GB

Graphics card memory : Recommended are at least 256 MB

### 3.4.2 Software Requirement

Operating System: LINUX (Debian) An operating system is a program that acts as an interface between a user and a hard disk. The OS manages computer hardware and system resources, such as memory and hard disk. It also manages the interaction between users and the computers by accepting input from the user, interpreting the input and generating the output. The OS forms the base on which any application software is developed.

Programming Language: HTML5 HTML5 is the latest and most enhanced version of HTML[2]. Technically, HTML is not a programming language, but rather a markup language. HTML5 is the next major revision of the HTML standard superseding HTML 4.01, XHTML 1.0, and XHTML 1.1. HTML5 is a standard for structuring and presenting content on the World

Wide Web.HTML5 is a cooperation between the World Wide Web Consortium (W3C) and the Web Hypertext Application Technology Working Group (WHATWG). The new standard incorporates features like video playback and drag-and-drop that have been previously dependent on third-party browser plug-ins such as Adobe Flash,Microsoft Silverlight,and Google Gears.

**JavaScript:** Javascript is a scripting language produced by Netscape for use within HTML Web pages. JavaScript is loosely based on Java and it is built into all the major modern browsers. Features of javascript: JavaScript is a lightweight, interpreted programming language Designed for creating network-centric applications Complementary to and integrated with Java Complementary to and integrated with HTML Open and cross-platform.

**WebGL:** WebGL (Web Graphics Library) is a JavaScript API for rendering interactive 3D graphics and 2D graphics within any compatible web browser without the use of plug-ins. WebGL is integrated completely into all the web standards of the browser allowing GPU accelerated usage of physics and image processing and effects as part of the web page canvas. WebGL elements can be mixed with other HTML elements and composited with other parts of the page or page background. WebGL programs consist of control code written in JavaScript and shader code that is executed on a computers Graphics Processing Unit (GPU). WebGL is designed and maintained by the non-profit Khronos Group.

WebGL elements can be mixed with other HTML elements and composited with other parts of the page or page background. WebGL programs consist of control code written in JavaScript and shader code that is executed on a computers Graphics Processing Unit (GPU). WebGL is designed and maintained by the non-profit Khronos Group.WebGL offers Tight integration with HTML content, including layered compositing, interaction with other HTML elements etc. Browser Support: The latest versions of Apple Safari, Google Chrome, Mozilla Firefox, and Opera all support many HTML5 features and Internet Explorer 9.0 will also have support for some HTML5 functionality.

# Chapter 4

## Testing

### 4.1 Introduction

Software testing is an investigation conducted to provide stakeholders with information about the quality of the product or service under test. Software testing can be stated as the process of validating and verifying that a software program application product meets the requirements that guided its design and development, works as expected and can be implemented with the same characteristics. The mechanism for determining whether a software program or system has passed or failed is known as a test oracle. In some settings an oracle could be a requirement or a use case, while in other systems it might be a heuristic. It may take several test cases to determine whether a software program or system is working properly.

### 4.2 White-box Testing

The aim is to check whether all lines of the code are executed at least once. This can be done by checking the following conditions.

1. Checking whether all the objects in the scene has been completely loaded.
2. Checking the mouse click is working on all the crates.
3. Checking the crates are disappearing when similar ones are found.
4. Checking if the crates are going to back to its original state, in case, if the two are dissimilar.
5. Checking if the score is being updated properly.

### 4.3 Unit Testing

The game basically consists of three parts.

1. The Game Environment
2. Crates Animation
3. Mouse Click

**The Game Environment**

The Game begins showing off a title page. From there, it passes to game page. The game environment is loaded. There is a help and about section at the top. Background music is initiated. All these functions were observed to be working perfectly upon testing.

**Crates Animation**

There are eight crates present in the game. The crates bounce up and down continuously. They rotate as well. This unit was tested successfully.

**Mouse Click**

This unit was tested by clicking on each of the crates. The colour of the crates were changed upon mouse click. If two similar crates are found, they disappear. Otherwise, they go back to their original state. This unit was found to be working successfully.

