3D game development using BABYLON.js

Raja Naseer Ahmed Khan

B.Sc.(Hons) in Software Development

May 9, 2020

Final Year Project

Advised by: Dr Brian McGinley
Department of Computer Science and Applied Physics
Galway-Mayo Institute of Technology (GMIT)

Contents

1 Introduction **6**
- 1.1 Parts of the Project . . . 6
  - 1.1.1 Dissertation . . . 6
  - 1.1.2 Project Code . . . 7
  - 1.1.3 Summary of Each Chapter . . . 7
  - 1.1.4 Requirements Specification in the Project . . . 8

2 Context **10**

3 Methodology **12**
- 3.1 Agile Development Approach . . . 12
  - 3.1.1 Principles of Agile Model . . . 14
  - 3.1.2 Advantage of Agile Model . . . 14
  - 3.1.3 Disadvantages of Agile Model . . . 15
- 3.2 Version Control . . . 15

4 Technology Review **16**
- 4.1 Frontend Development Layer . . . 16
  - 4.1.1 WebGL . . . 16
  - 4.1.2 Introduction to Babylon.js 3D engine . . . 16
  - 4.1.3 Tools of Babylon.js . . . 18
  - 4.1.4 Babylon.js vs Three.js . . . 20
- 4.2 Backend Layer . . . 20
  - 4.2.1 Node Express vs HAPI . . . 20
  - 4.2.2 Socket.io vs WebSocket . . . 24
  - 4.2.3 Java vs JavaScript . . . 26
  - 4.2.4 What is Wamp . . . 27
  - 4.2.5 Heroku on Git . . . 28
- 4.3 Databases Layer/Tier . . . 29
  - 4.3.1 My-SQL Vs MongoDB . . . 29
  - 4.3.2 My-SQL . . . 29

4.3.3 MongoDB 30

5 System Design **34**
5.1 Python Server of Initial Development 34
5.2 Using Wamp for 2nd Stage Development 35
5.3 Contents of Game Folder 35
5.3.1 Tank Fight.php 35
5.3.2 main.js file 35
5.3.3 index.php 35
5.3.4 db.php 35
5.3.5 login.php 36
5.3.6 logout.php 36
5.3.7 registration.php 36
5.3.8 dashboard.php 36
5.3.9 auth session.php 36
5.3.10 multiplayer Folder 36
5.3.11 Tank Fight Folder 36
5.3.12 Dude Folder 36
5.3.13 Sound Folder 36
5.3.14 images Folder 37
5.3.15 css Folder 37
5.4 Using Visual Studio Code 37
5.5 Making of index.html 37
5.6 Code Logic Page main.js 37
5.6.1 First Scene With Camera and Light and 3D Box 38
5.7 Creating Ground/Terrain from HeightMap Graphics File . . . 39
5.8 Assigning Movement and Fire/Shoot Projectiles from the Object (Event Listeners) 40
5.9 Movement of car using front vector 41
5.10 Destroying the projectile(Bullets or cannon balls) after certain time 42
5.11 Importing the animating pre built characters from Babylon (also called Meshes) 42
5.12 Collision Detection and Imaginary Boxes 43
5.13 Creating the cameras in different prospective 44
5.14 The Assets Manager 44
5.15 Using sound 45
5.16 Using particle system 46
5.17 Using Socket.io for Multiplayer 47
5.18 Deploying it to Heroku server 48

6 System Evaluation **50**
6.1 Testing 50
6.2 Build on Prototype 50
6.3 Testing Stages 50
6.3.1 First Stage 50
6.3.2 Second Stage 50
6.4 Unit Testing 51
6.5 Performance 51
6.6 Evaluation of Objectives 51
6.7 Limitation, Issues and Improvements 52
6.7.1 Server 52
6.7.2 Client 52

7 Conclusion **53**

8 Appendices **54**

About this project

Abstract JavaScript is very popular in web developers, like every programming language JavaScript has pros and cons although it is easy to use language, runs on any device that supports web browser, it is not the fastest language performance wise, it is not compiled hence, hard to read and debug code, vulnerable security wise, different browser may understand different syntax of JavaScript. In this project, we will be experimenting to make a game using JavaScript engine called Babylon. Which is based on pure JavaScript syntax and may not be easy task to achieve as compare to other frameworks and game development engine like Unity3D available. For the sake of learning and experimenting, we would go through step by step process to develop a game which will produce a multi-player mode using Node server, and finally we will conclude our experiences with comparisons to other available JavaScript libraries for example. Three.js etc

Authors I am Raja Naseer Ahmed Khan student at Galway Mayo Institute of Technology (GMIT), Dublin Road Campus, Galway Republic of Ireland. I am developing this project for my final (4th) year of B.Sc. (Hons) in software development.

Chapter 1

Introduction

In this module for Applied project with minor dissertation we are assigned to demonstrate our understanding of years of study of Bachelor of Science in Software Development. We progressed through the initial stage of learning basic of computer programming to complete software system development using the modern technologies and techniques. We have grown our understanding on how the system is built and progressed using modern tools provided to monitor, collaborate and develop systematically.

In this project we will use the said Babylon JavaScript engine and we will make a 3D game which can be run on any browser(multi-platform). This game will be using mouse and keyboard keys to move characters around the ground and shoot cannon, laser, and bullets projectile to destroy other meshes(players,enemies). We will also use the socket.io and node express server to run and create multiplayer mode to instantiate multiple player objects. Also, this project will be stored on Heroku.com for hosting online and we will get an online link of cloud storage at Heroku. The main purpose and our goal in this project will be to enhance our understanding to develop complete system to production by using any suitable methodology of system development life cycle (S.D.L.C) i.e. agile, waterfall models.

1.1 Parts of the Project

1.1.1 Dissertation

In the dissertation writing section, we will research the present past and the future scope of our project, we will also document all of the project functionality in details in order for reader to understand, how we have approached

the programming design and development of this project. We will introduce our concept of the game project. We will also provide detailed understanding of different technologies and development models which are being used in modern development environment today.

1.1.2 Project Code

In the project coding section, we will demonstrate how to build a game which will be monitored by the assigned supervisor of our software department from G.M.I.T on weekly basis. In the weekly meeting with our supervisor we will show our weekly sprints of the project also discuss the user's stories which we will use for following week to be implemented in project. In each sprint we will provide working copy of the project. And by doing incremental approach we will complete our final product. Hence, this approach will be agile system development life cycle. We will produce a simple yet, efficient, and easy to use application. We will also gain understanding of problems arise.

1.1.3 Summary of Each Chapter

- Context
  In this chapter we will investigate and research how 3D game development has expanded to play a significant role in our daily life. We will research a wide variety of topics relating to game development in JavaScript 3D engines. Starting with the early development of JavaScript, giving us base to develop all online content including the websites, apps, and games. we will discuss the pro and cons of using JavaScript and the modern game development environment.

- Methodology
  In this chapter we will investigate the way we approached our plan and we organized and develop our project. We will talk about the different methodologies that are adapted and used to achieve the research and building of our project, also we will discuss why we have implemented the said approach of methodology. This chapter will give a view to reader that, how was system was developed from start to finish.

- Technology Review
  In this chapter, we'll cover the technical side of our project, looking back on technologies that made up the ultimate revision of the project. we'll explain the various technologies, we added and the way they were implemented throughout the project. we'll look over the online stack

we used and therefore the technologies we added to form a more robust and useful web-based 3D game. we'll re-evaluate why we used the given technologies and therefore the benefits we saw in them over others.

- System Design
  This chapter will focus on architecture and design of our web-based game Car Shoot. We will show code snippets to make reader understand what code was doing. Also, we will provide visual representation in form of diagrams, where suited. It will discuss the technologies involved in building the system from start to finish. We will also demonstrate our basic mathematical formulas used for the game to function properly.

- System Evaluation
  In this section we will assess the game project we have developed. we will check health and fitness of our game, also the testing and the system is expansible, adaptable,flexible and modular. we will assess the and evaluate the expansion plans and we will also address where things can be improved.

- Conclusion
  In this chapter, we will discuss the outcome of our project. We will discuss was the project successful or failure. What issues rise during the project. What were the final thoughts on completion of our project? Would we recommend the said project for further research or to be build on the technologies used?

1.1.4 Requirements Specification in the Project

- Must be able to move the object around the terrain(ground) and shoot projectile of few different types that is cannon and laser beam towards our enemies.

- Must be able to kill the enemy in few shoots (health assigned to enemy will decreases eventually)

- Each shoot should play sound of shooting.

- Person who gets hit by the projectiles (bullets or balls) should release blood particles.

- Can have multiple levels using teleportation portal to bring to another scene or world.

- Can be played by the multiple players at same time using any web socket technology.
- Uploaded game to Heroku server is working as expected and link is accessible by outer world.

