Functional Specification

Student name: Conor Reddin

Student number: 13336766

Program: Computer Applications

Title: Physics Room (4th year project)

Module Code: CA400

Supervisor: Mark Humphry

Date: 23/11/2016

Declaration

I declare that this material, which I now submit for assessment, is entirely my own work and has not been taken from the work of others, save and to the extent that such work has been cited and acknowledged within the text of my work. I understand that plagiarism, collusion, and copying are grave and serious offences in the university and accept the penalties that would be imposed should I engage in plagiarism, collusion or copying. I have read and understood the Assignment Regulations set out in the module documentation. I have identified and included the source of all facts, ideas, opinions, and viewpoints of others in the assignment references. Direct quotations from books, journal articles, internet sources, module text, or any other source whatsoever are acknowledged and the source cited are identified in the assignment references. This assignment, or any part of it, has not been previously submitted by me or any other person for assessment on this or any other course of study.

Functional Specification Contents

Table of Contents

1. Introduction

1.1 Intro

The following is a functional spec provided for an overview of “The Physics Room”. The purpose of this is to provide an understanding of the system design, requirements and architecture.

1.2 Overview

The product to be developed is a physics room that can be accessed from any modern internet browser. The application will be written using Three.js. The application should allow a user to enter its URL in a web browser. The user is taken to a physics simulator in which the user has complete control over the laws of physics. In this simulator the user is able to determine the environmental factors such as gravity strength, friction, air resistance. Once complete, an object (e.g. a ball) can be created and placed in the room. The object’s mass and elasticity are determined by the user and then it is thrown in the room. The user can then observe the objects behaviour.

All that should be needed to run this application is a PC and a web browser (e.g. Google Chrome or Firefox). The potential for this product is vast and could be used in multiple ways. One such way in which this application could be used is for educational purposes for secondary, or even primary, school students.

The application could be used to solve physics related equations such as calculating the stopping time or distance of a thrown object at a specified mass. This application could even include a feature so the user can be shown the calculations that are being done in order to demonstrate how it is producing the result that are being displayed.

1.3 Business Context

There are many means of education constantly being developed in the world in order to adapt to the needs of different students. Every student is different and it is important that the educator understands this as one way of teaching may not be as effective as another. My idea is to add to the ever growing list of tools at an educator’s disposal. This application could make giving the student an idea of how the laws of physics operate an easier task by adding visual aide to the demonstration.

1.4 Glossary

Three.js – is a cross-browser JavaScript API used to create and display animated £D computed graphics using WebGL.

WebGL – (Web Graphics Library) is a JavaScript API for rendering interactive 3D computer graphics with any compatible browser without the use of a plugin.

API – (Application programming interface) is a set of subroutine definitions, protocols, and tools for building software and applications.

Justinmind – is an authoring tool for web and mobile app prototypes and high-fidelity website wireframes.

2. General Description

2.1 Product / System Functions

The application is focused around the applied mathematics of Newtonian physics. It will act as an aid when answering physics related questions when learning or studying for an exam (e.g for the Leaving Cert). The user will have access to multiple pre-built physics rooms to demonstrate the application, each of which will answer a different question. The user will then be able to adjust each room in order to see how their changes affect their result. The code for this application will require multiple files and functions to handle all of this.

Third party scripts

- **Physi.js**
  - This is a third party file used to incorporate physics into the application and provide the laws of physics to the simulator.
- **Three.js**
  - This is a third party file that provides the JavaScript library to use WebGL.

My own scripts

- **Index.html**
  - This will be the main html page that will tie the program together.
- **World.js**
  - This will be the file that generates the world in which the physics will take place.
  - Functions that could be used for this are:
    - createWorld()
    - createObject()
    - run()
    - reset()
- **Objects.js**
  - This is where all of the objects will be stored for use within the application.
  - Functions that could be used for this are:
    - getCircle()
    - getSquare()
    - placeObject()
    - removeObject()
- **Menu.js**
  - This is where the menu that will act as the user interface will be generated for the application.
  - Functions that could be used for this are:
    - start()
    - return()
    - chooseWorld()

2.2 User Characteristics and Objectives

A potential target community of users for this application is students in secondary level education. Students who wish to gain a better understanding of the laws of physics can use this application to get the answer to a particular question (e.g. distance travelled by object in a given time, etc.). Potential users may also wish to access the application to “tinker” and have fun in order to get a better understanding the impact and effect of changing certain physical properties (e.g. force due to gravity, etc.). Users of this application are expected to know how to use an internet browser on a computer and how to navigate a menu on a website.