## **4.4 Integration Testing**

The game environment was loaded first and it calls other classes. Upon integrating and executing the entire system, the background music, mouse click blended in perfectly. The game was tested with gamers and no bugs were reported. And thus, the phase of testing was concluded.

# Chapter 5

## Implementation

### 5.1 Introduction

The implementation is the stage in the project where the theoretical design is turned into a working system, and gives confidence to the new system for the users that it will work correctly and efficiently. It is an activity that is contained throughout the development phase. It is the process of bringing a developed system into operation use and turning it over to the user. The new system and its components are to be tested in a structural and planned manner. A successful system is to be delivered and user should have the confidence that the system would work both efficiently and effectively.

### 5.2 Implementation Procedure

The programming language used is JavaScript. Javascript is an easy-to-use programming language that can be embedded in the header of your web pages. It can enhance the dynamics and interactive features of your page by allowing you to perform calculations, check forms, write interactive games, add special effects, customize graphics selections, create security passwords and more. JavaScript, also known as ECMAScript (the untrademarked name used for the standard), is a dynamic programming language. It is most commonly used as part of web browsers, whose implementations allow client-side scripts to interact with the user, control the browser, communicate asynchronously, and alter the document content that is displayed. It is also used in server-side network programming with runtime environments such as Node.js, game development and the creation of desktop and mobile applications. With the rise of the single-page web app and JavaScript-heavy sites, it is increasingly being used as a compile target for source-to-source compilers from both dynamic languages and static languages.

## 5.3 Area of implementation

### 5.3.1 Three.js

Three.js is a lightweight cross-browser JavaScript library/API used to create and display animated 3D computer graphics on a Web browser. Three.js uses WebGL. The source code is hosted in a repository on GitHub. Three.js allows the creation of GPU-accelerated 3D animations using the JavaScript language as part of a website without relying on proprietary browser plugins. This is possible thanks to the advent of WebGL. High-level libraries such as Three.js or GLGE, SceneJS, PhiloGL or a number of other libraries make it possible to author complex 3D computer animations that display in the browser without the effort required for a traditional standalone application or a plugin.

This library contains many standard graphical webpages implemented using JavaScript. We have collaborated two standard models.

- An ocean overlooked by the sky.
- A wooden crate rotating over its axis.

The result of the collaboration was extended to many crates moving up and down in their Y-axis using an array.

### 5.3.2 An HTML page

HTML is a computer language devised to allow website creation. These websites can then be viewed by anyone else connected to the Internet. It is relatively easy to learn, with the basics being accessible to most people in one sitting; and quite powerful in what it allows you to create. It is constantly undergoing revision and evolution to meet the demands and requirements of the growing Internet audience under the direction of the W3C, the organisation charged with designing and maintaining the language. HTML consists of a series of short codes typed into a text-file by the site author – these are the tags. The text is then saved as a html file, and viewed through a browser, like Internet Explorer or Netscape Navigator. This browser reads the file and translates the text into a visible form, hopefully rendering the page as the author had intended. Writing your own HTML entails using tags correctly to create your vision. You can use anything from a rudimentary text-editor to a powerful graphical editor to create HTML pages. WebGL, as is obvious from the name, uses a compatible web browser to run its features. The JavaScript code is added in the `<script>` tag of an HTML file.

# Chapter 6

## Conclusion

In this project, we analyse the concept and features of WebGL. The main advantage in using WebGL is that it is cross-platform, cross-browser, royalty-free API used to create 3D graphics in a Web browser. Three.js allows the creation of GPU-accelerated 3D animations using the JavaScript language as part of a website without relying on proprietary browser plugins.

Implementing WebGL on our webpage is made easier if we use the three.js library. The three.js library is used extensively in our project to use the WebGL renderer and create three-dimensional objects. The WebGL technology helps in making the games and webpages designed to be more immersive and less server intensive.

