[1] [2]Babylon.js JavaScript Game Development

Raja Naseer Ahmed Khan

B.Sc.(Hons) in Software Development

February 02, 2020

Department of Computer Science and Applied Physics,

Galway-Mayo Institute of Technology, Dublin Road, Ireland



Contents

[1 Abstract 4](#_Toc37170103)

[1.1 Project Overview: 4](#_Toc37170104)

[1.2 Parts of the Project 5](#_Toc37170105)

[1.3 Requirements Specification in the Project 5](#_Toc37170106)

[2 WebGL 5](#_Toc37170107)

[3 Introduction to Babylon.js 3D engine 6](#_Toc37170108)

[3.1 Sample Code to draw sphere on the canvas [4] 6](#_Toc37170109)

[3.2 Tools: [5] 7](#_Toc37170110)

[3.2.1 Sandbox. 7](#_Toc37170111)

[3.2.2 Playground 7](#_Toc37170112)

[3.2.3 Spector.js 8](#_Toc37170113)

[3.2.4 Exporters 8](#_Toc37170114)

[3.2.5 Node Material Editor 8](#_Toc37170115)

[4 Heroku on Git 8](#_Toc37170116)

[5 Technology Review 9](#_Toc37170117)

[5.1 Java vs JavaScript: 9](#_Toc37170118)

[5.2 Babylon.js vs Three.js 11](#_Toc37170119)

[5.3 Socket.io vs WebSocket 12](#_Toc37170120)

[5.4 Node Express vs Hapi 13](#_Toc37170121)

[6 Methodology 16](#_Toc37170122)

[6.1 Version Control 16](#_Toc37170123)

[6.2 Agile Development Approach 17](#_Toc37170124)

[7 System Design 23](#_Toc37170125)

[7.1 Python Server of Initial Development 23](#_Toc37170126)

[7.2 Contents of Game Folder 24](#_Toc37170127)

[7.3 Using Visual Studio Code 25](#_Toc37170128)

[7.4 Making a Main Page 25](#_Toc37170129)

[7.5 Code Logic Page 25](#_Toc37170130)

[**a.** First Full Scene With Camera Lights and Box 25](#_Toc37170131)

[**b.** Output Image from Above Code**.** 26](#_Toc37170132)

[7.6 Creating Ground from Heightmap graphic files 27](#_Toc37170133)

[7.7 Assigning Movement and Fire/Shoot from the Object (Event Listener) 28](#_Toc37170134)

[7.8 Movement of car using front vector. 29](#_Toc37170135)

[7.9 Destroying the project after certain time. 29](#_Toc37170136)

[7.10 Importing the animating prebuilt characters from Babylon (also called Meshes) 29](#_Toc37170137)

[7.11 Creating the cameras in different prospective 30](#_Toc37170138)

[7.12 Using sound 30](#_Toc37170139)

[7.13 Using particle system 31](#_Toc37170140)

[7.14 Using Socket.io for Multiplayer 32](#_Toc37170141)

[7.15 Deploying it to Heroku server 33](#_Toc37170142)

[8 Conclusion and Recommendations 34](#_Toc37170143)

[9 Appendices 35](#_Toc37170144)

[10 Bibliography 35](#_Toc37170145)

1. Abstract

**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 syntaxes 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.

**Keywords:** JavaScript, Babylon.js, babylonjs, JavaScript game development, multiplayer game, socket.io

**About the Author:** I am Raja Naseer Ahmed Khan studying 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.

**Acknowledgements:** Special thanks to Dr. Brian McGinley for the support during the 1-year long project, also for helping all year long for enhancing the project and thesis.

* 1. Project Overview:

This module for Applied project with minor dissertation we are assigned to demonstrate our understanding of 4 years of study of Bachelor of Science in Software Development. We have progressed through the initial stage of learning basic of 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 (SDLC) i.e. agile, waterfall models.

* 1. Parts of the Project
  + Dissertation

In the dissertation writing section, we will research the present past and future of our project scope and, we will also document all the project functionality in details for the reader to understand that how we have approached the coding and 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 are being used in modern development environment today.

* + 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 of GMIT on weekly basis. In the weekly meeting with our supervisor we will show our weekly sprints of the project also 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. 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.

1. WebGL

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]

1. 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 & 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.

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

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](http://www.babylonjs.com)

* 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
  1. Sample Code to draw sphere on the canvas [4]

<!-- 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();

});

* 1. Tools: [5]
     1. **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! [1]

* + 1. 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

* + 1. Spector.js

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

* + 1. 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.