Chapter 2

Context

JavaScript has always been one of the popular languages in developers. JavaScript started to revolutionize the internet what it is today. Since then, we have achieved to build the frameworks for JavaScript, we have made progress from the pure JavaScript complicated code to build frameworks, which are solving the complexities of making code naturally understandable. We have also developed frameworks to make JavaScript graphics and games attainable. In this research I have tried to make the pure JavaScript understandable for the coding students. so far, I haven't found any basic research on JavaScript for the students done in much detailed structure. I have researched that JavaScript may not be easy to understand, but if we can push coders to learn this, this might be very influential in their career. They would get deep understanding of all those frameworks, which are popular in web and game developers nowadays. If students are able to understand the basic of JavaScript, they will be to become real champion of coding and debugging of web, apps and game development. JavaScript is backbone of many popular frameworks. The subset of JavaScript is Typescript and with Script we have made Angular, React and lots of other popular frameworks. JavaScript has support in many game development engines as well. The unity is on of the very popular game development Integrated Development Environment (IDE) and it has support for JavaScript along the C#.

In this research we will try to use Babylon.js JavaScript engine to learn how to create the objects and animate them. we will use functions and classes to define the shape, size and animation of each object created on the screen. once, we have created the single player version of the game, we will try to use socket.io to run the game in multiplayer by creating instance of the game object on different sockets. Sockets are the part of web sockets technology and it is use to communicate between the computers over internet or local network.

so, far i have seen that in the degree course work for any computer science, pure JavaScript has been neglected and main focus is always on the framework.

- Provide a context for your project.
- Set out the objectives of the project
- Briefly list each chapter / section and provide a 1-2 line description of what each section contains.
- List the resource URL (GitHub address) for the project and provide a brief list of the main elements at the URL.

Chapter 3

Methodology

Right now, we will investigate the methodologies which are followed to design, compose, oversee and build up the venture. We will talk about the techniques that were adjusted and joined to finish the innovative work of the undertaking alongside why they were actualized. This segment means to offer knowledge to the peruse how the undertaking changed from research to definite programming while at the same time teaming up with supervisor.

Although my four years of studying at G.M.I.T a significant accentuation was constantly set on the significance of programming advancement strategies and the significance of picking the most reasonable procedure for a give venture. There are various procedures that have all been broadly investigated, for example, Cascade, RAD (Fast Application Improvement), Waterfall Programming and Pair programming to give some examples. For this undertaking, I have picked the fundamental present-day pioneer likewise dependent on the systems utilized by our most of the organization, a methodology known Agile or Incremental Development programming.

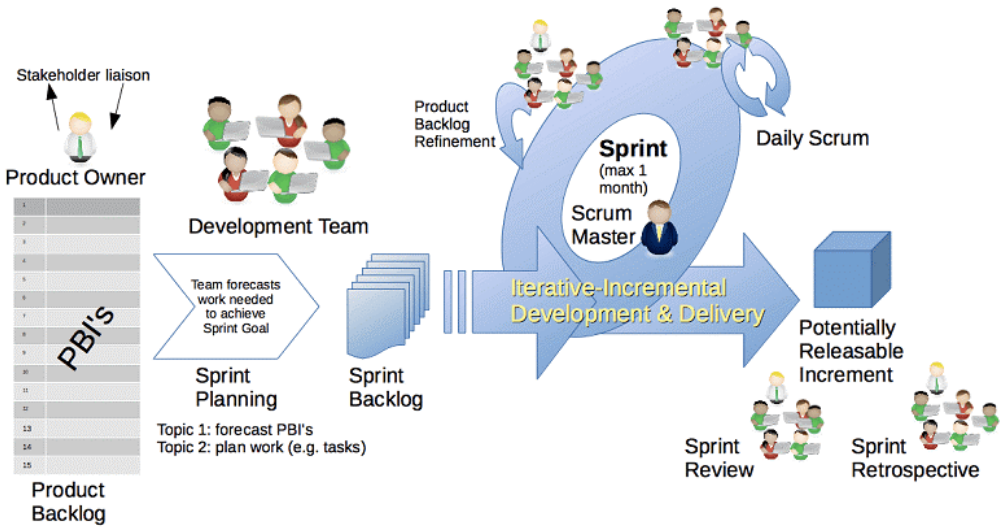
3.1 Agile Development Approach

The Agile model which is a combination of iterative and incremental models means that it is made up of iterative and incremental models. Process adaptability and customer satisfaction are considered in Agile model.

In earlier times iterative waterfall model was used to create software. But in today's time, developers have to face many problems. The biggest problem is that in the middle of software development, the customer asks to make changes in the software. It takes a lot of time and money to make these

changes. So to meet all these deficiencies, agile model was proposed in the 1990s.

The agile model was primarily designed to make changes between software development so that software projects can be completed quickly.



Our agile model is consists of the following steps:

- Requirements Gathering
- Requirements Analysis
- Design
- Coding
- Unit Testing
- Acceptance Testing

In our agile model the software product is divided into small incremental parts. In this, the smallest part is developed first and then bigger than that. And each incremental part is developed on iteration. Each iteration is kept short so that it can be easily managed. And it can be completed in two to three weeks. Only one iteration is planned, developed, and deployed at a time.

3.1.1 Principles of Agile Model

- There is a customer representative in the development team to maintain contact with the customer at the time of software development and to understand the requirement. When an iteration is completed, stakeholders and customer representatives review it and evaluate the requirements again.
- A working software demo is given to understand the customer's requirements. That is, it does not depend only on documentation.
- The incremental versions of the software have to deliver to the customer representative after a few weeks.
- In this model it is recommended that the size of the development team should be small (5 to 9 people) so that the team member who is able to communicate face to face.
- The agile model focuses on the need to quickly complete any changes in the soft-ware.
- In agile development, two programmers work together. One programmer does the coding, the other reviews that code. Both programmers keep changing their functions, that is, sometimes someone coding, sometimes a review.

3.1.2 Advantage of Agile Model

- In this, two programmers work together so that the coding is very good and there are very few mistakes in it.
- In this, the software project is completed in a very short time.
- In this, the customer representative has the idea of every iteration so that he can easily change the requirement.
- This is a very real approach to software development.
- It focuses on teamwork.
- There are very few rules in this and the documentation is also negligible.
- It does not require planning.
- It can be easily managed.
- It provides flexibility to developers.

3.1.3 Disadvantages of Agile Model

Its disadvantages are as follows:

- It cannot handle complex dependencies.
- There is confusion in development due to lack of formal documentation in it.
- It depends mostly on the customer representative, if the customer representative gives any wrong information then the software can become wrong.
- Only experienced programmers can take a decision in this. New programmers can-not take any decision.
- In the beginning of software development, the amount of effort and time it will take to make the software is not known.

3.2 Version Control

GitHub is a type of software company. If you are learning Web Development, Soft-ware Development, Ethical Hacking, Coding Language or working in this field, then GitHub.com will help you a lot, on this you will get only those people who are Web Development , Software Development, Hacking, Coding or working in GitHub, you can take help from them, take their codes and customize them and use them in your project.

Most important part is that you can collaborate the development of your project through GitHub. I created my repository which means that we will have software copy on cloud. All the times you change you upload it to your cloud repository. Like this your project was safe. You could revert back changes in case of misfortune with project.

Also, you can create branches on your project and keep working on separate branch. This technology helped me a lot during the development as I made several mistakes and my software was broken many times. I just reverted back to early stage of changes using git.

- What about validation and testing? Junit or some other framework.
- Selection criteria for algorithms, languages, platforms and technologies.

Chapter 4

Technology Review

In this section we will introduce you with technologies used in our project and will talk about our technical aspect of the project mostly. We will also look into the technologies which can be replaced by our used stack of technology and will compare the advantages and disadvantages in details.

4.1 Frontend Development Layer

Although our project is build entirely on backend as a website and will only server static index.html to client. still we can consider the graphics development in frontend tier.

4.1.1 WebGL

[1] The JavaScript API (Application Programming Interface) and WebGL is a JavaScript engine widely used to render interactive 3D and 2D graphics. WebGL is available by default in all major browsers, but the performance is higher in Firefox and Chrome, since we are using a newer version called WebGL 2, which is a major update of the API. When creating web content with massive structures of geometry in 3D, one should use WebGL since it makes use of the GPU rather than the CPU. This is a good thing since GPUs have a much higher performance when it comes to rendering, and visualization. [3]

4.1.2 Introduction to Babylon.js 3D engine

BabylonJS is an open source JavaScript framework for building 3D games and web graphics with HTML5 and WebGL.

BabylonJS is developed by Microsoft employees in the year 2016. David Catuhe, a principal program manager for the Window and devices group at Microsoft is the main person behind developing BabylonJS and making it a big success.

