ASTEROID GAME ENGINE

Aaron de Miranda Colaço
Kimberly Cabral
Joston Fernandes
Natasha Priolkar
Prajakta Kuncolienkar

Dissertation submitted in partial fulfillment of the requirements for the degree of

BACHELOR OF ENGINEERING

IN

INFORMATION TECHNOLOGY

OF GOA UNIVERSITY

2016

INFORMATION TECHNOLOGY
PADRE CONCEICAO COLLEGE OF ENGINEERING
VERNA GOA – 403722

ASTEROID GAME ENGINE

Aaron de Miranda Colaço
Kimberly Cabral
Joston Fernandes
Natasha Priolkar
Prajakta Kuncolienkar

Dissertation submitted in partial fulfillment of the requirements for the degree of

BACHELOR OF ENGINEERING

IN

INFORMATION TECHNOLOGY

OF GOA UNIVERSITY

2016

INFORMATION TECHNOLOGY
PADRE CONCEICAO COLLEGE OF ENGINEERING
VERNA GOA – 403722

PADRE CONCEICAO COLLEGE OF ENGINEERING
VERNA GOA – 403722

ASTEROID GAME ENGINE

Bona fide record of work done by

Aaron de Miranda Colaço	**201203116**
Kimberly Cabral	**201203110**
Joston Fernandes	**201203130**
Natasha Priolkar	**201203207**
Prajakta Kuncolienkar	**201203202**

Dissertation submitted in partial fulfillment of the requirements for the degree of Bachelor of Engineering in Information Technology under Goa University

2016

Approved By:

……..……. Prof. **Andrea D'Souza**

……..……. Prof. **Cynara Fernandes**

……..…….. Prof. **Razia Sardinha**
Head of the Department

……..……. **Dr. Luis Mesquita**
Principal

Examined By:

Examiner 1: ……...…….
(Name / Signature / Date)

Examiner 2: ……...…….
(Name / Signature / Date)

CONTENTS

CHAPTER **Page No.**

Acknowledgement

We take immense pleasure in the successful completion of **Asteroid Game Engine**. The environment at Padre Conceicão College of Engineering, Goa greatly facilitated the completion of this project.

We greatly appreciate the support, encouragement, and understanding extended towards our project by our guide Prof. **Andrea D'Souza** and co-guide Prof. **Cynara Fernandes**. Thank you for postponing the reviews every time one of us couldn't make it.

We acknowledge the effort by our principal **Dr. Luis Mesquita** to instill in us the Zen-like calm needed to fix badly broken and terribly documented C++ code. We sincerely hope we never come across a situation like it in our professional career.

We thank **Tanay PrabhuDesai** for helping us get started on the right foot and showing us how to follow a consistent pattern in development and testing.

We thank **Saurabh Nanda** for taking the time off from his busy schedule to advise us, and for the trigonometry which inevitably made us question our sanity.

Any acknowledgement would be incomplete without thanking our parents for all the support they extended.

LIST OF FIGURES

LIST OF ABBREVIATIONS

1. **CPU:** **Central Processing Unit**
2. **HTML:** **Hyper Text Markup Language**
3. **CSS:** **Cascading Style Sheets**

1. Introduction

Everybody likes to play games. The game we developed aims to be very light-weight and intuitive. There are no memory leaks, nor CPU intensive tasks.

It consists of a space environment with freely moving meteors. The user controls a rocket and shoots at the meteors to break them apart. The player is provided with 3 lives at the beginning of the game.

At the end of the game the user enters his name and e-mail address and has the option to share his score with friends.

No external game engine was used in the creation of **Astrofighter**.

The entire game logic is written in JavaScript and HTML5 & CSS3 are used for the front end. Ruby's Sinatra is used on the back end. The game does not require high bandwidth and loads minified files to improve already fast load times.

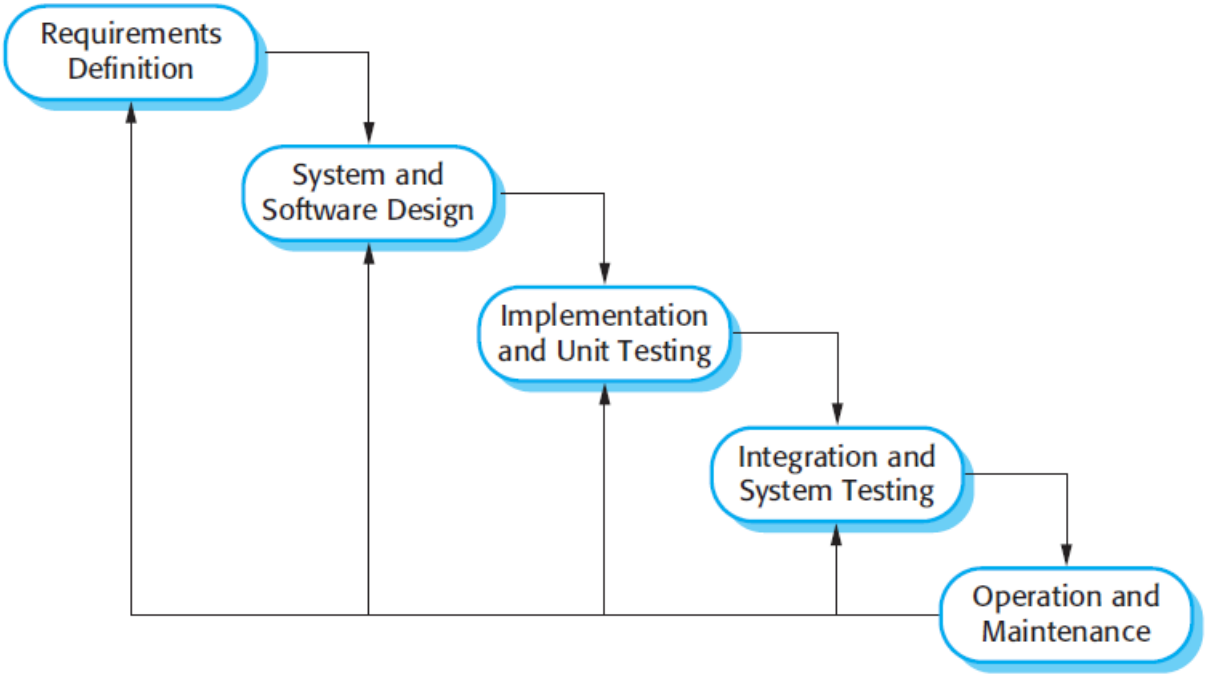
The biggest challenge proved to be the calculations for speed and collision detection.

