DOCUMENTATION

BALLOON SHOOTER GAME

**Table of Contents**

1. Philosophy
2. Threejs
3. Application
4. About Balloon Shooter
5. Technology
6. Application Structure
7. UI Architecture
8. Running the Application
9. Web Hosting
10. Contributor

## Philosophy

# **Three.js** is a cross-browser JavaScript library and Application Programming Interface (API) used to create and display animated 3D computer graphics in a web browser.Three.js uses WebGL. Three.js allows the creation of Graphical Processing Unit (GPU)-accelerated 3Danimations using the JavaScript language as part of a website without relying on proprietary browser plugins. This is possible due to the advent of WebGL.

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

# Three.js includes the following features:

# ***Effects***: Anaglyph, cross-eyed and parallax barrier.

# ***Scenes***: add and remove objects at run-time; fog

# ***Cameras***: perspective and orthographic; controllers: trackball, [FPS](https://en.wikipedia.org/wiki/First-person_shooter), path and more

# ***Animation***: armatures, [forward kinematics](https://en.wikipedia.org/wiki/Forward_kinematics), [inverse kinematics](https://en.wikipedia.org/wiki/Inverse_kinematics), [morph](https://en.wikipedia.org/wiki/Morph_target_animation) and [keyframe](https://en.wikipedia.org/wiki/Keyframe_animation)

# ***Lights***: ambient, direction, point and spot lights; shadows: cast and receive

# ***Materials***: [Lambert](https://en.wikipedia.org/wiki/Lambertian_reflectance), [Phong](https://en.wikipedia.org/wiki/Phong_shading), smooth shading, textures and more

# ***Shaders***: access to full OpenGL Shading Language ([GLSL](https://en.wikipedia.org/wiki/GLSL)) capabilities: [lens flare](https://en.wikipedia.org/wiki/Lens_flare), [depth pass](https://en.wikipedia.org/wiki/Depth_pass) and extensive post-processing library

# ***Objects***: meshes, particles, sprites, lines, ribbons, [bones](https://en.wikipedia.org/wiki/Skeletal_animation) and more - all with [Level of detail](https://en.wikipedia.org/wiki/Level_of_detail)

# ***Geometry***: plane, cube, sphere, torus, 3D text and more;

***Utilities***: full set of time and 3D math functions including [frustum](https://en.wikipedia.org/wiki/Frustum), matrix, [quaternion](https://en.wikipedia.org/wiki/Quaternion), [UVs](https://en.wikipedia.org/wiki/UV_mapping) and more

# ***Export and import***: utilities to create Three.js-compatible JSON files from within: [Blender](https://en.wikipedia.org/wiki/Blender_(software)), [openCTM](https://en.wikipedia.org/wiki/OpenCTM), [FBX](https://en.wikipedia.org/wiki/FBX), [Max](https://en.wikipedia.org/wiki/3D_Studio_Max), and [OBJ](https://en.wikipedia.org/wiki/Wavefront_.obj_file)

# Three.js runs in all browsers supported by WebGL 1.0.

Three.js is made available under the [MIT license](https://en.wikipedia.org/wiki/MIT_license)

# Every three.js app (in fact, nearly every real-time 3D app) will have the following basic components:

# a **Scene** - this is a holder for everything else. You can think of it as a “tiny universe” in which all your 3D objects live.

# a **Camera** - this is directly equivalent to the concept of a camera in the real world, and you’ll use this to view your scene.

# a <**canvas>** - this is an HTML canvas element, and just like a canvas in the real world, it starts out as a blank rectangle, just waiting to hold your beautiful creations!

* a **Renderer** - this is a machine that takes a **Camera** and a **Scene** as input and outputs beautiful drawings (or renderings) onto your **<canvas>.**

## Application

Three.js is the world's most popular JavaScript framework for displaying 3D content on the web, providing you with the power to display incredible models, games, music videos, scientific and data visualizations, or pretty much anything else you can imagine, right in your browser and on your smartphone! Each section is divided up into several chapters, each of which is a short, self-contained lesson. Section One introduces all the foundational concepts, and once completed, you can go through the other sections in any order. Each section introduces and teaches the concepts by guiding you to create small example applications in which we'll explore real-world scenarios.