BabylonJS can run on any browser which supports WebGL that is Internet Explorer 11+, Firefox 4+, Google Chrome 9+, Opera 15+, etc. does have WebGL support. Babylon can be downloaded or referenced in the script source section of HTML page.

Babylon supports many 3rd party software for development of 3D animations and graphics that is Blender, 3DMax. Lots of support available on www.BabylonJS.com.[2]

BabylonJS has the following type of 3D scenes supported for the development of 3D game and in web graphics:

- Draw box, sphere, cylinder, cone, height map ground, animated characters.
- Scene can have Cameras, multiple view cameras and Lights.
- Objects can have Meshes, textures and Materials.
- Supports Sprites assets to develop animated meshes and sound.
- Supports Mesh Intersection and collision detection.
- Physics engine plug-in to be used for easy access to real life physics in object.
- Supports Assets/Action Manager to fast load of assets and scenes.
- Supports Solid Particles.
- Support for Bones and Skeletons for animation.
- Adding music and sound to the scene.

Sample Code to draw sphere on the canvas[3]

```
<!-- Following should be in the enclosed in script tag of html page-->

    var canvas = document.getElementById('canvas');
    var engine = new BABYLON.Engine(canvas, true);
    var createScene = function(){
    var scene = new BABYLON.Scene(engine);

    scene.clearColor = new BABYLON.Color3(1, 0.8, 0.8);

    var camera = new BABYLON.ArcRotateCamera('Camera', 2, 0.8, 10,
        new BABYLON.Vector3(0, 0, 0), scene);

    scene.activeCamera.attachControl(canvas);

    var light = new BABYLON.PointLight('li', new BABYLON.Vector3(0, 0, 10),
        scene);

    var sphere = BABYLON.Mesh.CreateSphere('sphere', 10, 1.0,
        scene);

    return scene;
};

    var scene = createScene();
    engine.runRengerLoop(function(){
        scene.render();
});
```

4.1.3 Tools of Babylon.js

Sandbox

The Babylon.js provides a convenient way to view 3D models and convert them to glTF. For example, the Flight Helmet was created using the Maya exporter. First, navigate to sandbox.BabylonJS.com. Next, drag and drop your 3D file (and associated textures) into the sandbox. Your model should then load and appear in the viewport. Afterwards, you can launch the inspector, go to the glTF tab, and export as glb, the binary format of glTF. Now your model can be imported as a glb asset.

Playground

The place to try out coding with Babylon.js. Experimenting and changing any code in the playground and clicking on the Run button will not affect any original code in the playground you currently using. Original code can be restored by refreshing the browser. You can write the code in JavaScript or Typescript. The playground software compiles the code to JavaScript, in the background, before rendering

Spector

Spector is a tool aim to WebGL developers willing to know what happens on their context. It enables capturing all the available information from a frame. You will be empowered to look at your entire commands list with their associated visual states and context information.

You will be able to Explore and Troubleshoot your WebGL and WebGL2 scenes easily. This is compatible with any WebGL capable browser and all the web based 3d engines as well as vanilla scripts

Exporters

Babylon.js has exporters available for 3ds Max, Maya, and Blender. The 3ds Max exporter supports 3ds Max 2015 and higher, and exports to glTF as well as to the Babylon.js file format. The Maya exporter supports 2017 and higher, and exports to glTF as well as to the Babylon.js file format. The Blender exporter supports Blender 2.79 and only exports to the Babylon.js file format. However, the Khronos glTF-Blender-Exporter can be used to generate glTF files from Blender so they can be imported into Babylon.js.[4]

If an exporter is not available from Babylon.js but it supports exporting to glTF or the obj file format, then they can also be imported into Babylon.js! [1]

Node Material Editor

The Node Material is a new material introduced with Babylon.js v4.1. It lets you create a material based on custom shaders but without having to deal with shader code. All the shader creation will be done using either an UI (the Node Material Editor) or by creating and connecting nodes (the Node Material blocks).

4.1.4 Babylon.js vs Three.js

Both Three.js and Babylon.js are easy to use libraries for handling of graphics using WebGL animations.

Three.js was created with one goal in mind: to take advantage of web based renderers for creating GPU enhanced 3D graphics and animations. As such, this framework employs a very broad approach to web graphics without focusing on any single animation niche.

This flexible design makes Three.js a great tool for general purpose web animations like logos or modelling applications (great examples can be found here).

Where Three.js attempts to bring a wide range of animation features to the WebGL table, Babylon.js takes a more targeted approach. Originally designed as a Silver light game engine, Babylon.js maintains its penchant for web-based game development with features like collision detection and anti-aliasing. As previously stated, Babylon.js is still fully capable of general web graphics and animations as evidenced by the demos found on the front page of its website.

In the end, these two relatively young frameworks enable web developers to more easily take advantage of the powerful 3D opportunities afforded by WebGL. As such, anyone with an interest in 3D web development should certainly take a closer look at this cutting-edge technology.[5]

4.2 Backend Layer

We will discuss here some of the most popular backend development server technologies for the web development.

4.2.1 Node Express vs HAPI

NODE

[6] Node.js is an Open Source Server-Side Run-time Environment built on Chromes V8 JavaScript Engine. Node.js provides an Event Driven, Non-Blocking Input Output and Cross-platform Run-time Environment to create a highly Scalable Server-Side Application using JavaScript. Node.js are used to create various types of applications such as Command Line Application,

Web Application, Real Time Chat Application, REST API Server etc.

Simple Node Express Server

```
// Import Node.js core module
Var http = require('http');

//create web server
var server = http.createServer(function (req, res) {

//check the URL of the current request
        if (req.url == '/') {

// set response header
        res.writeHead(200, { 'Content-Type': 'text/html' });

// set response content
        res.write('<html><body><p>This is home Page.</p></body></html>');
        res.end();

}

else if (req.url == '/student') {
        res.writeHead(200, { 'Content-Type': 'text/html' });
        res.write('<html><body><p>This is student Page.</p></body></html>');
        res.end();
}

else if (req.url == '/admin') {
    res.writeHead(200, { 'Content-Type': 'text/html' });
    res.write('<html><body><p>This is admin Page.</p></body></html>');
        res.end();
}

else
        res.end('Invalid Request!');
});

server.listen(5000); //6 - listen for any incoming re-quests
```

```
console.log('Node.js web server Running port 5000');
```

Node Parts Explained

- Node.js REPL Terminal

REPL stands for Read Eval Print Loop and represents a computer environment such as a Computer Console or Unix / Linux Shell where a command is entered and the system responds with Output in an Interactive Mode.

- Node.js Module

Module in Node.js is a simple or complex performance that is organized in one or more JavaScript files that can be used in the entire Node.js application. Each Module in Node.js has its own Context so it cannot interfere with the other Module or pollute the Global Scope Also each Module can be placed in a separate .js file under a different folder .

Node.js implements the CommonJS modules standard. CommonJS is a group of Volunteers that defines the JavaScript Standards for Web Server Desktop and Console Application.

- Node.js Web Server

To access the web pages of any web application you need a web server. The Web Server handles all HTTPS requests for Web Application. For example, IIS is a Web Server for ASP.NET Web Applications and Apache is a Web Server for PHP or Java Web Applications.

- Node.js File System

Node.js File System Module as a File Server allows you to work with the file system on your computer. Uses the Require () method to include the File System Module. The File System uses the following types of Module Read Files, Create Files, Update Files, delete files, Rename files.

- Node.js Debugging

You can Debug Node.js Application using various tools. Such as Core Node.js Debugger, Node Inspector, Built-in Debugger IDEs etc. Node.js provides a Built-in Non-graphic Debugging Tool that can be used on all plat-forms. It provides various Commands to Debug Node.js Application.

- Node.js Frameworks

In the Node.js Web Server section, writing a very low level of code is required to create a Web Application using Node.js. There are many Third Party Open Source Frameworks available in Node Package Manager which makes Node.js Application Development faster and easier. You can choose an Appropriate Framework according to your Application Requirements.

- Node.js Data Access

Node.js supports all types of databases no matter whether it is a Relational Database or NoSQL Database. However, NoSQL databases like MongoDb are the best fit with Node.js. To access Databases from Node.js you first need to install Drivers for the Databases you want to use.

HAPI

Hapi.js is an open source framework for web applications, it is used to build the for JSON API, application programming interface, API servers, websites, and HTTP proxy applications. HAPI is robust and deals with all aspects of REST APIs also very easy to build and maintain, it can be integrated to any front end development platform for ease of building SPA(Single Page Applications) applications.

Code Example

```
//import hapi package
Const hapi = require('hapi');

//create server
Const server = new hapi.Server();

//connect to localhost port 3000
Server.connection({
        Host: 'localhost',
        Port: '3000'
});

//server start
Server.start(err => {
        If(err) {
                Throw err;
```

```
        }
        Console.log(‘Server Running at PORT ${serv-er.info.port}’);
});
```

4.2.2 Socket.io vs WebSocket

WebSocket

It is the communication Protocol which provides bidirectional communication between the Client and the Server over a T.C.P connection, WebSocket remains open all the time so they allow the real-time data transfer. When clients trigger the request to the Server it does not close the connection on receiving the response, it rather persists and waits for Client or server to terminate the request.