2.3 Operational Scenarios

- An example is run

The user wishes to see a demonstration of a physics question. The user, therefore, clicks on one of the pre-made examples and clicks run. The user then observes the application performing the example and displaying the result. The user is able to see the mathematics being performed and the end result. Upon running the application the user will be presented with a message saying “finished” followed by two buttons asking if the user wants to reset the example or go back to the main menu. If the user is still unclear of what something means or how it contributed to the outcome the user is able to adjust the equation and see how the result differs. The user may continue this until they are satisfied.

- An educator is giving a demonstration using the application

The educator will load up the application on their desktop and will then be able to project the simulator up in front of the class for the students to see. Once the application is up and running the teacher can carry out any changes in the side menu, while explaining everything to the students. This should assist the educator as it will make the explanation clearer and be effective at keeping the students interested.

2.4 Constraints

Time

There may be issues with time throughout the course year. This application will be developed alongside full time education meaning other assignments, exams and study may make distributing the time effectively difficult. There are many routs of growth for this project. In fact the list of additions that could be made to this application could never end. This could impact the development of the project as it will be important to know when to move on to the next phase.

JavaScript and Three.js

Although I have experience with JavaScript I have no experience with Three.js or WebGL. This will inevitably result in hurdles that I must overcome. Fortunately there are bountiful sources of information online to help me.

Appearance

With the amount of time that may need to be given in order to make the application function. I am concerned that an adequate amount of time will not be given to the visual appeal of the project. In web design it is important to give an application a professional look in order to give it a more trustworthy feel. For this reason care will be given to the graphical side of this project but it may not be as good as it could have been.

Requirements

- The application will require a device that supports HTML5 (a computer built within the last 5 years or a new tablet).
- The application will not be adapted to mobile devices as the interface would be too small and awkward for the user.
- Older machines (i.e. those built more than 5 years ago) may have difficulty running the application effectively.
- The application should be able to run on any modern browser such as Safari (v9.1), Google Chrome (v52), Microsoft Edge (v14) or Firefox (v48).

3. Functional Requirements

Main Menu

- **Description –** When the user first loads up the application this will be the first thing the user sees. Once loaded up the user will be presented with multiple buttons to generate different worlds. There will be buttons for different examples in the simulator such as “stopping distance example” or “deceleration example”.
- **Criticality –** This will be to ensure that the user will be able to get right to their reason for using the application without needing to filter through unnecessary details.
- **Technical issues –** This will get its own section of code separate from the rest. This shouldn’t be too difficult to make.
- **Dependencies –** This will be what initiates the world to generate and loads the other functions.

World

- **Description –** This is the function that will generate the desired world for the user in the simulator. This environment is what the user will use to both carry out and observe the test being performed.
- **Criticality –** This is the environment the user will be able to carry out their experiments.
- **Technical issues –** The main menu will need to be able to initiate the different functions in this code. Once it is run it will need to be able to accommodate new objects as well as re-initialise the physics as they change. As the program changes so will this so this will most likely be the code that is repeatedly altered the most.
- **Dependencies –** This will be the bridge that connects the user input to the rest of the program. As such this will need to have access the majority of the rest of the code.

Side menu

- **Description –** This will be displayed alongside the simulator while the application is running. This will be the interface that will allow the user to specify the laws of physics as well as to generate objects. Once the user is done changing the physics the click the “Apply” button and then all the changes will be set.
- **Criticality –** This will be a key part of the application as this will be what allows the user to adjust to world to their needs in order to carry out their experiments.
- **Technical issues –** Since this will be what the user uses to change the environment this will be one of the most difficult things to implement.
- **Dependencies –** It will need to have access to the world, objects and physics sections of the code so that the user can alter them as needed.