Threejs is mostly used for making 3D action games and video games. Some of the famous real life applications are as follows:

* Time Spiders
* Lost in Transmission
* Stein World
* Spectres of the Cold
* Trudo
* Snakes and Ropes
* Balloon Shooting
* Might is Right
* Plume and the forgotter letter
* Phaser Driving
* Malrrabas Tower
* Racer
* Project Malgate
* Dragons in the sky
* Runner Game

## About Balloon Shooter

* Keyboard based balloon shooting game ,where a player need to move the gun pointer through the keyboard navigation keys.
* A player can shoot as many as balloon in 5 minute time.
* +10 points will be given for each shoot.
* Game should support touch devices and navigation buttons.

## Technology

Following libraries were used during the development of balloon shooting game:

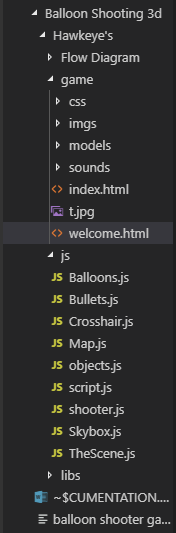
* **Threejs**: Cross browser javascript library and API.
* **Jquery**: Lightweight javascript library.
* **HTML**: Used to create documents on www.
* **CSS**: Describes presentation of web pages.
* **Javascript**: High level programming language that all modern web browsers support.
* **000webhost**: Generates a URL for hosting website.
* **Facebook SDK**: The Facebook SDK is what allows mobile app developers to integrate Facebook within a mobile app. SDK stands for software development kit, and it allows for a website or app to integrate with Facebook seamlessly.

Some other js libraries are:

* **Ammo.js**
* **Dat.gui.js**
* **Howler.js**
* **Jquery-1.9.0.js**
* **MTLLoader.js**
* **OBJLoader.js**
* **Physi.js**
* **Physijs\_worker.js**
* **PointerLockControls.js**
* **Stats.js**
* **Three.js**

## Application Structure

Threejs is a framework that makes our life very easy since we don't have to choose the versions of different dependencies. I have tried to follow the same ideology while creating the project structure, at first it might seem like overwhelming, but do believe me once you start writing your pieces the structure will help you immensely by saving your time and thinking about questions which are already answered. The structure look as follows :



We have worked on different modules, like creation of UI, creation of different objects, creation of animations, worked on calculations part and some extra things like frequency of gun hitting the balloon, increment of levels and so on.