Features of WebSocket are:

- WebSocket helps in real-time communication between the Client and the web server.
- This protocol helps in transforming to cross-platform in a real time world between the server and the client.
- This also enables the business around the world for real-time web application to enhance and to increase the feasibility.
- The major advantage it stands over an HTTP connection that it provides full duplex communication.

.

Why do we need WebSocket?

- It provides the full duplex communication which helps in persisting the connection established between the Client and the Web Server.
- It also lives up to the standards and provides the accuracy and efficiency stream events to and from with negligible latency.
- WebSocket removes the overhead and reduce complexity.
- It makes real-time communication effortless and efficient.

Socket.IO

It is a library which enables real-time and full duplex communication between the Client and the Web servers. It uses the WebSocket protocol to provide the interface. Generally, it is divided into two parts, both WebSocket vs Socket.io are event-driven libraries.

- Client Side: it is the library that runs inside the browser.
- Server Side: It is the library for Node.js

.

Key features of Socket.IO

- It helps in broadcasting to multiple sockets at a time and handles the connection transparently.
- It works on all platform, server or device ensuring the equality, reliability, and speed.
- It automatically upgrades the requirement to WebSocket if needed.
- It is a custom real-time transport protocol implementation on top of other protocols.
- It requires both libraries to be used Client side as well as a server-side library.
- IO works on work-based events. there are some reserved events which can be accessed using the Socket on server side like Connect, message, Dis-connect, Ping and Reconnect.
- There are some Client based reserved events like Connect, connect-error, connect-timeout and Reconnect etc.

Why do we need Socket.IO:

- I handle all the degradation of your technical alternatives to get full duplex communication in real time.
- It also handles the various support level and the inconsistencies from the browser.

4.2.3 Java vs JavaScript

JavaScript

[7] JavaScript is a client-side scripting language.
By Using it, you can add dynamic HTML to a web page. You can also change the existing content and modify the page style. It was originally developed to add dynamic and interactive elements to websites. It is an open and cross-platform that aims to provide a better experience for the user. Just like PHP and ASP, JavaScript code can be inserted anywhere within the HTML of a web page. Cookies can be obtained with the help of JavaScript and they can also be set. It is also used to ask questions to visitors and perform a massage show.

JavaScript is a powerful scripting language, which is used to create web pages and applications. You should learn it because today, as a web technology, about 95 percent of websites are using it. Almost every web developer says that in the coming time, JavaScript has the highest scope.

JavaScript is called JS in programming language. It is interpreted programming language interpreted with object-oriented capabilities. It is also called dynamic computer programming language. Its program is called Script. Using this, you can apply different types of things in web pages. Its scripts can be provided and executed as plain text.

Web development is one area where JavaScript is used the most. Because it is a Client scripting language, it is used to create web pages. Most websites use it for verification.

JavaScript is also used well in creating web applications. Initially many programming languages was used for the development of technology browser and personal computer. But JavaScript took it to the next level. Due to this, such applications were developed which hardly anyone would have thought about. Node JS is used to create Web Server. Node JS has many advantages. The servers built by it are very fast and do not use buffering. In addition, it is single threaded with event looping which is used in non-blocking manner. JavaScript is also used in creating mobile applications. The most important thing that only JavaScript can do is to create applications without web contexts. With this help, both Android and iOS applications can be made. This is some of the areas where JavaScript is used more. Apart from this, it is also used in making Games and Server applications. There are many other

things that can do this, but this is the important area.

JavaScript is one of the three main technologies of the World Wide Web (WWW), followed by HTML, CSS. All the main websites on the Internet use it. Because it enables interactive web page, that's why it is used a lot in creating web application. It is very lightweight due to which JavaScript is used as a major part of the web page nowadays.[8]

Java

Due to the name of JavaScript as java, people think that JavaScript is a part of Java platform, but Java is completely different from JavaScript. Java and JavaS-cript have the following differences: -

- JavaScript is an OOP scripting language while Java is an OOP programming language.
- JavaScript is run only in the browser while Java code runs in both the JVM and the browser.
- JavaScript code is not easy and complicated while Java code is complicated and difficult.
- JavaScript is used to create dynamic web pages with HTML while Java is used to create stand alone and live applications using applet.
- We can run it in the browser without compiled it while the Java code has to be compiled.
- JavaScript supports more platforms than Java.
- Java's objects are based on class while JavaScript is based on prototype.
- JavaScript is dynamic typed language while Java is static typed language.
- JavaScript is not a stand-alone language. It is used with html, while Java is a stand-alone language, it does not require any other language.

4.2.4 What is Wamp

This is a free software package that we use for web development on our computer, if you want to use CMS (Content Management System) like WordPress, Joomla, Zencart on your computer, then you have to take help of wampserver.

Because it is all based on PHP (Server Side Language), it reads the need of the server to run all of them. Apart from this, all these CMS keep their data in the database, so we also need the database.And in WampServer, we get all three PHP, Server (Apache2), Database (MySQL) only, apart from this, PHPMyAdmin also comes in this package so that you can manage your database very easily.

WampServer is made up of four words, let's know the meaning of these four words

- W-indows operating system You must know this well because most people in world use this OS only to run a computer, ie this package is only for windows operating system like LAMP for Linux OS and MAMP for Mac OS.
- A-pache Server It is used to run Local Web Server (localhost) on Windows OS, so that you can test your webpages on your computer's browser as you do on live server.
- M-ySQL This is a database management system in which we store all the data (post content, user profile, comment etc.) of our website.
- P-HP This is a server side scripting language that runs on Apache Server and with the help of this we extract the data from MySQL database and create a dynamically webpage.

4.2.5 Heroku on Git

Heroku is a free online cloud service and we will be using it to deploy our game online. Hence, it is important to describe the Heroku services. Heroku is type of Platform as a Service (PaaS) product supported by Amazon Web Services and is incredibly different from Elastic Compute Cloud. It's important to differentiate 'Infrastructure as a Service' and 'Platform as a Service' solutions as we consider deploying and supporting our application using these two solutions. Heroku is much simpler to use than Amazon Web Services Elastic Compute Cloud. Perhaps it's even too simple. But there is an honest reason for this simplicity. The Heroku platform give us ready run-time environment and application servers. Plus, we get pleasure from smooth joining with many development panels, already installed Operating System, and longer needed servers. That is why with Heroku, we do not have to consider infrastructure management, unlike with Amazon Web Services EC2. We only must choose a subscription plan and alter our plan when necessary. Heroku host web apps not the website as we have website kind of in browser game, we need

to trick Heroku think that this is a web-app, we can make one PHP index file and will refer to index.HTML from that php file. Using the tag <?php header('Location: index.html'); ?>

4.3 Databases Layer/Tier

4.3.1 My-SQL Vs MongoDB

we will discuss here these two very popular databases systems, which are being used by developers.

4.3.2 My-SQL

My-SQL is a fast, easy to use relational database. It is currently the most popular open-source database. It is commonly used in conjunction with PHP scripts to create powerful and dynamic server-side applications. My-SQL is used for many small and large businesses. It is developed, marketed and supported by a Swedish company My-SQL AB, it is written in C and C ++. Due to the popularity of My-SQL

My-SQL is becoming so popular for these reasons -

- My-SQL is an open-source database, so you don't have to pay a dime to use it.
- My-SQL is a very powerful program so it can handle a large set of functionality from the most expensive and powerful database packages.
- My-SQL is customizable as it is an open source database and open-source GPL licenses allow programmers to modify SQL software according to their specific environment.
- My-SQL uses the standard form of the well-known SQL data language.
- My-SQL supports many operating systems such as PHP, PERL, C, C ++, JAVA, etc.
- My-SQL is faster than other databases so it can also work well with large data sets.

- My-SQL is very compatible with PHP, which is the most popular language for web development.
- My-SQL supports large databases up to 50 million rows or more in a table. The default file size limit for the table is 4GB, but you can (if your operating system can handle it) increase it to a theoretical limit of 8 million terabytes (TB).

My-SQL Features

- Easy to use - My-SQL is easy to use. You only need to have basic knowledge of SQL. You can build and interact with My-SQL with just a few simple SQL statements.
- It is secure - My-SQL has a solid data protection layer that protects sensitive data from intruders. Passwords are encrypted in My-SQL.
- Client / Server Architecture - Follows the My-SQL client / server architecture. There is a database server (My-SQL) and arbitrarily many clients (application programs), which communicate with the server; That is, they query the data, save changes, etc.
- It is scalable - My-SQL can handle almost any data at most 50 million rows or more. The default file size limit is about 4 GB. However, you can increase this number to a theoretical limit of 8 TB of data.
- High Performance - My-SQL is faster, more reliable and cheaper because of its unique storage engine architecture.
- High Flexibility - My-SQL supports a large number of embedded applications which makes My-SQL very flexible.
- High Productivity - My-SQL uses triggers, stored procedures, and views that allow the developer to deliver a higher productivity.