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]

* + 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).

1. 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 runtime 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.

That article does an honest job explaining the differences between Heroku and Amazon Web Services, but it is such as you can choose other IaaS (infrastructure) providers aside from Amazon Web Services. So finally, Heroku seems to merely simplify the method of employing a cloud provider but at a price.

Heroku host webapps not the website as we have website kind of in browser game, we need to trick Heroku think that this is a webapp, 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’); ?>

1. Technology Review
   1. Java vs JavaScript:
   2. JavaScript.

#### JavaScript is a client-side scripting language.

#### 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 webpage. 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% 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.

* 1. 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 JavaScript 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.
  1. 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](http://threejs.org/examples/)).

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 Silverlight game engine, Babylon.js maintains its penchant for web-based game development with features like collision detection and antialiasing. 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](http://www.babylonjs.com/).

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

* 1. Socket.io vs WebSocket
  + **WebSocket**:

It is the communication Protocol which provides bidirectional communication between the Client and the Server over a TCP 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.

* + - **Key features of WebSocket:**
      * WebSocket helps in [real-time communication](https://www.educba.com/real-time-analytics/) 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](https://www.educba.com/how-to-build-web-applications-using-mongodb/) 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](https://www.educba.com/websocket-vs-rest/) 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, Disconnect, 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.
  1. Node Express vs Hapi
  + **Node**

Node.js is an Open Source Server-Side Runtime Environment built on Chromes V8 JavaScript Engine. Node.js provides an Event Driven, Non-Blocking Input Output and Cross-platform Runtime 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.

* + - **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 platforms. 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.

**Simple Node Web 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 requests

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

* **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 ${server.info.port}’);

});

1. Methodology

Right now, will investigate the methodologies 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.

All through my four years at GMIT 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/Incremental Development programming.

* 1. Version Control

GitHub is a type of software company. If you are learning Web Development, Software 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 incase 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.

* 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.

The agile model consists of the following steps: -

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

In 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.

**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 software.
* 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.

**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.

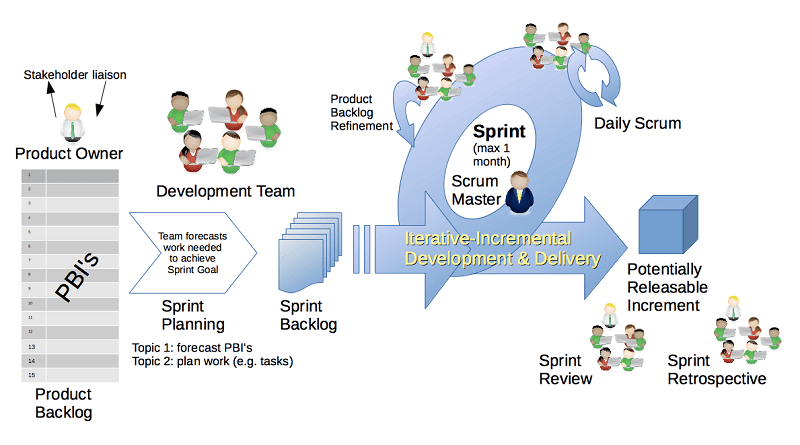
**Disadvantages of Agile Model**

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 cannot 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.

**Type of Agile SDLC models**

* Scrum



Scrum is a Software Product Development Strategy that prepares software developers as a team to reach a common goal. Manufactures ready market products. This is a widely used subset of agile software development.

Scrum is a system through which people and organizations can develop specific and appropriate ways of working for their time and condition.

The scrum rules and principles all serve the experimental principle or experimental working methods of control, as the best example for dealing with complex or complex situations.

In any case, there are more than the rules and principles for scrum. Scrum is more about behaviors than ways of working.

* Scrum Players

Three main role of Scrum team can be as follows:

* + - The Scrum Team
    - The Scrum Master

A Scrum Master is a facilitator and Servant Leader who encourages and demands self-organization from the development team, enables close cooperation across all roles and functions, addresses resource issue and disobedience of scrum practices, protects the team from external and internal distractions, removes impediments so the team can focus on the work at hand and follow scrum practices and is not typically a manager or lead, but he is an influential leader and coach who does not do direct command and control.