'**Astrofighter**' has been used to refer to the entire application created.

The project is officially named '**Asteroid Game Engine**'.

2. Project Organisation

2.1 Process Model



2.2 Tools and Techniques

Development Environment:

- **Operating system:** Ubuntu 14.04 LTS
- **CPU:** Intel x86/x86_64 architecture CPU
- **Graphics:** NVIDIA GeForce 710M
- **Editor:** Sublime Text 3 (beta)
- Ruby-2.2.1
- Sinatra 1.4.7
- JavaScript (ECMAScript 6)
- SQLite3 & PostgreSQL
- Git 1.9.1
- Gimp 2.8.10

3. Software Requirements Specification

3.1 Purpose

The purpose of this document is to give a detailed description of the requirements for Astrofighter. It will illustrate the purpose and complete declaration for the development of the system. It will also explain system constraints, interface and interactions with other external applications.

3.2 Document Conventions

Main Sections Titles:

Font: Liberation Serif
Face: Bold
Size: 20

Subsection Titles:

Font: Liberation Serif
Face: Bold
Size: 18

Other Text Explanations:

Font: Liberation Serif
Face: Normal
Size: 14

3.3 Intended Audience and Reading Suggestions

This Software Requirements document is intended for:

- **Developers** who can review project's capabilities and more easily understand where their efforts should be targeted to improve or add more features to it (design and code the application – it sets the guidelines for future development).
- **Project testers** can use this document as a base for their testing strategy as some bugs are easier to find using a requirements document. This way testing becomes more methodically organized.
- **End users** of this application who wish to read about what this project can do.

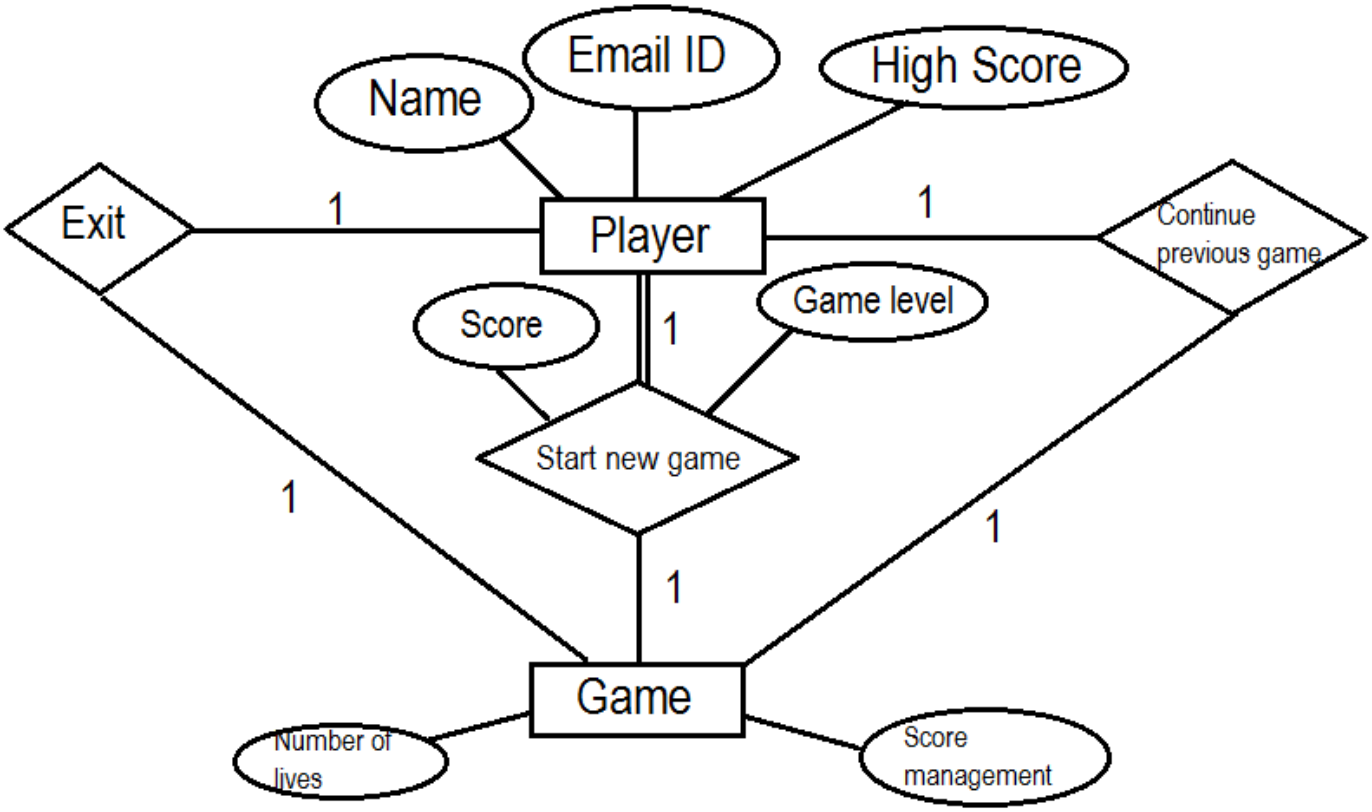
3.4 Project Scope

Astrofighter is a game application which provides end users an interface to play a space based game.