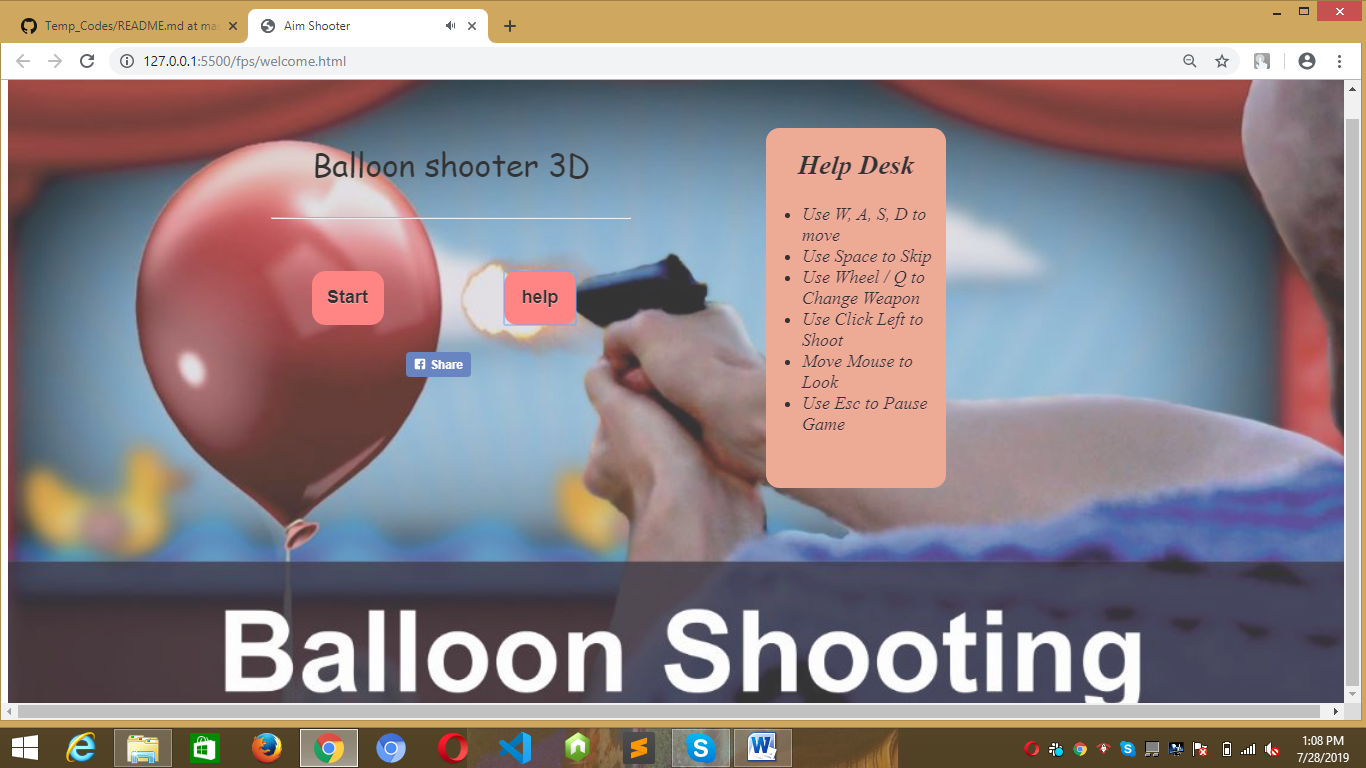
* For making balloons we have used sphere geometry and added texture and material to it. We have used *MTLLoader.js* for loading an .mtl resource (material template library format). MTL file format is a companion file format to.OBJ that describes surface shading (material) properties of object within one or more .OBJ files.
* For making bullets we have used *ammo.js* library which provides the functionality like original bullet.
* We are using *OBJLoader.js* which is a simple data format that represents 3D geometry in a human readable format as a position of each vertex.
* We are using *three.js* library for making crosshair and using matrix methods and applying translations by giving positions across x, y, and z axis. Similarly we have used three.js for creating scene, map and skybox.
* For keyboard navigations and mouse controls we are using *PointerLockControls.js*. In order to react according to full mouse and keyboard input, pointer lock functionality part of browsers can be setup, called and used.
* We have used sounds in our game, for that purpose we are using *howler.js*. It makes working with audio in js easy and reliable across all platforms.
* *Physi.js* is used in place of three.js to provide dynamic environment. Provides simple control over friction and restitution (bounciness), integrated collision detection and events.