* + - The Scrum Product Owner

The Product Owner is responsible for maximizing return on investment (ROI) by identifying product features, translating these into a prioritized list, deciding which should be at the top of the list for the next Sprint, and continually re-prioritizing and refining the list. The Product Owner has profit and loss responsibility for the product, assuming it is a commercial product. Product Owner in Agile is like a spokesperson for customer and needs to represent them,

A Product Owner owns the Product backlog and writes user stories and acceptance criteria, is responsible for prioritizing the Product Backlog is prioritized and decides the release date and the content, accepts or rejects product backlog item, has the power to cancel the Sprint, if he thinks the Sprint goal is redundant and is the one who is responsible for the Return on Investment (ROI) of the product.

* Scrum Artifacts

In Scrum, artifacts are “information radiators” and they serve to capture the shared understanding of the team at a point in time.

In a co-located Scrum team, artifacts play a vital role for the team to reflect themselves on how they are doing with the sprint goal. Artifacts defined by Scrum are specifically designed to maximize transparency of key information so that everybody has the same understanding of the artifact.

Per the latest Scrum guide, Scrum framework defines 3 essential artifacts.

* + - Product Backlog
    - Sprint Backlog
    - Product Increment
* Definition of Done in Scrum

it is the acceptance criteria that are common to every single user story. It satisfies that code have had a review, have been tested and be immediately deployable.

* Scrum Events

There are five events in the scrum

* + - The Sprint
    - The Sprint Planning
    - The daily scrum
    - The sprint reviews
    - The sprint retrospective.
* Crystal Methodologies

The Crystal methodology is a family of smaller agile methodologies that is Crystal clear, Crystal yellow, Crystal red etc.

Crystal methodology focuses on early delivery of working software cycle, very high involvement of users. The Crystal methodology says each project should be unique and requires the application of different processes, practices, and policies. This is why crystal to be as one of the most lightweight agile methodologies.

* Feature Driven Development (FDD)

Involved with product owners, developers, stakeholders, the Feature Driven Development (FDD) methodology involves turning models into builds at every 2 weeks iterations. as compare to XP and SCRUM, Feature Driven Development (FDD) centers on strict operations that involve the walkthrough of domains, as well as design, code and inspection.

The model on lots of features. For every distinctive attribute, a development and design plan are implemented. after a series of inspections, a unit test is performed to see whether it is ready for the development stage.

Feature Driven Development (FDD) recommends a strict organization of the building process, which results in viable software that can be produced on a consistent basis. On the plus side, Feature Driven Development (FDD) facilitates high quality documentation and design also code assessment. However, the method demands a high level of design skill and planning prediction — early undesirable feature can lead to lengthy corrections. Feature Driven Development (Feature Driven Development (FDD)) is most optimal for big business developers in the banking and financial sectors, where process maturity and quality control are obligatory.

* Lean Software Development

The Lean Development Methodology is used to decrease waste, cost, and efforts. Lean software development can be a list of principles which then will be applied for software development to decrease programming effort, budgeting, and defect etc. The seven principles of lean software development are as following.

* Eliminate Waste

Do not add anything which is not valuable to customers, eliminate it as waste.

* Build Quality In

Wherever is possible use pair programming for enhance in coding quality. Use test driven development to avoid the time consumption and before coding to meet customer expectations

* Create Knowledge

Always document down the development experiences, use code reviews, comment the code, do training sessions, use tools to manage the project.

* Defer Commitment

Do not commit to and project before understanding and research, keep analyzing the important decisions.

* Deliver Fast

Delivering fast reduces the expenses, and customer satisfaction

* Respect People

Does not matters new hires or seniors, keep communicating, solve the issues, keep the people involved with each other without conflicts. It will increase the respect among each other’s as well productivity will go higher.

* Optimize the Whole

Do not overload developers, testers, or any part of the team. Keeping the project optimize will enhance the software quality and working capacity of individual.

* Extreme Programming (XP)

Extreme Programming (XP) involves a high level of participation between 2 parties within the software deliveries process. customers and developers. the 1st phase helps enhance development by focusing the most useful features of a given product through tests. The developers base each accomplished set of software upgrades on this feedback while continuing to check new features every few weeks.

Extreme Programming (XP) has its pros and cons. Extreme Programming involves a high level of collaboration and a minimum of up-front documentation. It is an efficient and chronic delivery model. However, the methodology also requires an excellent level of discipline, also as lots of involvement from people beyond the globe of data technology. Furthermore, so as for the simplest results, advanced Extreme Programming proficiency is significant on a part of every team member.

Extreme Programming (XP) is most suitable among small teams comprised of experienced developers with a proven data in communication and management.