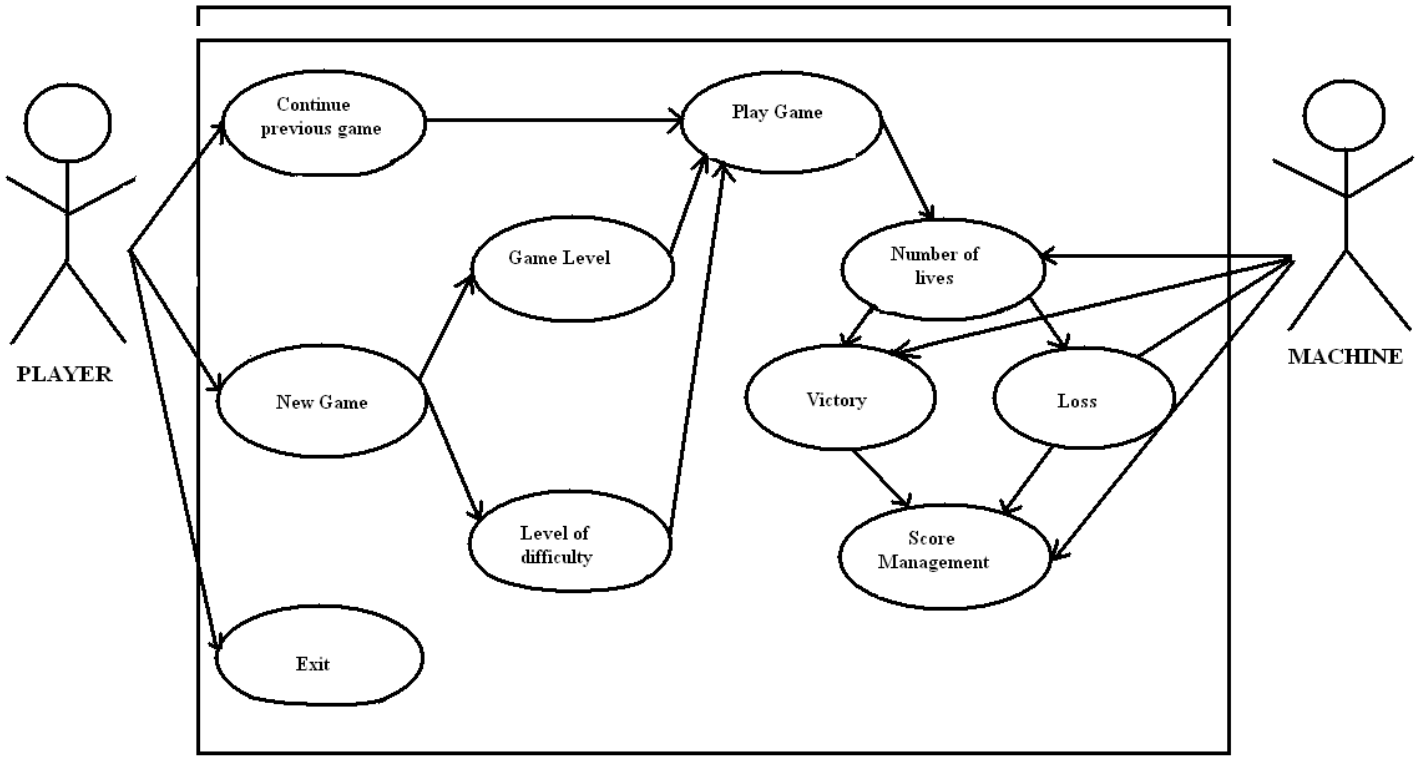
It is a low bandwidth game that is fun and intuitive, with a lot of visual cues.

We hope to foster a competitive gaming environment.