When we tested our game in different browsers we could infer that different browsers have different speed of execution of the code. We could also infer that some of the old browsers does not support the game but latest browsers like Google Chrome, Opera Browser and Firefox browser supports the game. If the system does not have a good graphics card, the WebGL game will not run as the browser uses the GPU of the system to render the scene, and hence shows a blank screen. Our team has got a wonderful experience by working on this project. We have advanced through various stages of software development as well as some new technologies and programming concepts.



## REFERENCES

- [1] Three.js <http://threejs.org/>
- [2] "Mr doob", <http://github.com/mrdoob/three.js>
- [3] Tony Parisi, *WebGL:Up and Running*, O'Reilly Media Publication, 2012
- [4] JavaScript <http://www.w3schools.com/js/>
- [5] Stackoverflow, <http://stackoverflow.com/questions/10041723/how-to-embed-audio-within-a-website>

# Appendix A

## Source Code

### A.1 Adding the wooden crate

```
var geometry = new THREE.BoxGeometry( 200, 200, 200 );
var texture = THREE.ImageUtils.loadTexture( 'textures/crate.gif' );
texture.anisotropy = renderer.getMaxAnisotropy();
for( var i=0; i<8; i++)
{
    material[i] = new THREE.MeshBasicMaterial( { map: texture } );
    crates[i] = new THREE.Mesh( geometry, material[i] );
    scene.add( crates[i] );
}
```

### A.2 Introducing Mouse Click

```
function onDocumentMouseDown( event ) {
    event.preventDefault();
    var k, flag = 0;
    mouse.x = ( event.clientX / renderer.domElement.width ) * 2 - 1;
    mouse.y = - ( event.clientY / renderer.domElement.height ) * 2 + 1;
    raycaster.setFromCamera( mouse, camera );
    var intersects = raycaster.intersectObjects( crates );
    for ( var i = 0; i < intersects.length; i++ )
    {
        SELECTED = intersects[i].object;
        for( var j=0; j<crates.length; j++)
        {
```

```
if (SELECTED.position.x == crates[j].position.x)
if (SELECTED.position.z == crates[j].position.z)
thisObject = j;
}
score();
var audio = new Audio('sounds/cling.wav');
audio.play();
if (objects.length < 2)
{
objects.push(crates[thisObject]);
if (objects.length == 2)
{
item1 = objects.pop();
item2 = objects.pop();
if ((item1.position.x == item2.position.x) && (item1.position.z == item2.position.z))
{
item1.material.color.setHex(defaultColor);
item2.material.color.setHex(defaultColor);
}
else
{
color1 = item1.material.color.getHex();
color2 = item2.material.color.getHex();
if (color1 == color2)
{
setTimeout(function() {
scene.remove(item1);
scene.remove(item2);
count -= 2;
if (count == 0) {
if (click > 18) {
var audio = new Audio('sounds/boo.wav');
audio.play();
alert("Congrats!\nYou have a terrible memory!\nYour SCORE is " + click);
}
}
```

```
else {
var audio = new Audio('sounds/applause.wav');
audio.play();
alert("Congrats!\nYou won!\nAnd Your SCORE is" +click);
}}},1000);
}
else
{
setTimeout(function(){
item1.material.color.setHex(defaultColor);
item2.material.color.setHex(defaultColor);
},1000);
}}}}
render();}
```

### A.3 Implementing random colors

```
while(randCol.length!=8)
{
rand = Math.floor((Math.random() * 8) + 1);
if (!(include(randIndex,rand)))
{
randCol.push(randObj[rand]);
randIndex.push(rand);
}
else
{
continue;
}}
```

### A.4 Keeping Score

```
function score() {
text2 = document.createElement('div');
text2.style.position = 'absolute';
text2.style.width = "100px";
text2.style.height = "30px";
```

```
text2.style.font = "italic bold 20px CLucida Console, Monaco, monospace";
text2.style.backgroundColor = "#D7D7D5";
text2.style.borderRadius= "10px 20px 10px 20px"
text2.style.color="#060604";
text2.style.opacity="0.9";
text2.style.borderWidth = "medium";
text2.style.borderStyle = "outset ";
text2.style.borderColor = "#060604";
text2.innerHTML = "SCORE- "+ ++click;
text2.style.top = 50 + 'px';
text2.style.right = 100 + 'px';
text2.style.padding = "10px 10px 10px 10px";
text2.style.textShadow = "1px 1px 1px grey";
text2.style.textAlign = "center ";
document.body.appendChild(text2);
}
```