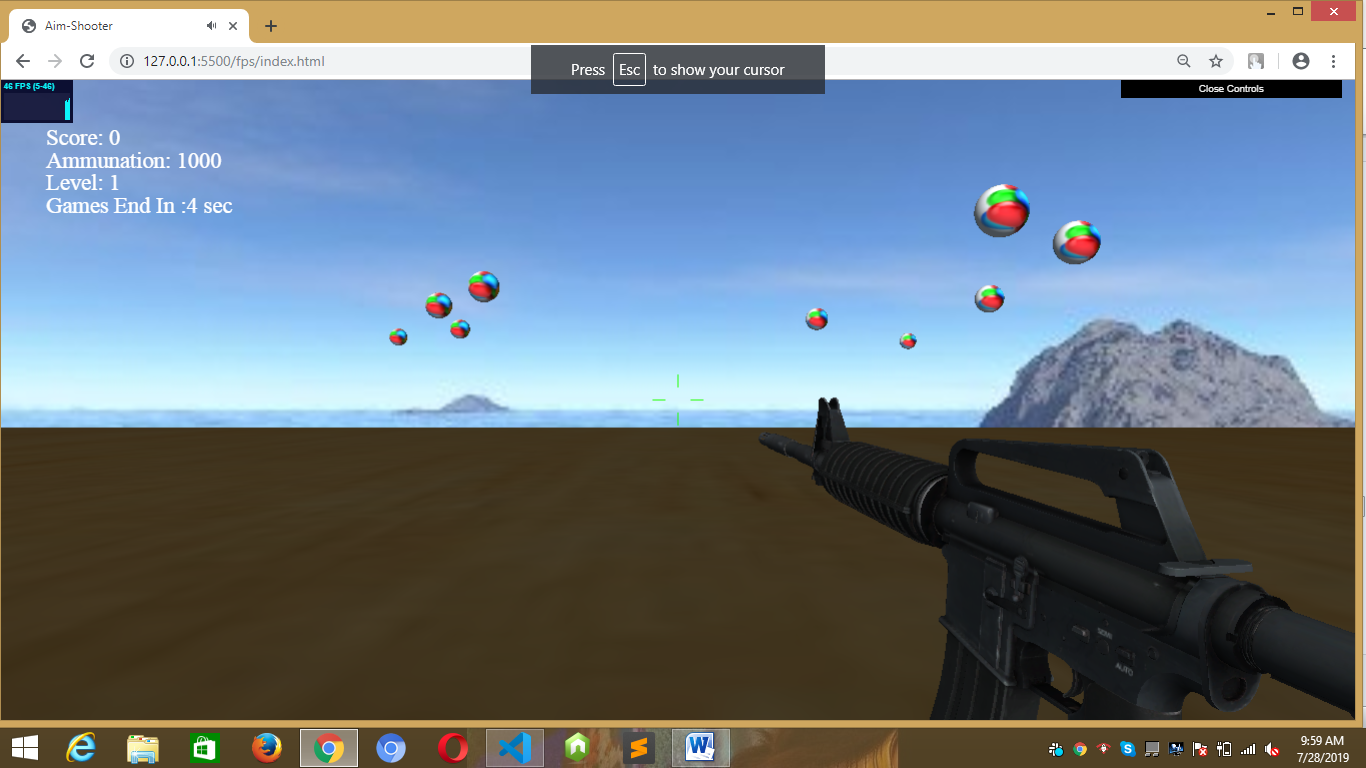
## UI Architecture

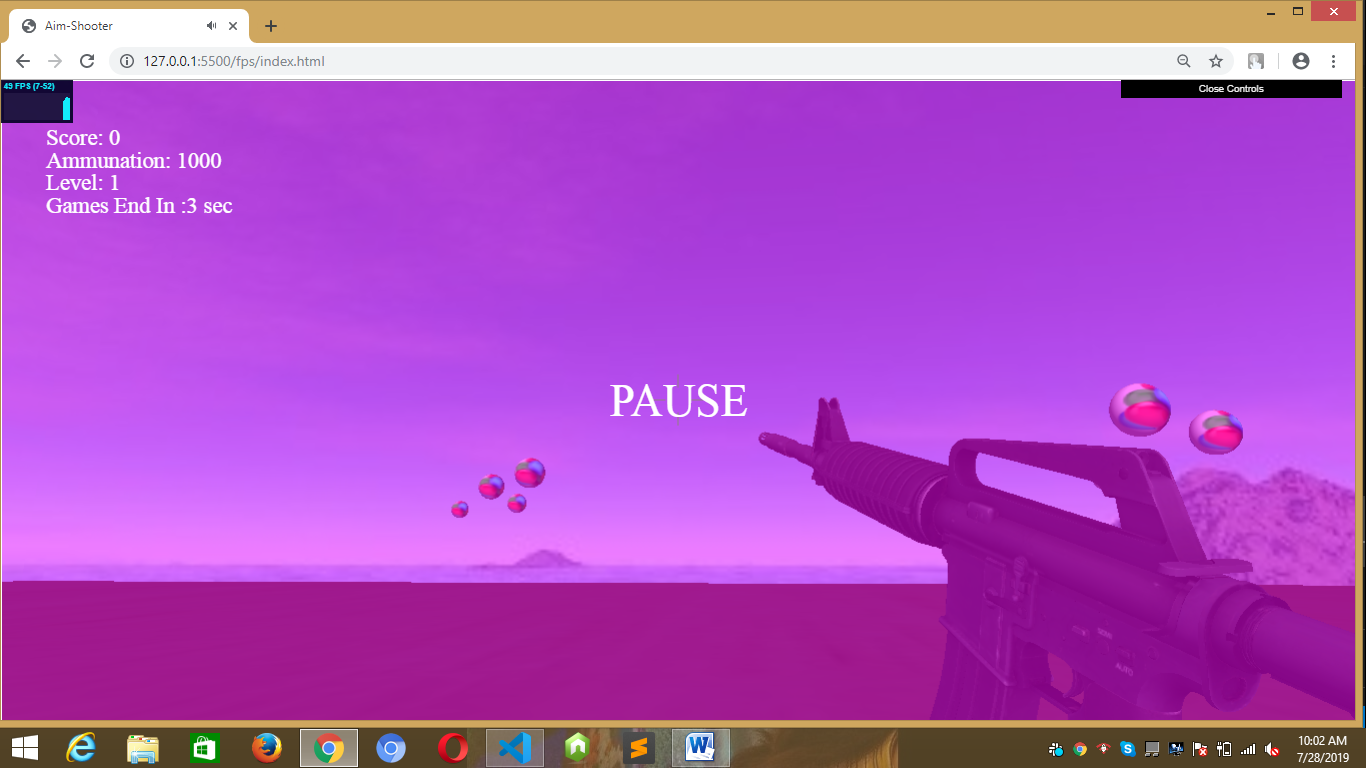
UI architecture plays an important role in design of any project. UI of our game has been designed keeping in mind the requirements of our customers. This game is mostly enjoyed by kids who are attracted towards playing more and more as they get extra points on shooting a balloon and level increment brings thrill and excitement to win. In some organizations, the UI architect may implement interface components and needs to be skilled with front end development tools like HTML, CSS, Javascript and common frameworks. UI design is the process of making interfaces in software or computerized devices with a focus on looks or style. UI design typically refers to graphical user interface but also includes others, such as voice controlled ones.

We have used mainly HTML and CSS for design of UI which will facilitate ease of using to end users. Some functions of jquery are also used as it is a lightweight javascript library where we write less and do more. The purpose of jquery is to make it much easier to use javascript on your website. It is mostly used to provide animation effects.