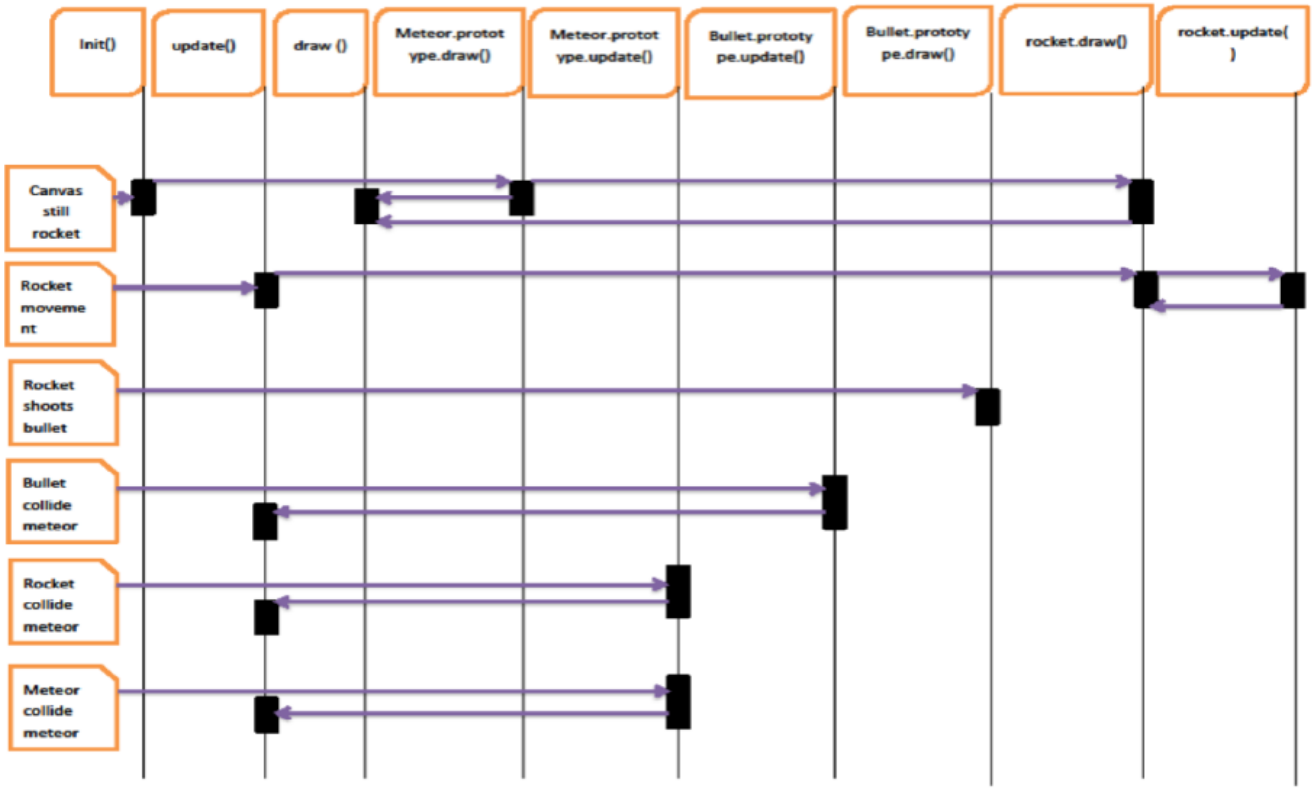
3.5 Entity-Relationship Diagram



3.6 Use Case Diagram



3.7 Sequence Diagram



4. Overall Description

4.1 Directory structure

4.2 Product Functions

- Meteor movement
  - Meteor.update : updates meteor coordinates
  - Meteor.draw : draws meteor on canvas
- Rocket movement
  - Rocket.update : updates rocket coordinates
  - Rocket.draw : draws rocket on canvas
- Bullet movement
  - Bullet.update : updates bullet coordinates
  - Bullet.draw : draws bullet on canvas
- Wrap around - Wraps around meteors and rocket

```
if (this.x > global.width) {
   this.x = 0;
} else if (this.x < 0) {
   this.x = global.width;
} else if (this.y > global.height) {
   this.y = 0;
} else if (this.y < 0) {
   this.y = global.height;
};
```

- Collision detection
  - Rocket & Meteors
  - Bullet & Meteors
  - Meteors
- Game audio - Audio for the game (crash, explosions)
- Lives
- User Scoreboard - List of all scores

4.3 System interfaces

Astrofighter integrates two internal systems to provide functionality:

- **Client** The game has an interface to the user’s client to receive user input and moves selections for the game.
- **Network** The game has an interface to the network in order to transmit information.

4.4 Operating Environment

The game needs an Internet browser that supports HTML5 and JavaScript to run, preferably Google Chrome.

V8 is Google's open source high-performance JavaScript engine, written in C++ and used in Google Chrome. It ensures the JavaScript game engine runs at optimal speeds.

5. External Interface Requirements

5.1 User Characteristics

Astrofighter is a very intuitive game and the only thing the user is required to know are the rules of the game, which are provided.

5.2 Hardware Interfaces

Astrofighter runs on any computer which meets the following criteria:

- Capable to use an Internet connection
- Includes memory storage
- Includes a mouse
- Includes a keyboard

5.3 Software Interfaces

Requires an HTML5 and JavaScript enabled web browser.

6. Other Non-functional Requirements

6.1 Performance Requirements

The game is expected not to hang, nor lag.
There are no undefined actions, nor unhandled errors.

Objects are de-referenced as soon as they are no longer needed. This ensures the garbage collector can easily recover memory that is no longer needed. This in turn ensures the game does not consume too much memory and there are no memory leaks.

6.2 Software Quality Attributes

The game should be

- **Correct:** It should ‘just work’.
- **Flexible:** The user can move, shoot, and carry out operations however he wants to. There are no fixed movements.
- **Maintainable:** The code is clean, modular, and adheres to standards.
- **Portable:** The game engine is portable to any framework.
- **Reliable:** The physics behind the game is reliable and works predictably. There are no undefined results.
- **Testable:** The game engine is easy to test; the code provides for easy testing, deleting, etc.

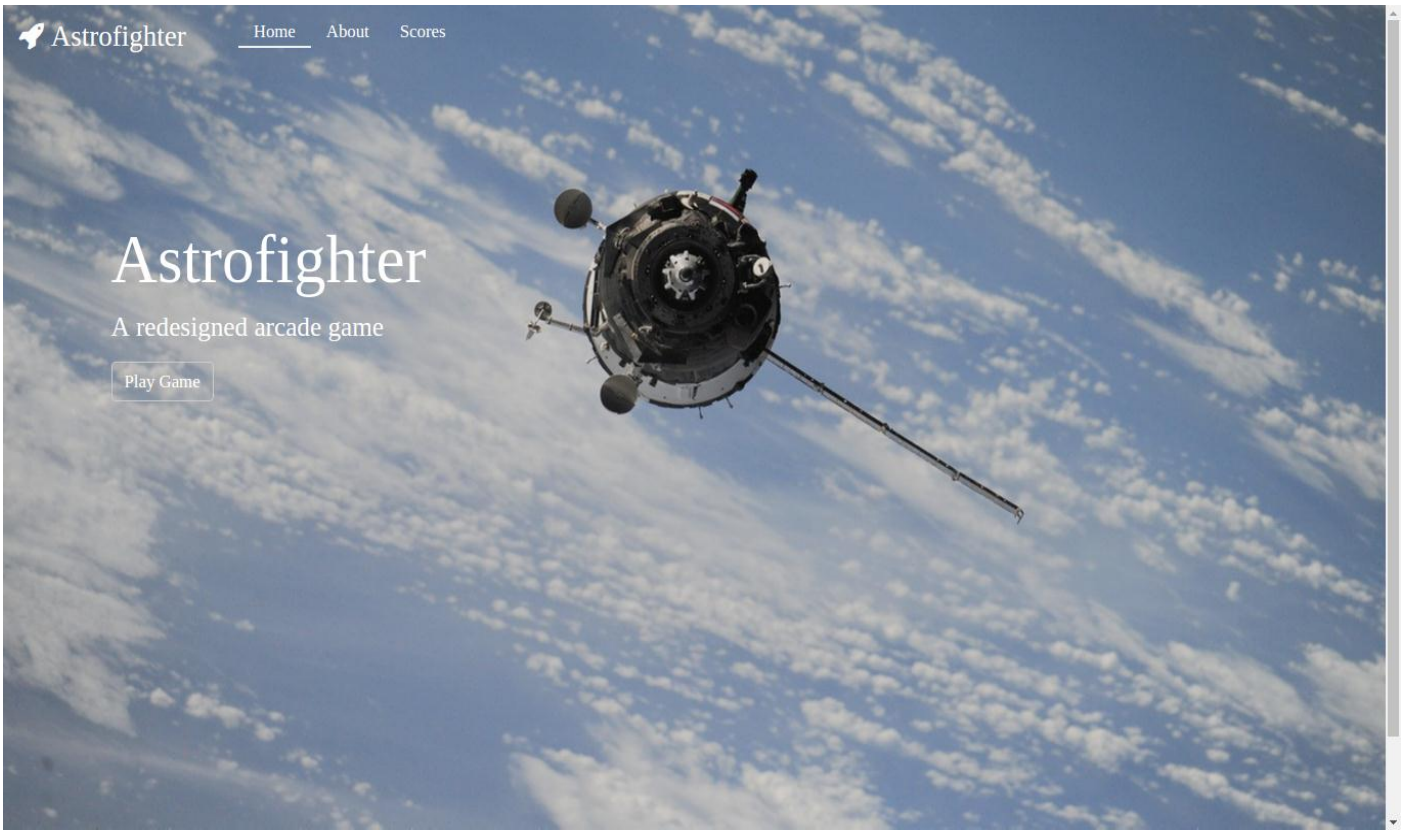
7. Implementation Details

```
astrofighter/
|-- config.ru
|-- development.db
|-- Gemfile
|-- Gemfile.lock
|-- LICENSE
|-- main.rb
|-- player.rb
|-- populate_db.rb
|-- public/
|   |-- audio/
|   |-- css/
|   |-- images/
|   |   `-- originals/
|   `-- js/
|-- README.md
|-- tasks.todo
`-- views/
```