# Appendix B

## Screenshots

### B.1 Crates - A 3D WebGL game

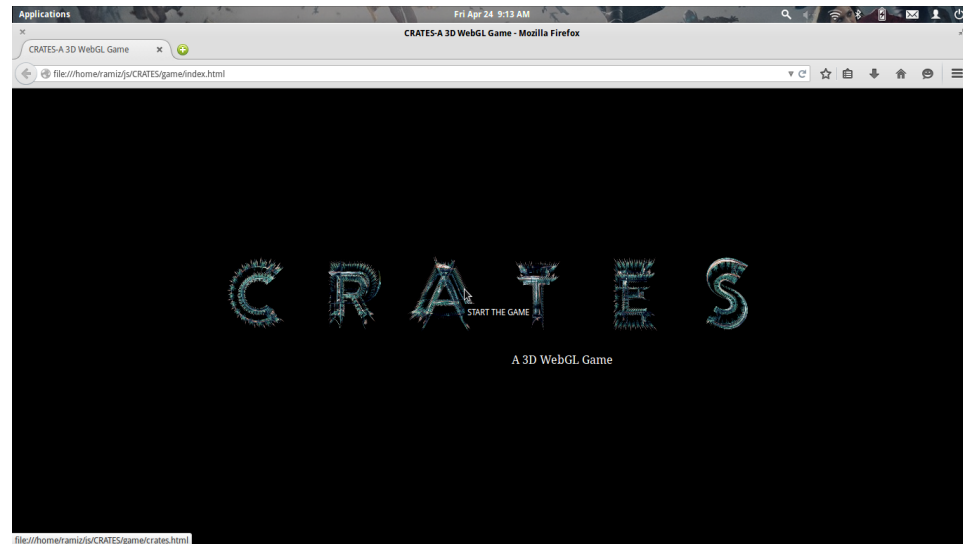


Figure B.1. Crates - Welcome Screen

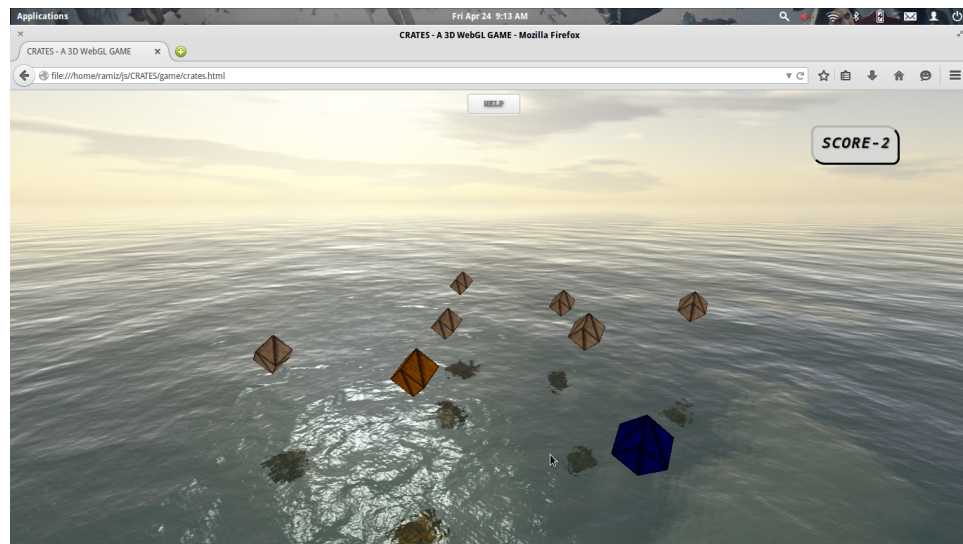


Figure B.2. The game in play