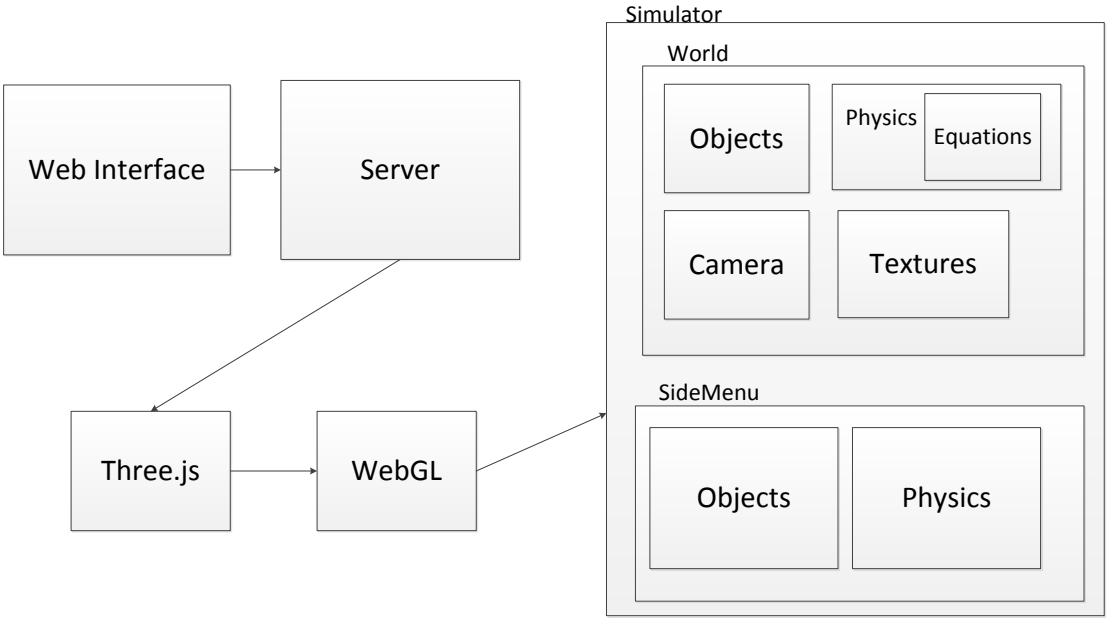
Objects

- **Description –** These are what the user will create inside the world. They will move according to the laws of physics in the world. The user will be able to specify the objects size, mass and elasticity.
- **Criticality –** This will be what the user uses in order to visually see the effect that the physics is having inside the world.
- **Technical issues –** Objects will get its own section of code. This may be difficult to implement because the user will need to be able to generate them freely and they will all need to use the physics functions.
- **Dependencies –** These objects will need to be able to occupy the world and will need to be able to use the physics functions.

Camera

- **Description –** This is what will allow the user to observe the application as it runs. The camera will need to be able to track the experiment as it is being carried out.
- **Criticality –** The user will be able to use this to look around the world and observe the simulator being run.
- **Technical issues –** Three.js comes with functions for using the camera and so this will hopefully be one of the simplest things to develop.
- **Dependencies –** The camera will need to be generated along with the world and will need to be able to occupy it.