7.1 Routes:

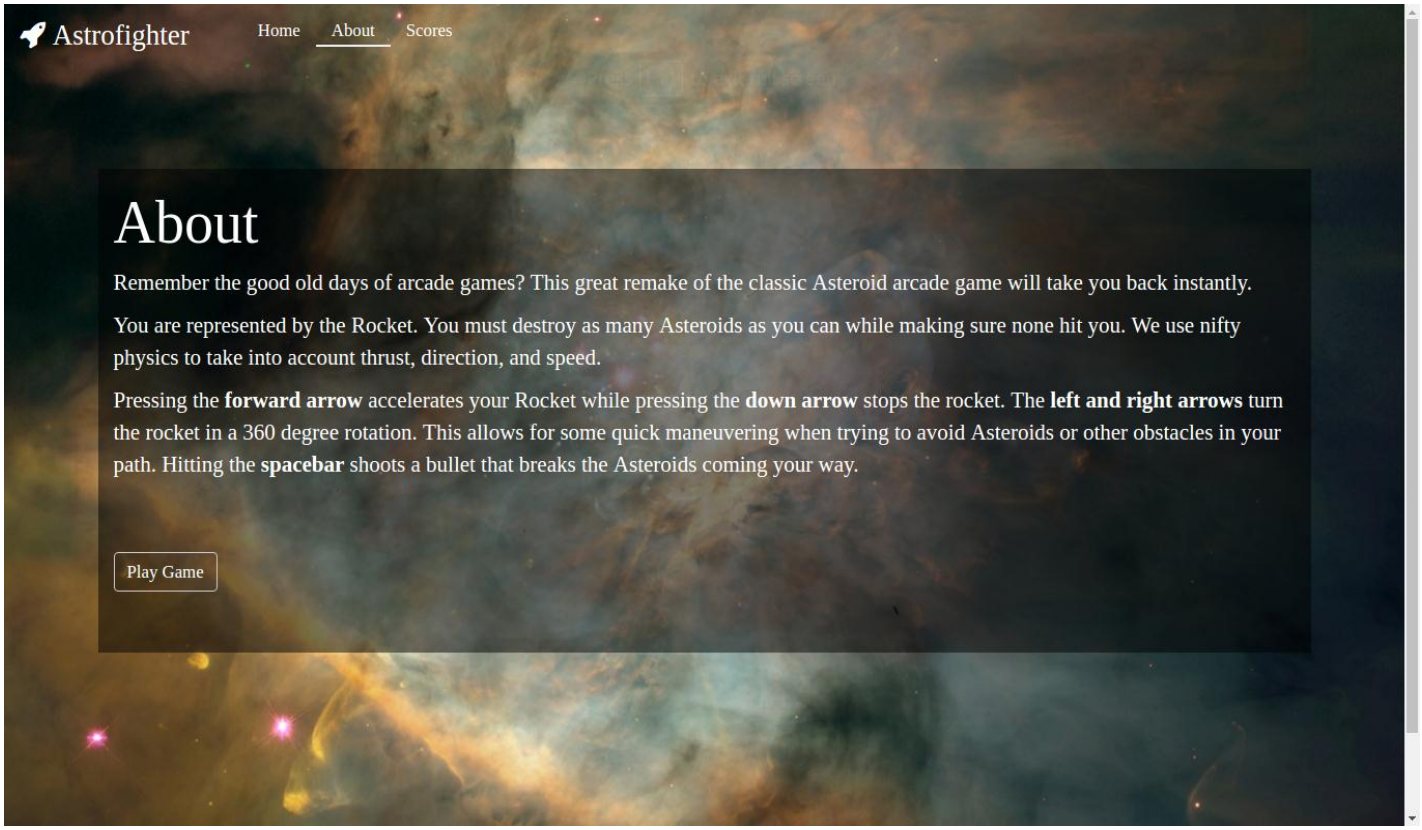
The game consists of various routes, each of which has a specific function.