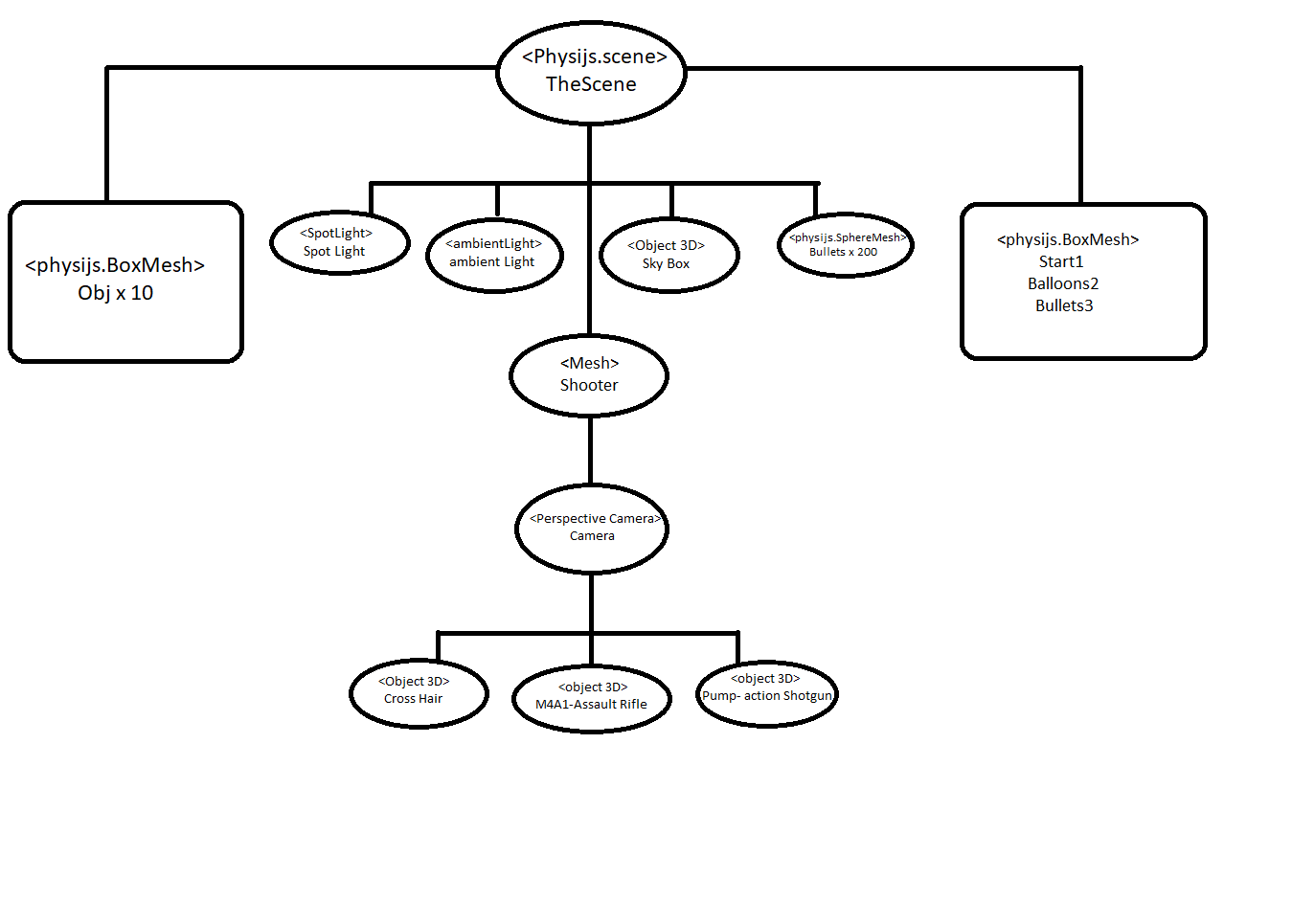
UI design basically consists of a start button, on click of which the game will get started. A help desk button is provided which contains instructions about how to play the game and there is a settings button for user to configure amount of time to play. We have used colorful images of balloons to make it enticing. In the end it displays the final score of the player when all the bullets have been used.







## Flow Diagram



## Running the application

* The most basic requirement for running a threejs application is a Modern Web Browser - Any modern browser can run three.js, where ‘modern’ means anything released in the last 5 years or so.
* WebGL - WebGL is an API for connecting your browser up to your graphics card, providing you with a lot more graphical processing power than is traditionally available on a website

How to include three.js in your projects:

There are quite a few different ways to include three.js in your JavaScript application, some of them simple, some of them a little more complex, however, they all boil down to one simple idea: you need to include one (and only one) of these three files in your project:

1. three.js
2. three.min.js
3. three.module.js

* include them in a script tag in your HTML <script>.
* import them directly into your JavaScript files.
* **Approach 1**: Download Them All!
* The simplest thing do is just download the entire three.js project onto your computer and use the files from there.
* **Approach 2**: Link to the files from a CDN
* Alternatively, you can link the files from a CDN (Content Delivery Network), which is a remote site dedicated to hosting files so that you can use them in a website.
* <script src="https://threejs.org/build/three.js"></script>
* **Approach 3**: Install three.js as an NPM package
* npm install three –save
* import \* as THREE from 'three';

## Web hosting

Web hosting is a big computer where people store their websites, it refers to a company that rent out their computer/servers to store your website and provide internet connectivity so that other users can access to the files on your website

Different types of web hosts are:

* Shared Hosting
* VPS Hosting
* Dedicated Hosting
* Cloud Hosting

## Contributors

* Akash Bansod
* Saurabh Bhushan
* Zenab Sultan
* Pinkar Kumari
* Aayush Joshi
* Abhijeet Soni