4. System Architecture

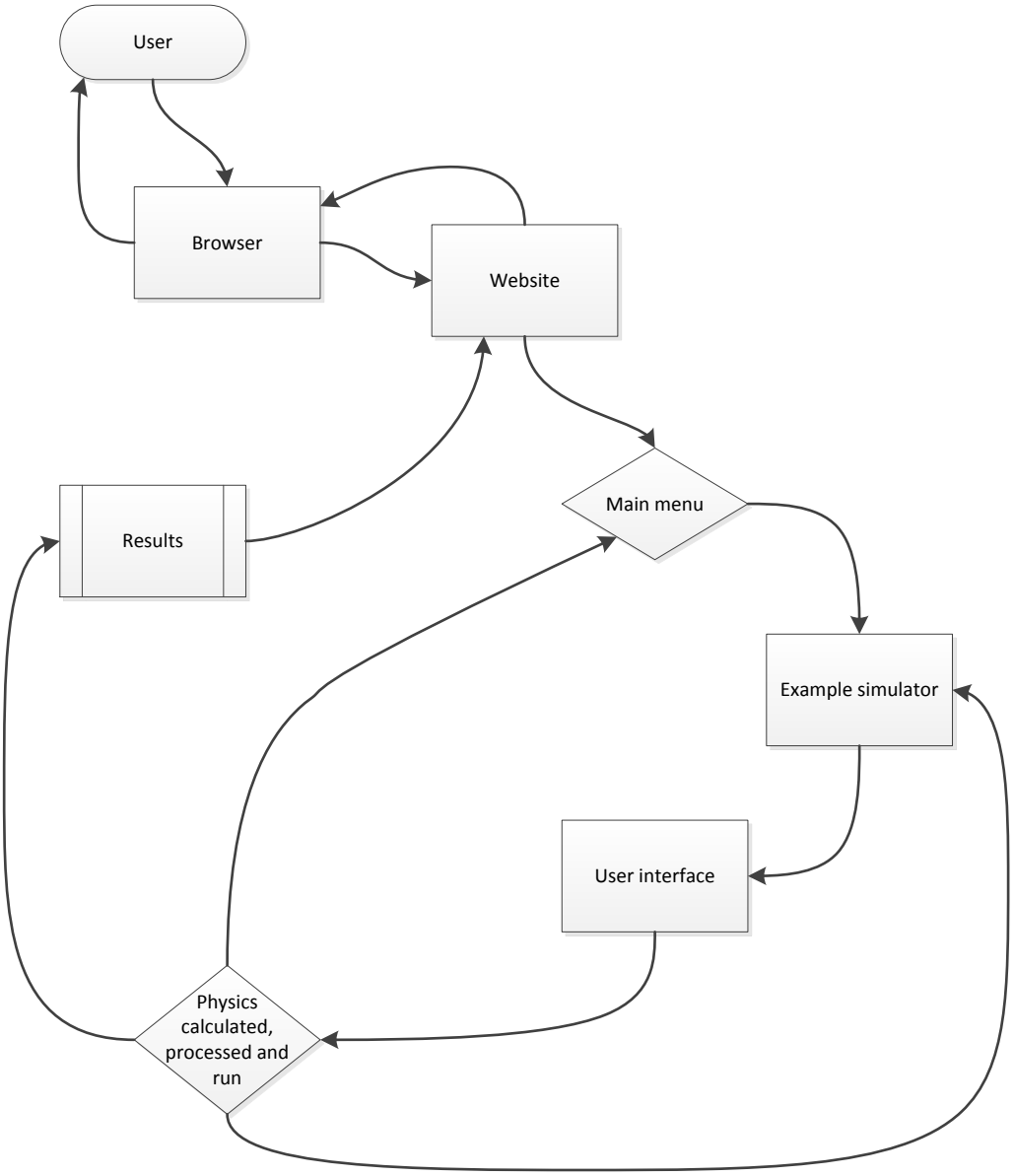


“The Physics Room” will be developed using WebGL (Web Graphics Library) which is a JavaScript API for rendering interactive 3D images. This project will also make use of a third party JavaScript API called “Three.js” which is a framework developed to assist in using WebGL.

The application will be compatible with Mac, Windows and Linux as well as any modern browser. I will, however, be focusing on Windows and Google Chrome for development.

As shown in the diagram above, the users will access the website via a browser on their computer. The website will then run the Three.js code to start up the simulator. Once the simulator is running the user will then be presented with the world and a side menu while the other aspects of the code run in the background.

5. High-Level Design



Above is a flow diagram showing a high-level running of the applications. The users will use a browser on their machine in order to access the application's website. The diagram shows which pre-made simulator the user will use. Finally, the diagram shows how once the application is run, the user will be shown the results and then taken back to the simulator to start again.

6. Preliminary Schedule

	21-Nov	05-Dec	19-Dec	02-Jan	16-Jan	30-Jan	13-Feb	27-Feb	13-Mar	27-Mar	10-Apr	24-Apr	08-May	22-May
Functional spec														
Solve intial issues														
Study javascript														
Create world														
Incorporate physics														
Make physic adjustable														
Graphics														
Develop menu														
Give menu functionality														
Develop maths														
Feedback analysis														
Writeup														
Ongoing improvements														
Preparation for demonstration														
Project demonstration														

7. Appendices

See below a sample question (taken from Leaving Cert physics paper) and a mock-up of the application created using Justinmind.

(*a*) A hurler strikes a sliotar with an initial velocity of 41 m s^{-1} at an angle of 30° to the horizontal. How far does the ball travel horizontally in three seconds?