4.3.3 MongoDB

Mongo DB is usually defined as a document-oriented database system. Which uses the concept of NoSQL, it provides automatic scaling as well as high availability, high performance. This open source product was developed by the company. MongoDB is available as a free database management tool under a Common Database License (GPL) as well as a commercial license as a manufacturer. MongoDB was intended to work with commodity servers.

Companies of various sizes across the world are using MongoDB as their database in all industries. A company called 10gen software started making it in 2007 and it was launched on February 11, 2009. For your information here, let us tell you that in 2013 this company changed its name to MongoDB Inc. kept.

In MongoDB, a database can be defined as a physical container for archiving or data. Here, on the file system, every database has its own collection of files. Typically, a MongoDB server consists of several databases. As we all know, collections can be defined as a group of MongoDB documents. Which exist in a single database. You can relate it to the table of the respective database management system. MongoDB collections do not implement the concept of schema. Documents that have a collection usually have different fields. Typically, all documents residing within a collection are for a comparable or related purpose.

It is essential that beginners need to know the purpose and requirement of MongoBD or its use, unlike SQL and other database systems. In simple words, it can be said that every modern day applications involve the concept of big data. Analysis of various forms of data, improved data handling features, deployment flexibility that are not capable enough to handle older database systems. Therefore, MongoDB is the next choice. Friends if you want to download and install mongoDB. So first you have to visit their website and download it. After that you can install it in your computer like other software. During its installation, you will be asked to download MongoDB Compass, you can install MongoDB Compass as well as MongoDB. With its help, you can use your database even without command line.

Why use MongoDB? Let you know, some basic requirements are supported by this NoSQL database, which is lacking in other database systems. These collective reasons make MongoDB popular in other database systems -

- Document-oriented data storage, i.e. data, is stored in a JSON style format that also enhances the readability of the data.
- Replication and High Availability of Data.
- MongoDB offers auto-sharking.
- Adhoc queries are supported by MongoDB, which helps to search for category queries, using fields or regex words.

- An index of values can be used to create and improve overall search performance in MongoDB. MongoDB allows any field to be indexed in a document.
- MongoDB has a rich collection of queries.
- Updates of data in MongoDB can be done at a faster speed.
- It can be integrated with other popular programming languages to handle structured as well as unstructured data within a wide variety of applications.

Benefits of using MongoDB

- It is easy to install, that is, install MongoDB.
- Since MongoDB is a schema-less database, there is no hassle of schema migration.
- Since it is a document-oriented language, document queries are used, which plays an important role to support dynamic queries.
- MongoDB easily scalable.
- Performance tuning is easier compared to other relational databases.
- This helps to provide faster access to data due to its nature of implementing internal memory for storing data.
- MongoDB is also used as a file system that can help in easy management of load balancing.
- MongoDB also supports the concept of regex (regular expression) as well as search using fields.
- Users can also run MongoDB as a Windows service.
- It does not require any VM to run on different platforms.
- It also supports speeding up data.

Features of MongoDB

- MongoDB is an open-source document-based database management tool that stores data in formats such as JSON. It is a highly scalable, flexible and distributed NoSQL database.

- As a NoSQL database, MongoDB removes the table-based structure of relational databases, known as BSON, to adapt documents such as JSON to dynamic schemas. This makes data integration faster and easier for some types of applications. MongoDB is built from scalability, high availability, and single server deployment to large and complex multi-site infrastructure.

- MongoDB Inc. was first introduced to MongoDB Inc. Developed by, then known as 10gen, in October 2007, originally a key part of the PaaS (Platform as a Service) product, similar to Windows Azure and Google App Engine. In 2009 the development was shifted to open source.

- Today, MongoDB has become one of the most popular NoSQL databases, being used as the backend of many major websites including eBay, Craigslist, SourceForge and The New York Times. MongoDB is available under the GNU Affero General Public License. While its language drivers are available under the Apache license, commercial licenses are also being provided there.

- Describe each of the technologies you used at a conceptual level. Standards, Database Model (e.g. MongoDB, CouchDB), XMl, WSDL, JSON, JAXP.

- Use references (IEEE format, e.g. [1]), Books, Papers, URLs (timestamp) – sources should be authoritative.

Chapter 5

System Design

In this section we will discuss the design, development and architecture of the game car war. In the following sections we will present the code snippets and visual dia-grams to help portray a basic understanding of the application design. The architecture is modeled on what know as Babylon.js 3D development engine. Babylon.js is a free open-source JavaScript graphics development library for building dynamics web-sites, games and supports most of the platforms known so far. The contents of this section will start with contents of the game and then followed by the in-depth knowledge of development stage and code to reach the final stage of our game.

The Basic design approach is to build a index.HTML page which will be served using any fast HTML server application for example

- Architecture, U.M.L etc. An overview of the different components of the system. Diagrams etc... Screen shots etc.

5.1 Python Server of Initial Development

For the initial development, I have used the python built in server as it was easy to start and run, used to listen on local port 8000, http://localhost:8000.

Following can be used on command prompt with python 2

```
python -m SimpleHTTPServer 8000
```

Following can be used on command prompt with python 3

```
python -m http.server
```

- Where 8000 stands for port to be used to run game on our local host.
- Also, we can use node express server and many other server libraries that can be installed using npm (node package manager)
- Also, we can use wamp server, but you need to reference to index.html within your index.php file to run it within the www folder of the wamp directory folder.
- in python 3 we don't need mention port, it will automatically starts on port 8000

5.2 Using Wamp for 2nd Stage Development

I have used the wamp server to build the MySQL database also to serve the PHP files of login, logout,registration, and index.

5.3 Contents of Game Folder

5.3.1 Tank Fight.php

The index.html file (it is home page of our game) page will be displaying canvas and will be connected to JavaScript using the Script Ref. also it has the style sheet CSS file connected for basic user interface designing. This is the entry point to our game which servers as home page.

5.3.2 main.js file

the Main.js fils is the JavaScript file where we will define our business logics, initializing the objects from Babylon.js libraries and connected to CSS, animation, sound and images.

5.3.3 index.php

This file bring us to log in page on visiting

5.3.4 db.php

We use this to connect to Mysql database with database name, and id password.

5.3.5 login.php

This file is a login form page

5.3.6 logout.php

This file bring us to logout page

5.3.7 registration.php

This is for Registration of user

5.3.8 dashboard.php

This file loaded once user logs in

5.3.9 auth session.php

This file tell us which page to load on website/game start, it will navigate you to login page

5.3.10 multiplayer Folder

Contains all the files and subfolder of multi player game with socket.io

5.3.11 Tank Fight Folder

Contains all the files for single player game with babylon

5.3.12 Dude Folder

The Dude folder contains the file and folders downloaded from github repository of Babylon.js. it is the animated character we will be using to walk around our terrain of ground and will serve as basic enemy to be shoot and chase our player.

5.3.13 Sound Folder

This folder contains the sound files which are needed to play games and attached to main.js. i.e. shooting sounds, dying sounds and in game sounds.

5.3.14 images Folder

This folder contains the images and graphics files which are attached to main.js i.e. the heightmap files for our ground, also the texture files for our objects.

5.3.15 css Folder

This folder Contains cascading styling sheet files which are attached to main.js and important for styling and functioning the game properly.

5.4 Using Visual Studio Code

Although any nice text editor can be used that is Notepad++, Visual Studio, Textpad, Notepad, but I have chosen to use Visual Studio Code because I am very well used to it for last few years. Visual Studio Code can be downloaded for free from https://code.visualstudio.com/.