- GET / - home page



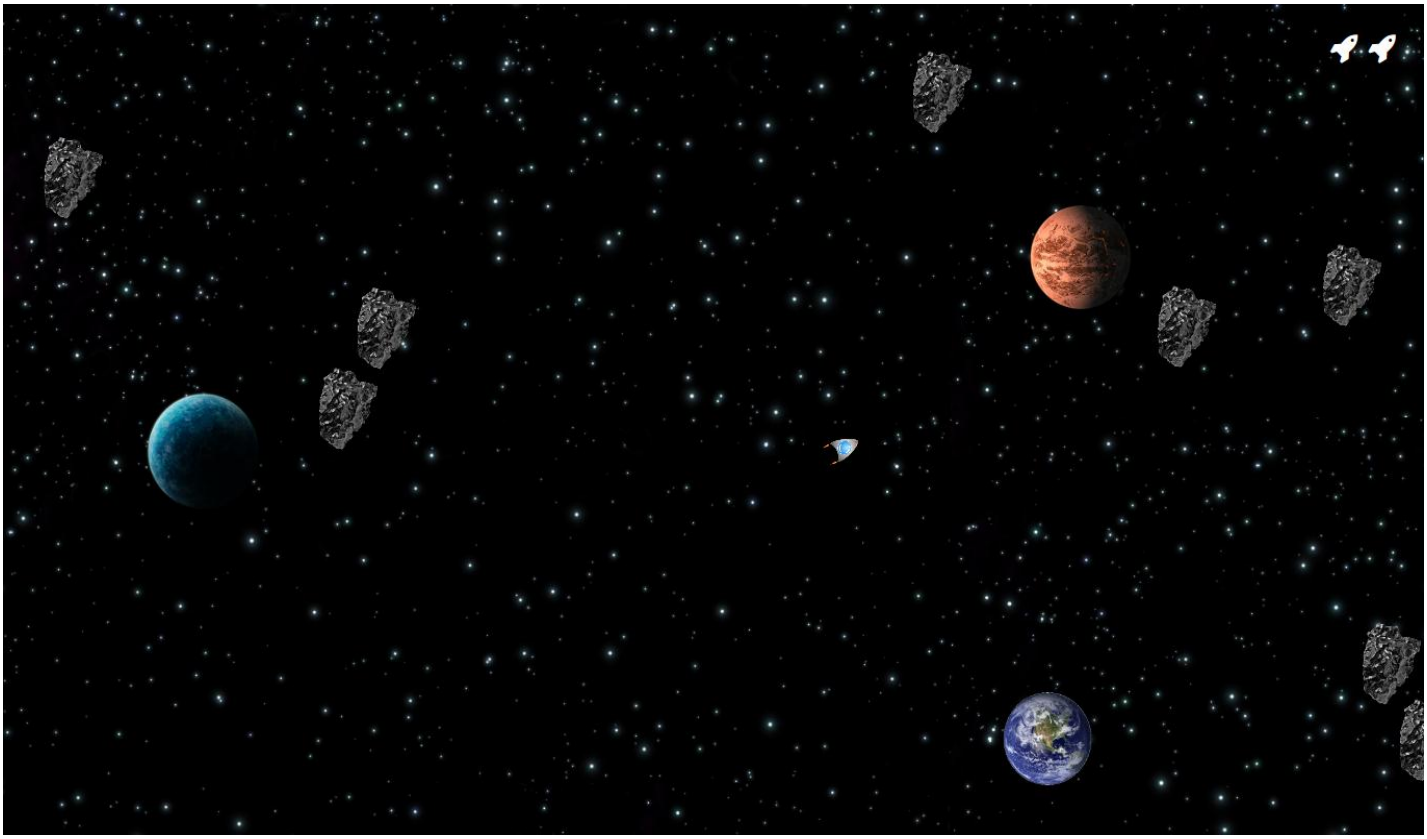
  - First page of the application.
  - The user can click on the *Play Game* button to head directly to the game.

- GET **/about –** About page



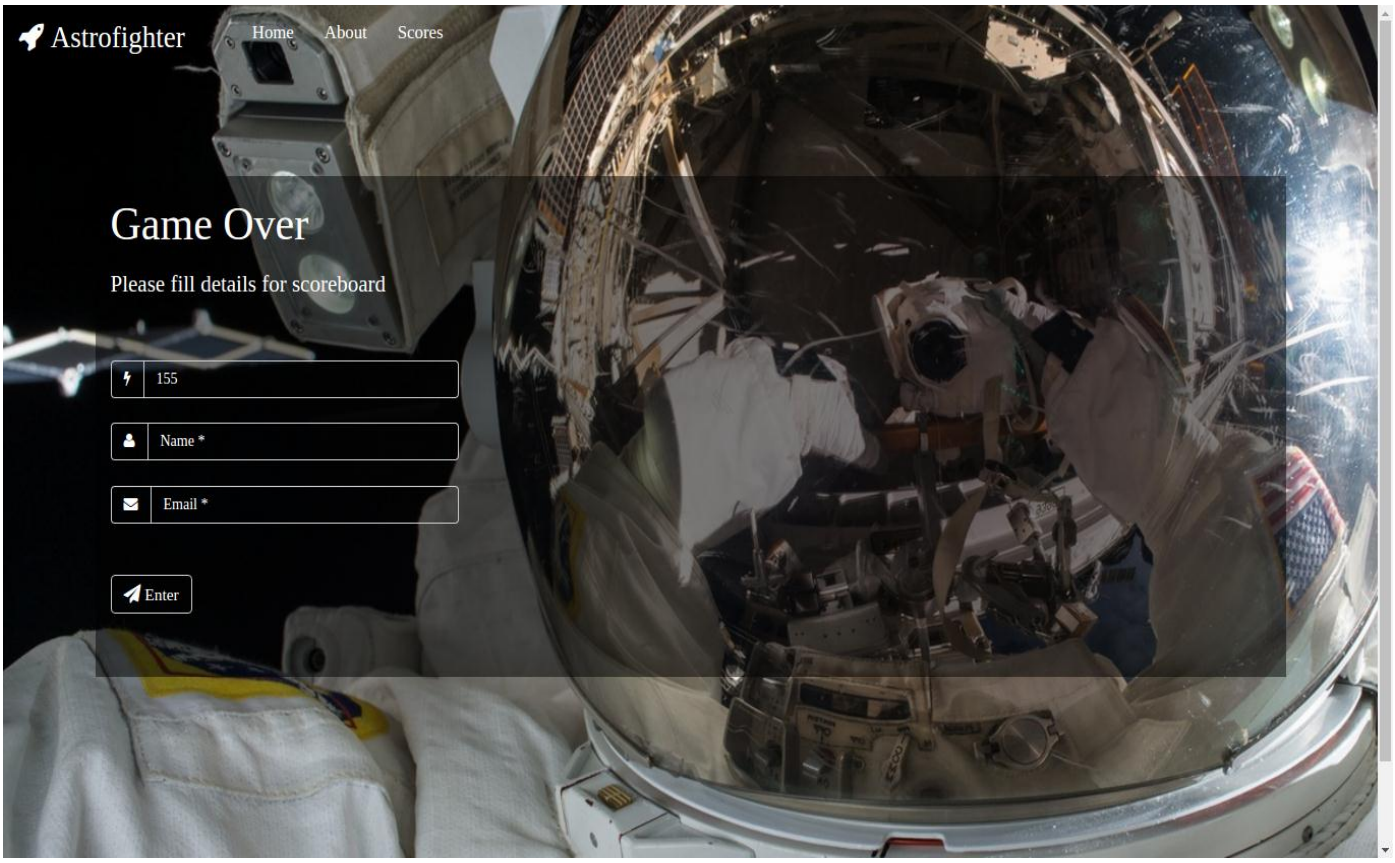
  - Provides some background to the game.
  - Provides the rules and controls required to play the game.