* Pair Programming

Pair programming is an agile software development technique in which two programmers work together at one computer/device/pc. One, the developer, writes code while the other, the monitor reviews each line of code as it is typed in. The two programmers switch roles quiet often. While reviewing, the monitor also considers the direction of the work, coming up with ideas for enhancements and likely future problems to discuss. This gives freedom to the developer to focus all of his or her attention on the practical aspects of completing the current task, using the monitor as a safety net and guide.

1. 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 diagrams 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 websites, 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

* 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

* python 3 has built in support for server by using the command python -m http.server.

python 2 has http server using command python -m SimpleHTTPServer 8000

* 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.
  1. Contents of Game Folder
  + index.html

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.

* + js\Main.js

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.

* + 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.

* + 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.

* + 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.

* + CSS Folder

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

* 1. 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/>

* 1. Making a Main Page

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/

* 1. Code Logic Page

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.

### First Full Scene With Camera Lights and 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, 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);

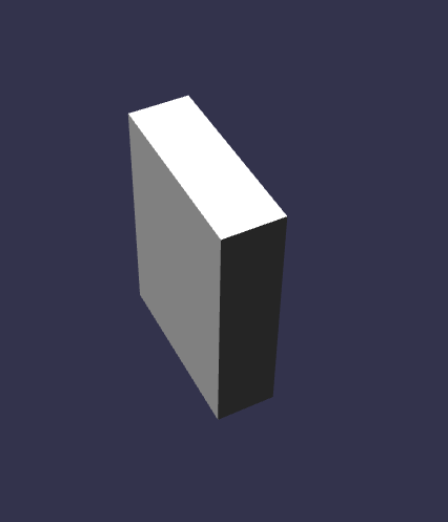
    // Add and manipulate meshes in the scene

    var box = BABYLON.MeshBuilder.CreateBox("box", {height: 1, width: 0.75, depth: 0.25}, scene);

    return scene;

};