5.5 Making of index.html

Main page usually is called index.html which contains the entry point to our, we include here source file which will contain our main logic, also we will make canvas here to keep the app secure because business logic will be in different file only reference of that file here. It is structured in very basic hypertext markup language(HTML) markup language can be seen and learned at w3 schools at https://www.w3schools.com/

5.6 Code Logic Page main.js

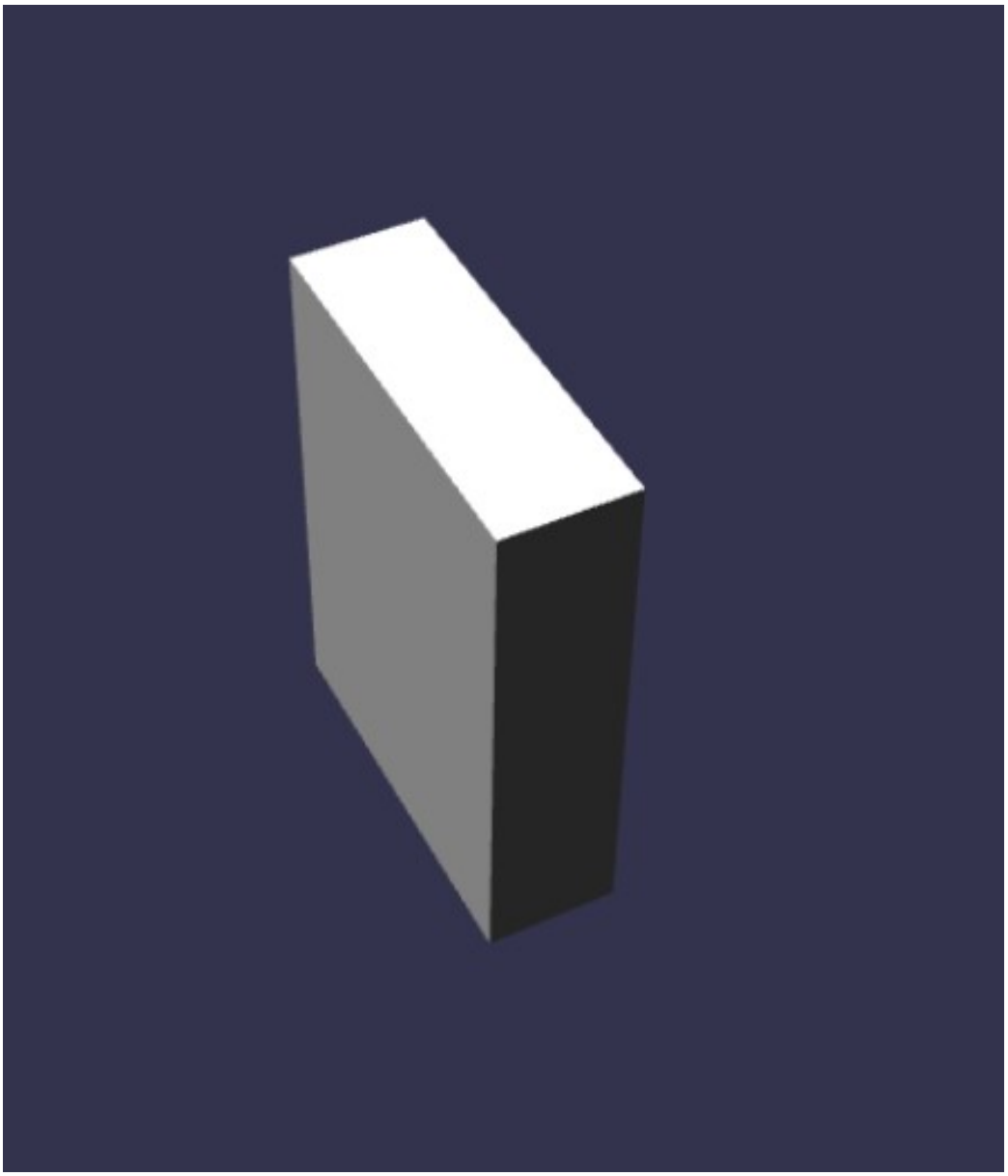
It contains all the functionality of our game, creation of objects and communication to server is done via this page. It's in the js folder and called main.js

For any 3D development in JavaScript or other game engine we must create a scene which is the container of the world, followed by camera to view the scene, and finally the lights to make thing visible. Just consider without lights our eyes we cannot see this beautiful world. Following are the code snippets to create full first scene.

5.6.1 First Scene With Camera and Light and 3D Box

```
var createScene = function () {

    // Create the scene space
    var scene = new BABYLON.Scene(engine);

    // Add a camera to the scene and attach it to the canvas
    var camera = new BABYLON.ArcRotateCamera('Camera', 3 * Math.PI / 4, Math.PI
        4, BABYLON.Vector3.Zero(), scene);
    camera.attachControl(canvas, true);

    // Add lights to the scene
    var light1 = new BABYLON.HemisphericLight('light1',
        new BABYLON.Vector3(1, 1, 0), scene);
    var light2 = new BABYLON.PointLight('light2',
        new BABYLON.Vector3(0, 1, -1), scene);

    // Add and manipulate meshes in the scene
    var box = BABYLON.MeshBuilder.CreateBox('box', {height: 1, width: 0.75,
        depth: 0.25}, scene);

    return scene;

};
```

Output image from Above code

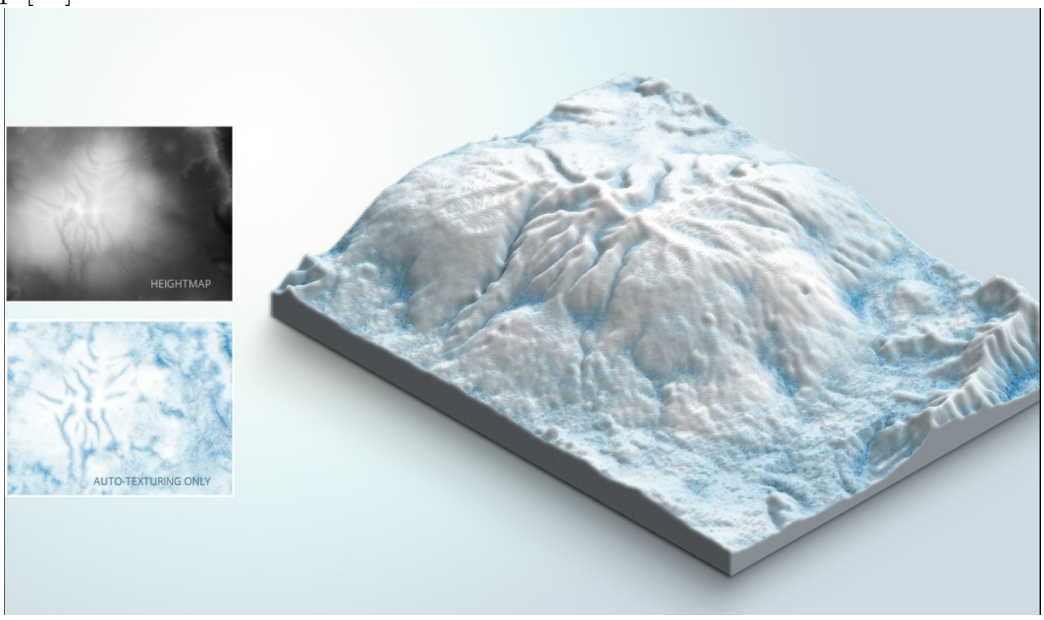


5.7 Creating Ground/Terrain from HeightMap Graphics File

The Heightmap files are one of the most cost-efficient and effective technologies for reconstructing 3D surface texture details.[9]

A heightmap can be considered a grayscale image where the brightest white

(closest to FFFFFF) is considered as the highest point of the map, and the darkest black (closest to 000000) is considered as the lowest point of the map.[10]



Code for Creating HeightMap[9]

```
//Code for creating Ground from heightmap //
function CreateGround(scene) {
    var ground = new BABYLON.Mesh.CreateGroundFromHeightMap('ground',
     'imag-es/hmap1.png', 2000, 2000, 20, 0, 1000, scene,
      false, OnGround-Created);
```

5.8 Assigning Movement and Fire/Shoot Projectiles from the Object (Event Listeners)

You can attach the EventListeners Method to the event handler with the element, And you can overwrite the event handler. In EventListeners you can add multiple event listener to an element. You can add many events in the same element such as 'Click' event, onchange event. Through the Event Listeners, you can control events very easily. Through this you can increase the readability of JavaScript and HTML. And you can understand the code easily. You can also control HTML code as well. You can also remove EventListeners easily. You can use removeEventListener () to remove EventListeners.

In this project we used keyup, and keydown to shoot cannon ball and laser Code was very simple in babylon.js to move the object forward we used w key when key is down it keep sending signal and object keep moving, when key up means that key is not being used and object stop moving.