- GET **/game/?** - Game page



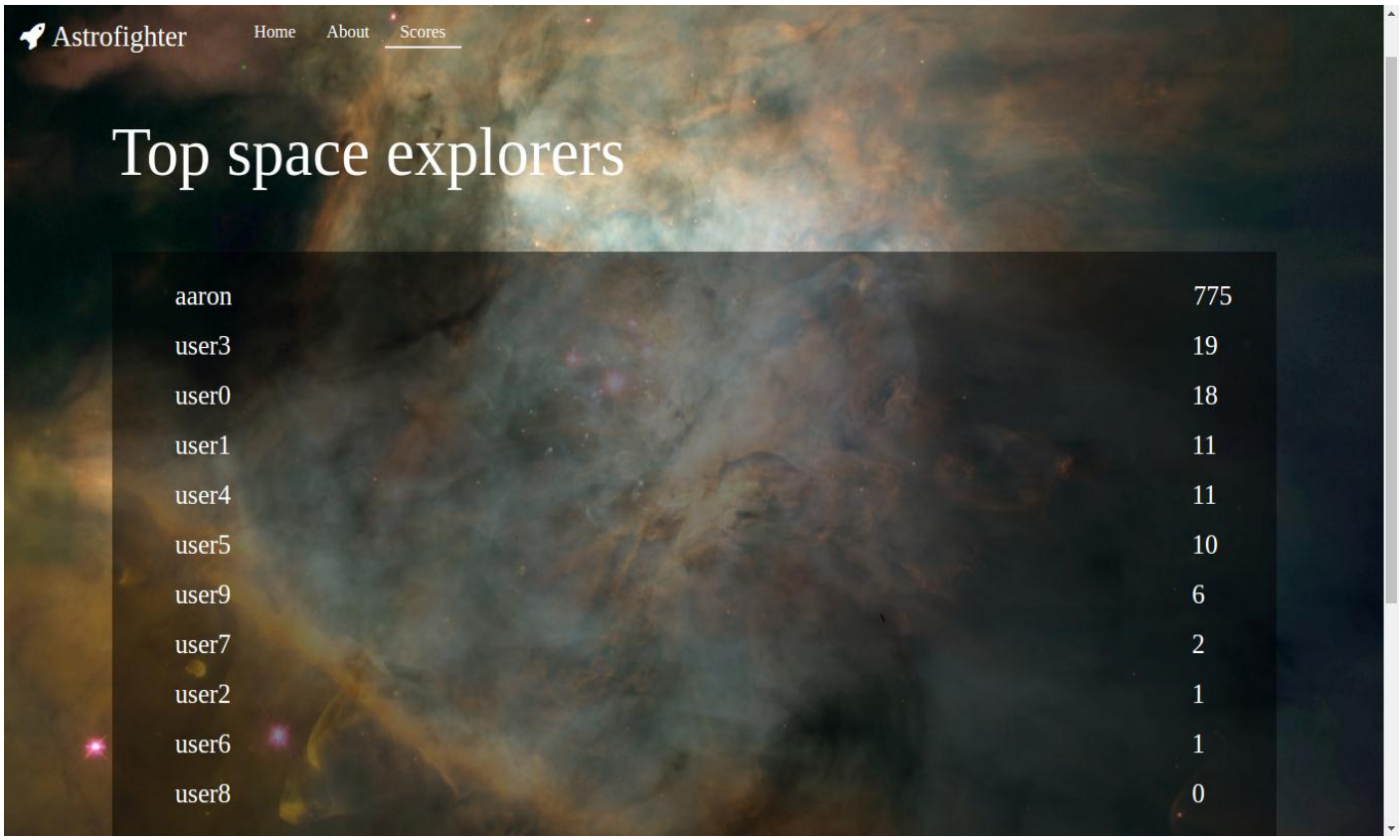
  - This is the route to actually play the game.
  - The number of lives remaining is displayed in the top right corner.

- GET **/gameover/?** - Game-over page. Enter player details



  - Player enters
    - Name
    - E-mail
  - The score cannot be edited by the player. It is generated based on the number of meteors destroyed.