### Output Image from Above Code.

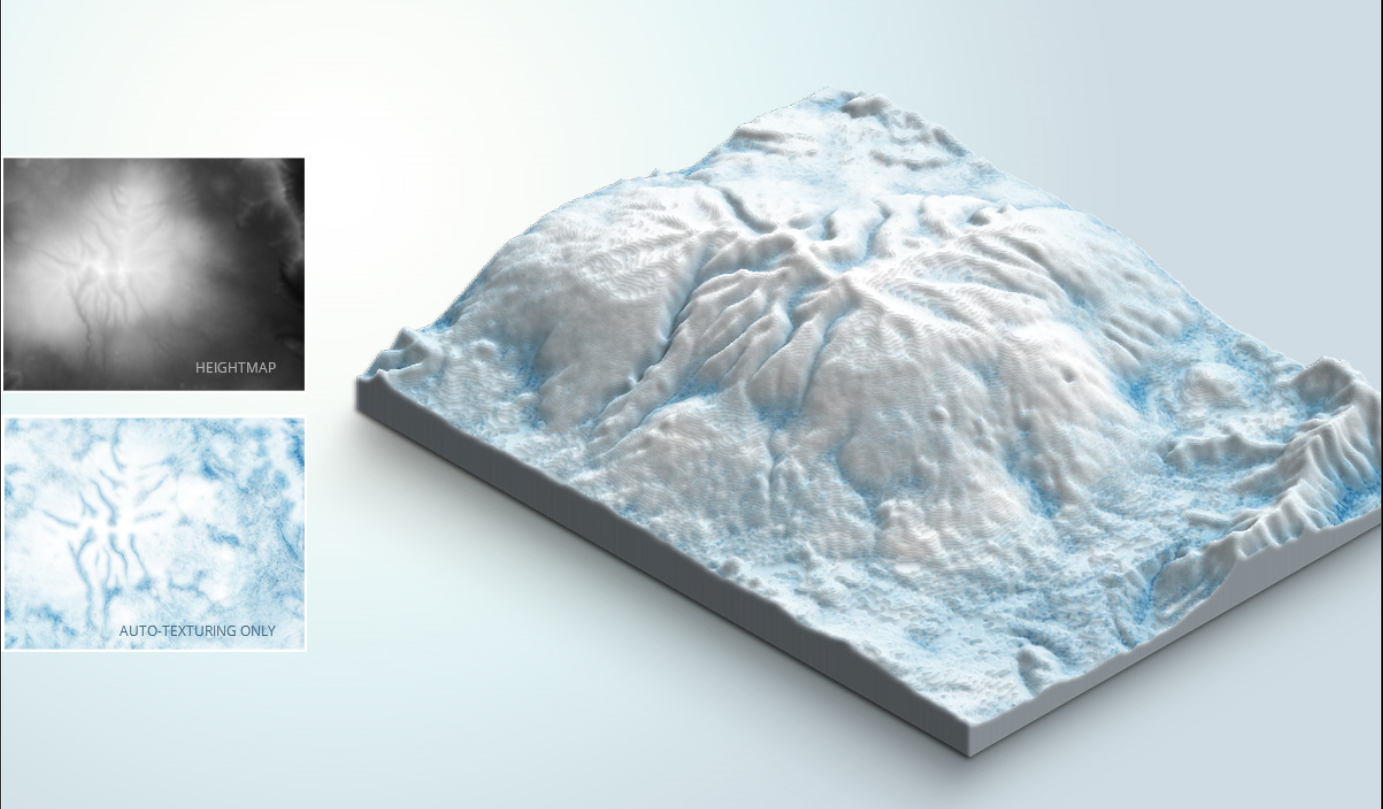


Link https://doc.babylonjs.com/babylon101/first

* 1. Creating Ground from Heightmap graphic files

The Heightmap files are one of the most cost-efficient and effective technologies for reconstructing 3D surface texture details [6]

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.



Code Example:

//Code for creating Ground from heightmap //

function CreateGround(scene) {  
 var ground = new BABYLON.Mesh.CreateGroundFromHeightMap("ground", "images/hmap1.png", 2000, 2000, 20, 0, 1000, scene, false, OnGroundCreated);

Link https://doc.babylonjs.com/babylon101/height\_map

* 1. 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)

***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;  
 }

});

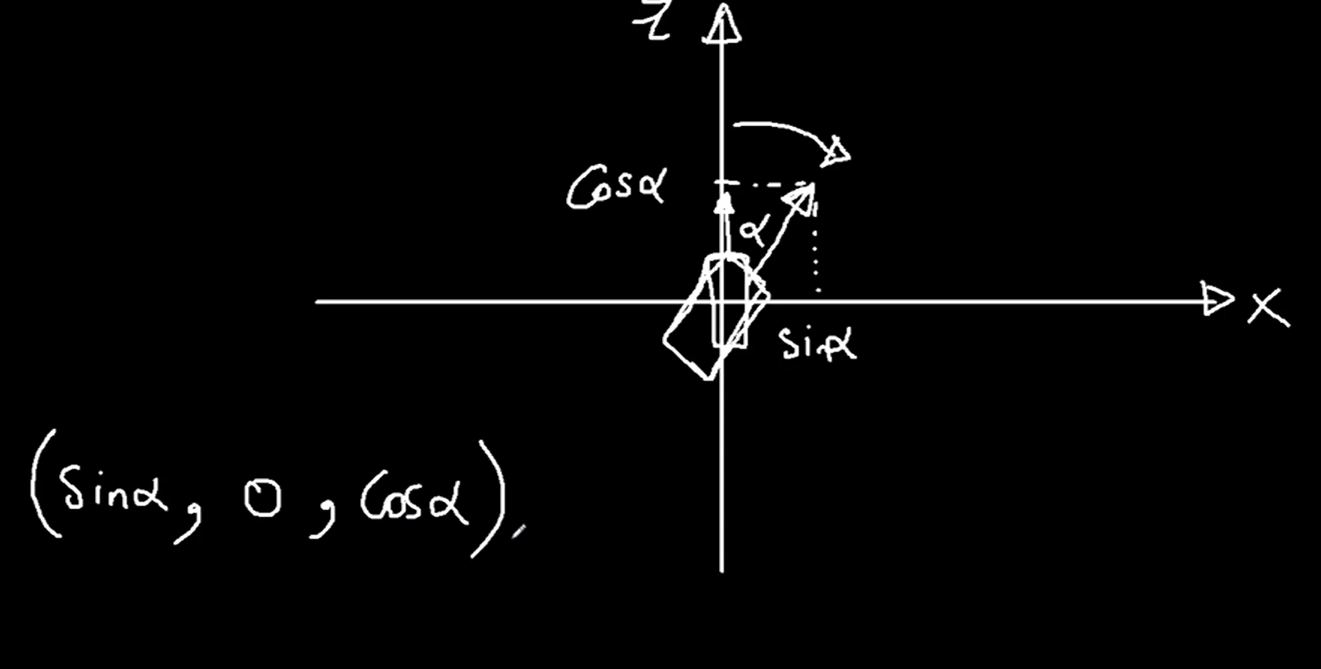
Code for some fire projectiles keys can be found as follows

tank.fireCannonBalls = function (scene) {

tank.fireLaserBeams = function (scene) {

* 1. 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.htm

* 1. Destroying the projectile(Bullets or cannon balls) after certain time.