Key down(key is pressed to move/function

```
document.addEventListener('keydown', function (event) {
    var scene = Game.scenes[Game.activeScene];
    if (event.key == 'w' || event.key == 'W') {
        isWPressed = true;
    }
});
```

Key up (key is back up or not being pressed to stop moving object

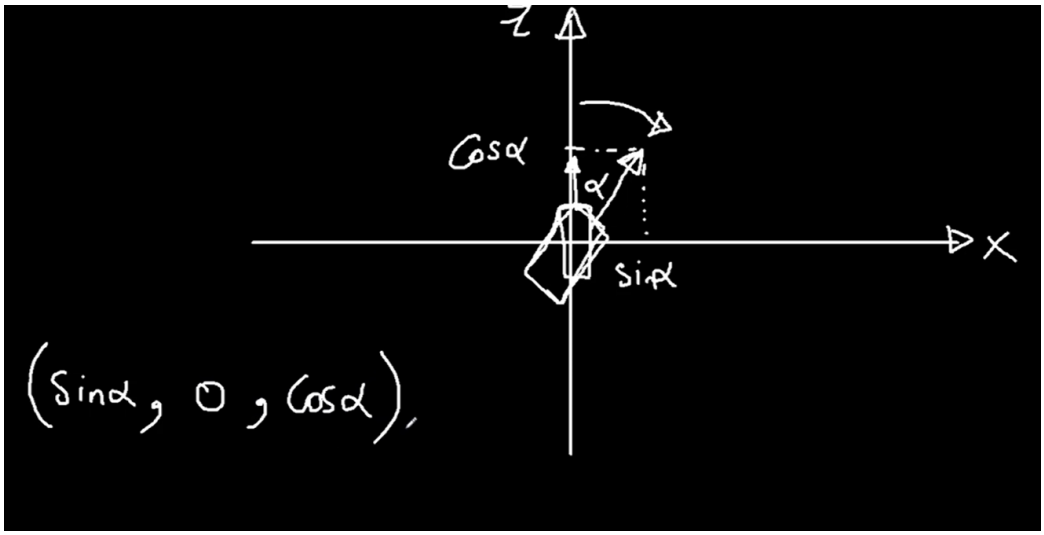
```
 document.addEventListener('keyup', function (event) {
    var scene = Game.scenes[Game.activeScene];
    if (event.key == 'w' || event.key == 'W') {
        isWPressed = false;
    }
});
```

Code for shooting Projectiles

```
tank.fireCannonBalls = function (scene) {
tank.fireLaserBeams = function (scene) {
// use the key listener within those above function
// which on key down will trigger projectile
```

5.9 Movement of car using front vector

By using only the keys and without any calculation, the object only move left right sideways not rotating front part of it like we rotate cars. for that reason we have to calculate the front of the object, which will be called front vector which will have x,y,z position of front of the object. on pressing left, right or a, d keys this front vector will be changed accordingly. We will use cos and sin functions to calculate the front vector position when keys is pressed each time. That will give us new position of front vector and object will rotate like it rotates vehicles.



When a is pressed a (rotation will be in negative direction or in right hand turning direction) we can use formula front vector = sin(obj.y), 0.z, cos(obj.y)

When d is pressed (rotation will be in positive direction, which is turning left) we can use formula front vector = sin(obj.y), 0.z, cos(obj.y).

Here is the link to some good explanation `http://www.euclideanspace.com/threed/games/examples/cars/startStop/index.html`

5.10 Destroying the projectile(Bullets or cannon balls) after certain time

```
setTimeout(function () {

            cannonBall.dispose();
        }, 3000);
```

5.11 Importing the animating pre built characters from Babylon (also called Meshes)

Mesh allows for a static 2D image to be broken down into customized polygons. These polygons can then be stretched or warped by the manipulation of each individual vertex. The ability to warp an image on a skeleton is immensely useful for creating secondary motion in an animation without re-

quiring any additional textures. The vertices can also be skinned to multiple bones, allowing the mesh to follow and bend with the changing directions of a skeleton. The influence each bone has over a vertex can also be adjusted and smoothed.

Babylon has built the extensive open source library of free meshes/characters which can be used modified with their tools. All the resources are available on their website www.babylonjs.com.

In our project we have created the class Dude which will pass the arguments to Dude.babylon.js to animate, resize and assign walking animation speed to one of our animated character and also will be used to for the creating box around him, so that he has collision detection on bounding box once its hits or collides which other characters.

Code Example used as follows:

```
function createHeroDude(scene) {

    var meshTask = scene.assetsManager.addMeshTask('DudeTask', 'him',
        'Dude/', 'Dude.babylon');
```

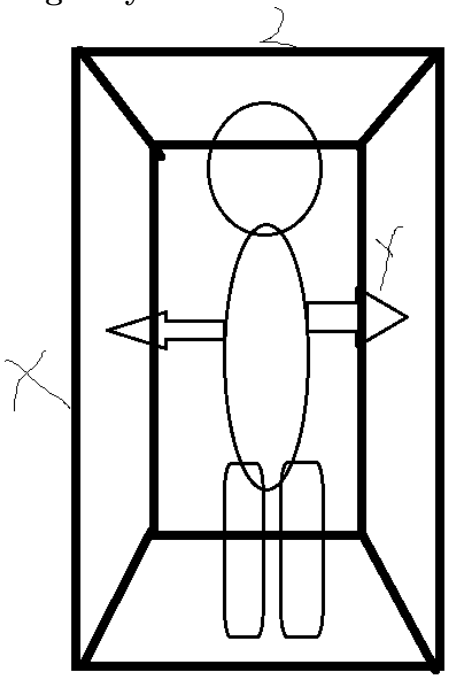
Here we have told the assets manager that the dude is in the Dude folder and the file has details for creation of Mesh is in Dude.babylon.

Other animated meshes can be seen with tutorial on the link `https://doc.babylonjs.com/resources/meshes_to_load`

5.12 Collision Detection and Imaginary Boxes

If we do not enable the collision detection around the objects, we will not be able to detect that object has been hit by projectile or collided with each other, also if we do not enable the collision all objects will be passing through them. For the object which is made of many meshes will need an imaginary box around it, which will contain all the meshes in one, just like a box holding lots of stuff in it. we can then keep track of the location of x, y and z direction of the box and will do collision detection by detecting those coordinates. Further information can be found on link `https://doc.babylonjs.com/api/classes/babylon.boundingbox`

Imaginary Box Around the Object



5.13 Creating the cameras in different prospective

I have used here few different camera 1 camera is rotating around with mouse, an-other one is following the object that is a box and the dude mesh (animated character) on pressing key it changes the camera being used. It gives the player feel of first-person shooter. Use the provided link to get sample code and tutorial `https://doc.babylonjs.com/babylon101/cameras.Wehaveuseduniversal`, follow and arc rotate camera in this project.

5.14 The Assets Manager

As the sound, images and 3D models can take a time to load in memory as games can be bit heavy on resources and client might see incomplete assets on their screen as loading is not complete. An assets manager comes handy as it helps to organize all those resources very easily. It helps they system to load the resources in memory at once and which can be used easily by the game later on the request. You can find the complete tutorial and

information about how to code the assets manager you can visit `https://doc.babylonjs.com/how_to/how_to_use_assetsmanager`

5.15 Using sound

We have used the assets manager of the Babylon.js to manage the sounds. Which takes the binaryTask function to accept files.

```
function loadSounds(scene) {
    var assetsManager = scene.assetsManager;
    var binaryTask = assetsManager.addBinaryFileTask('laserSound',
    'sounds/laser.wav');

    binaryTask.onSuccess = function (task) {
        scene.assets['laserSound'] = new BABYLON.Sound('laser',
        task.data, scene, null, { loop: false });
    }

    binaryTask = assetsManager.addBinaryFileTask('cannonSound',
    'sounds/cannon.wav');

    binaryTask.onSuccess = function (task) {
        scene.assets['cannonSound'] = new BABYLON.Sound('cannon',
        task.data, scene, null, { loop: false });
    }

    binaryTask = assetsManager.addBinaryFileTask('dieSound',
    'sounds/die.wav');

    binaryTask.onSuccess = function (task) {
        scene.assets['dieSound'] = new BABYLON.Sound('die',
        task.data, scene, null, { loop: false });
    }

    binaryTask = assetsManager.addBinaryFileTask('gunSound',
    'sounds/shot.wav');

    binaryTask.onSuccess = function (task) {
        scene.assets['gunSound'] = new BABYLON.Sound('gun',
        task.data, scene, null, { loop: false });
```

```
    }
}
```

5.16 Using particle system

The Particle system is used to emit the texture of blood and other animation to show that person is hit. There is a built-in particle system to Babylon.js which can be access via following lines of code:

```
createDudeParticleSystem() {

        // Create a particle system
        var particleSystem =
            new BABYLON.ParticleSystem('particles',
                2000, scene);

        //Texture of each particle
        particleSystem.particleTexture =
            new BABYLON.Texture('images/flare.png',
                scene);

        // Where the particles come from
        particleSystem.emitter =
            new BABYLON.Vector3(0, 0, 0);
            // the starting object, the emitter


        // Colors of all particles
        particleSystem.color1 =
            new BABYLON.Color4(1, 0, 0, 1.0);

        particleSystem.color2 =
            new BABYLON.Color4(1, 0, 0, 1.0);

        particleSystem.colorDead =
            new BABYLON.Color4(0, 0, 0, 0.0);

        particleSystem.emitRate = 100;
```

```
        // Set the gravity of all particles
        particleSystem.gravity =
            new BABYLON.Vector3(0, -9.81, 0);

        // Direction of each particle after it has been emitted
        particleSystem.direction1 =
            new BABYLON.Vector3(0, -1, 0);

        particleSystem.direction2 =
            new BABYLON.Vector3(0, -1, 0);

        particleSystem.minEmitPower = 6;
        particleSystem.maxEmitPower = 10;

        return particleSystem;
    }
```

5.17 Using Socket.io for Multiplayer

The Socket.io is used to communicate between server and client, we can use web sockets to instantiate the new object each time the request is received by server. Socket.io has socket.emit and socket.on methods to send and receive the data. Server has socket and client has client socket.

We have used following code to start node server with express within the main folder of the project (works if node is installed as well and server.js file created with the required code to start server.

node filename.js

We have some generic code to start the server file as follows:

```
var express = require('express');
var socket = require('socket.io');
var app = express();

app.use('/', express.static(__dirname));
app.get('/', function(req, res)
{
        res.sendFile('index.html');
```

```
});

var server = app.listen(3000, function () {
    console.log('server just started listening on port 3000 ....');
});//localhost:3000
```

The Express server is required to get server functionality, the socket is creating the multiple instance of supplied object. res.sendfile("index.html") is sending back the response as the web-page. The line app.listen tells the server to start at port 3000, hence browser can access the index.html at localhost:3000.

Array of object is created, once the request is received it tells socket.io to create new socket and assign socket id and push it in the array of object. socket.broadcast will tell every computer connected within the socket network that player has joined. Delete function delete the player once the browser is closed by the client.

As server and socket does not know where the location of the new player is, client has to send his x and y coordinates to update the location of its object. hence, we can see the player moving around as it is sent by the client.

5.18 Deploying it to Heroku server

Heroku is cloud storage service, where you can deploy your web apps and get link for the internet access. is designed to deploy web app not website, the difference is that web app has backend not only the HTML static pages. In order to use Heroku to deploy our game we must trick you into thinking that our game is a web application and not a web site. And we can do this by adding a dynamic file to our project. for that reason, we have created PHP file called index.php. PHP is used for backend development. This dynamic file will be served by Heroku and it will have a reference to our main index.HTML file to run the game.

In order to deploy your game to Heroku, you will need following things to be in-stalled and configured.

GIT
For pushing the game to the Heroku you will need command line interface of git installed. Link `https://git-scm.com/download/win`

Heroku
Also, you will need to install the Heroku command line interface. Link `https://devcenter.heroku.com/articles/heroku-cli`

Procedure to deploy web apps to Heroku

- create an account on Heroku website
- go into your project home directory folder.
- type Heroku login and verify your account with username and password.
- Now do the ordinary git commands on command prompt within the project folder, git init, git add ., git commit
- now create he Heroku application by command Heroku app:create application-name

After the process is finished you will see the message with the link to your deployed app. Which can be accessed by anyone using internet browser.

Chapter 6

System Evaluation

6.1 Testing

It was crucial to test the functionality of our project. Each component must be tested to provide working copy of our project. Full stack development was used to achieve this game project.

6.2 Build on Prototype

We first built the single player game and then built the multi player on top of our single player prototype. Because it was crucial to know first game functionality before progressing to next advance stage.

6.3 Testing Stages

6.3.1 First Stage

First testing of the game was performed on single player game. We checked if the game is loading on HTML page properly. Is it able to move the car around the terrain? Is it able to shoot laser and balls? Whether those projectiles able to hit and destroy the opposite player. Was the different scene working once we touch down the gate to another world?

6.3.2 Second Stage

In the second phase of testing we tested the multi player game. We tested that on instantiating the new browser window new object of car is created.

Whether player was able to move the car around the train.

6.4 Unit Testing

We used unit testing to test every line of code as we progress through the project. If I was not getting desired output, I had to debug the code first, before moving on to next stage.

I have used system testing on each stage of deployment to ensure all the components worked properly in a deployed environment. This testing method involved using game and all its features. We tried to test as wanted to build game to highest quality.

6.5 Performance

The game performed very well. The biggest problem was to make things be visible to each other in multi-player socket.io mode. Database chat system was working as expected but onload method was creating trouble once it sends the message, as game page was reloading. Loading of high content of 3D graphics was issue on performing well as well, because those require high speed connection to the internet.

6.6 Evaluation of Objectives

Here we will discuss the objectives we set for the project, and what were the task we planned in our objectives:

Deliver a game project which understands 3D JavaScript game development requirements.

Produce a simple easy to use web game

I did produce simple and easy to use web game. Using the Babylon.js and node express, also we tried to save the game on Heroku online server. Using the Babylon.js I was able to draw and animate basic character players. It was easy to navigate GUI, but syntax was not that easy to learn and debug.

6.7 Limitation, Issues and Improvements

As the project developed and I got more and more experience with Full Stack development methods, I came up with the better solution to develop against the problem.

6.7.1 Server

For the login, registration we have used PHP and stored the database in mysqli using the wamp application. for the multiplayer game we have used node express and socket io together as a server. As I become more familiar with JavaScript, node express and socket.io, I came to know better understanding of developing server-based games and getting the request and response over the internet using the http. We created multiple pages navigating from each other. The only way to access the game dashboard, which has link to single and multi-player mode is only accessible after registration and login.

6.7.2 Client

Client has used the browser to load the game as the project was stored on a local server and it was served in web browser, just like we load and visit websites.

Chapter 7

Conclusion

Although, it was a great learning experience to develop the game with pure JavaScript code within Babylon.js graphics WebGL engine, we have learned lots of the critical syntax and the programming logic's, I had dig deeper in the critical syntax with full of complicated functions and methods. There was a plenty of help on the inter-net, especially on the Babylon website www.BabylonJS.com but still,[11] I would like to suggest that using any plain JavaScript graphic engines are good for learning but should be avoided for production and ease of development is far reaching with these plain libraries. Hence, I would recommend using the Unity3d or some kind of frame-work which has support available for the game development, as they have drag drop, resize, transform, translate and multi-player network support easily integrated in the their environments to build the games. Working on pure JavaScript libraries requires very strong understanding of graphics and mathematics. From movements, animations, colors, and triggering the events require high understanding of mathematics i.e. rotating a object in similar way car does will require to move the object by front vector, which requires complicated understanding of sin, cos functions.

Chapter 8

Appendices

- **Github Repository Source Code Of Project**
  https://github.com/g00351263/Applied-Project-2019
- **Github Repository Source For Single Player**
  https://github.com/g00351263/Applied-Project-2019/tree/master/Tank%20Fight/Single%20Player
- **Github Repository Multiplayer Source Code**
  https://github.com/g00351263/Applied-Project-2019/tree/master/Tank%20Fight/multiplayer
- **Live instance For Multiplayer Running At Heroku Link Below**
  http://carmultiplayer.herokuapp.com/
- **10 minutes screen-cast game details available at**
  https://youtu.be/yzx-SLvj8gk
- **20 minutes screen-cast video with code explained is available at** https://youtu.be/_oOI20k4gz8

Bibliography

[1] N. Rovshen and G. John, "Native browser support for 3d rendering and physics using webgl, html5 and javascript," *BCI-LOCAL 2013 Local Papers of the Balkan Conference in Informatics*, 2013.

[2] D. C. Rousset and David, "Official website for babylon.js," 2020.

[3] P. F. Navarro, "3d programming with webgl and babylon.js for beginners," 2020.

[4] K. Coley, "Exporting 3d content for babylon.js," 2018.

[5] I. Nishanbaev, "A web repository for geo-located 3d digital cultural heritage models," *Digital Applications in Archaeology and Cultural Heritage*, vol. 16, p. e00139, 03 2019.

[6] N. D. T. Ioannis K. Chaniotis, Kyriakos-Ioannis D. Kyriakou, "Is node.js a viable option for building modern web applications? a perfomance evaluation study," *CrossMark Computing*, vol. 90, p. 23, 2014.

[7] K. Stefanoski, A. Karadimce, and I. Dimitrievski, "Performance comparison of c++ and javascript (node.js -v8 engine)," *n/a*, 09 2019.

[8] K. SUN, "Analysis of javascript programs: Challenges and research trends," *ACM Computing Surveys, Vol. 50, No. 4, Article 59.*, p. 36, 2017.

[9] J. D. W. Z. Muwei Jian, Yilong Yin, "Comprehensive assessment of non-uniform illumination for 3d heightmap reconstruction in outdoor environments," *Computers in Industry*, vol. 99, pp. 110–118, 2018.

[10] O. Nordquist and A. Karlsso, "Comparing performance when it comes to rendering voronoi height maps in 3d," 2017.

[11] S. Crown, "Improving visualization skills of engineering graphics students using simple javascript web based games," *Journal of Engineering Education*, vol. 90, 07 2001.