setTimeout(function () {

cannonBall.dispose();

}, 3000);

* 1. Importing the animating prebuilt 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 requiring 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](http://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 used as follows:

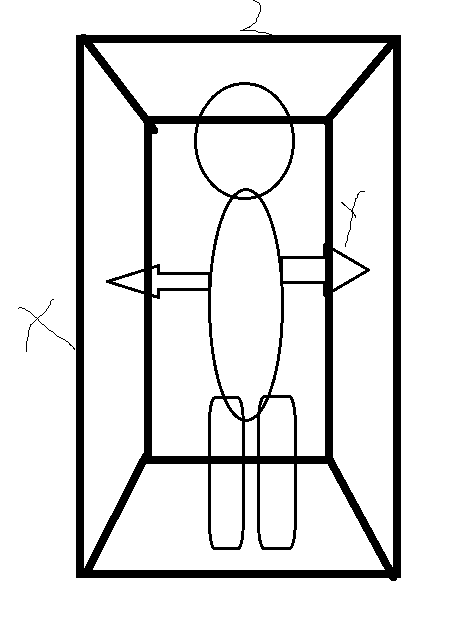
function createHeroDude(scene) {  
  
 var meshTask = scene.assetsManager.addMeshTask("DudeTask", "him", "Dude/", "Dude.babylon");

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

* 1. 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



* 1. Creating the cameras in different prospective

I have used here few different camera 1 camera is rotating around with mouse, another 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>. We have used universal, follow and arc rotate camera in this project.

* 1. 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

* 1. 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 });

}

}

[Link https://doc.babylonjs.com/api/classes/babylon.sound](Link%20https:/doc.babylonjs.com/api/classes/babylon.sound)

* 1. 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;

}

Link <https://doc.babylonjs.com/features/particles>

* 1. 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

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 webpage. 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.

* 1. Deploying it to Heroku server

Heroku is cloud storage service, where you can deploy your webapps and get link for the internet access. is designed to deploy webapp not website, the difference is that webapp 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 installed 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 webapps 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 applicationname

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.

1. Conclusion and Recommendations

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 syntaxes and the programming logics, I had dig deeper in the critical syntaxes with full of complicated functions and methods. There was a plenty of help on the internet, especially on the Babylon website www.babylonjs.com but still, 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 framework 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.

1. Appendices

* <https://github.com/g00351263/Applied-Project-2019>
* <https://www.babylonjs.com/>
* <https://socket.io/>
* <https://id.heroku.com/login>
* https://nodejs.org/en/

1. Bibliography

|  |  |
| --- | --- |
| [1] | K. Coley, "Exporting 3D content for Babylon.js," 10 2018. [Online]. Available: https://medium.com/@babylonjs/exporting-3d-content-for-babylon-js-76cb71a2df01. |
| [2] | J. G. Rovshen Nazarov, "Native browser support for 3D rendering and physics using WebGL, HTML5 and Javascript," *Local Papers of the Balkan Conference in Informatics,* vol. 13, no. 13, pp. 21,23,24, 2013. |
| [3] | A. K. Oscar Nordquist, "- Comparing performance when it comes to rendering Voronoi height maps in 3D," 2017. [Online]. Available: http://www.diva-portal.org/smash/get/diva2:1228221/FULLTEXT01.pdf. |
| [4] | P. F. Navarro, "3D Programming with WebGL and Babylon.js for Beginners," 2020. [Online]. Available: https://academy.zenva.com/product/3d-programming-with-webgl-and-babylon-js-for-beginners/. |
| [5] | D. C. a. D. Rousset. [Online]. Available: https://www.babylonjs.com/. |
| [6] | Y. Y. J. D. W. Z. Muwei Jian, "Comprehensive assessment of non-uniform illumination for 3D heightmap reconstruction in outdoor environments," *Computers in Industry,* vol. 99, pp. 110-118, 2018. |
| [7] | I. Nishanbaev, "A web repository for geo-located 3D digital cultural heritage models," *Digital Applications in Archaeology and Cultural Heritage,* vol. 16, no. March 2020, p. e00139, 2019. |
| [8] | L. A. M. D. LucaSciullo, "WoT Store: Managing resources and applications on the web of things," *Internet of Things,* vol. 9, no. n, p. 1000164, 2020. |