- GET **/scores** - Scoreboard page

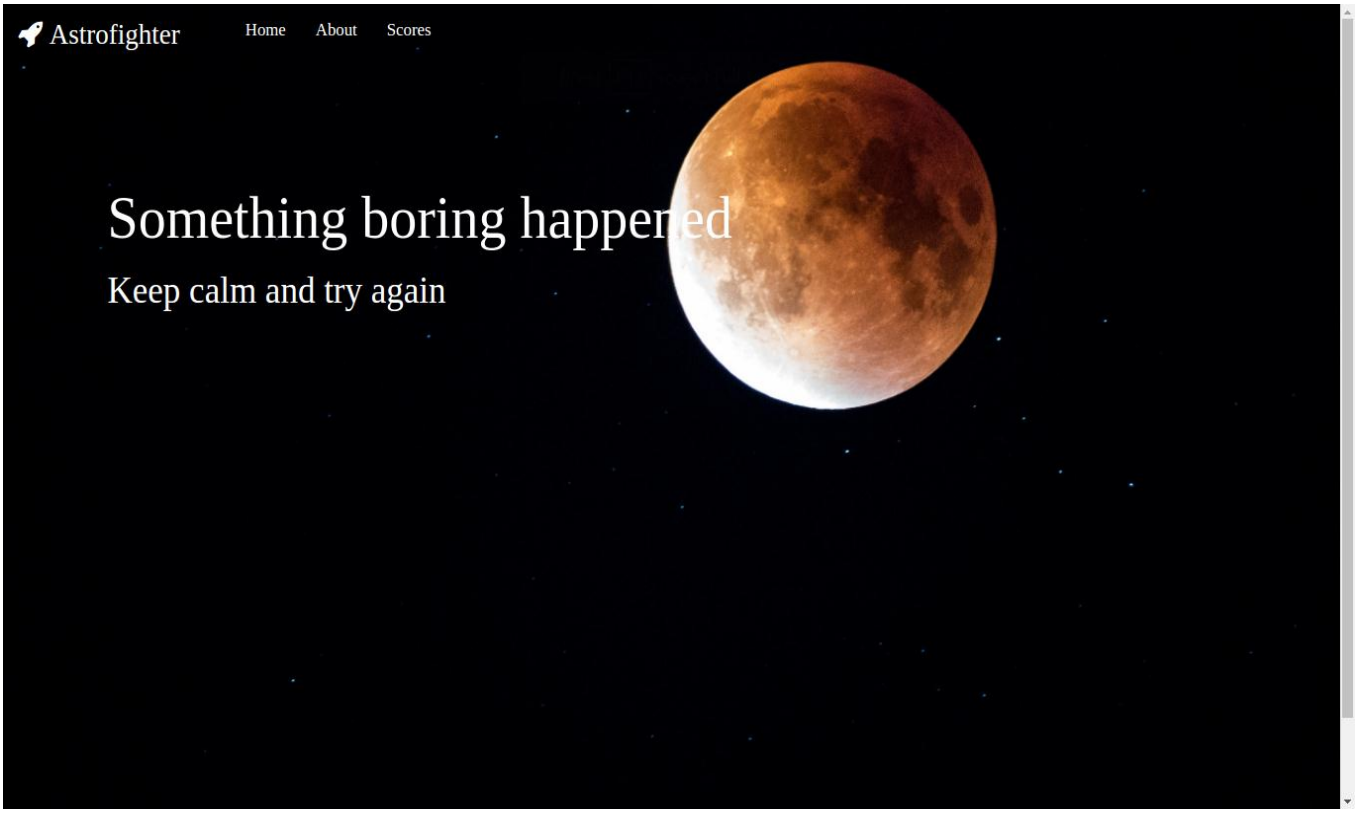


  - Displays the scores of all the players in descending order.

- **Not Found Page –** Broken Links

  - When a broken link or a resource that is not available is requested.

- **Server Error Page –** Unknown error on server side



  - When an unidentified server error occurs.

- POST **/players** - POST request route to update/add Player

7.2 Views

ERB (Embedded Ruby) is a feature of Ruby that enables you to conveniently generate any kind of text, in any quantity, from templates. The templates themselves combine plain text with Ruby code for variable substitution and flow control, which makes them easy to write and maintain.
ERB was used to generate the views for each of the routes.

```
views/
|-- about.erb
|-- error.erb
|-- game.erb
|-- game_over.erb
|-- home.erb
|-- layout.erb
|-- layout_head.erb
|-- nav.erb
|-- not_found.erb
`-- scores.erb
```

- **about.erb**

```
<link rel="stylesheet" type="text/css" href="/css/about.css">

<div class="about">
    <div class="container">
        <h1>About</h1>

        <p>Remember the good old days of arcade games? This great remake of the classic Asteroid
        arcade game will take you back instantly.</p>

        <p>
            You are represented by the Rocket. You must destroy as many Asteroids as you can
            while making sure none hit you. We use nifty physics to take into account
            thrust, direction, and speed.
        </p>

        <p>
            Pressing the <b>forward arrow</b> accelerates your Rocket while pressing the <b>down arrow</b>
            stops the rocket. The <b>left and right arrows</b> turn the rocket in a 360 degree rotation.
            This allows for some quick maneuvering when trying to avoid Asteroids or other obstacles
            in your path. Hitting the <b>spacebar</b> shoots a bullet that breaks the
            Asteroids coming your way.
        </p>

        <a href="/game" class="btn btn-default">Play Game</a>
    </div>
</div>
```

- **error.erb**

```
<link rel="stylesheet" type="text/css" href="/css/error.css">

<div class="error">
    <div class="container">
        <h1>Something boring happened</h1>
        <h2>Keep calm and try again</h2>
    </div>
</div>
```

- **layout.erb**

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <%= erb :layout_head %>
    </head>
    <body>
        <%= erb :nav %>

        <%= yield %>

        <div class="footer">
            <div class="container">
                <p>Created by <u><a href="http://aaroncolaco.com">Aaron</a></u>, <u><a href="http://aaroncolaco.com">
                Joston</a></u>, <u><a href="http://aaroncolaco.com">Kimberly</a></u>, <u><a href="http://aaroncolaco.
                com">Natasha</a></u>, <u><a href="http://aaroncolaco.com">Prajakta</a></u></p>
            </div>
        </div>

        <script type="text/javascript" src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.1/jquery.min.js"></
        script>
        <script async type="text/javascript" src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.5/js/bootstrap.min.js
        "></script>
    </body>
</html>
```

- **layout_head.erb**

```
<!--This makes sure the page renders properly on a mobile-->
<meta name="viewport" content="width=device-width, initial-scale=1.0, user-scalable=0">

<title><%= @title %></title>
<!-- <link rel="icon" type="image/png" href="/logo.png"> -->
<link rel="shortcut icon" type="image/png" href="/images/favicon.png"/>
<meta charset="UTF-8">
<meta name="description" content="Astrofighter - Arcade Game">
<meta name="keywords" content="game, physics engine, aaron, colaco, joston, kimberly, natasha, prajakta">

<meta name="author" content="Aaron">
<meta name="Distribution" content="Global">
<meta name="Rating" content="General">
<meta name="Robots" content="INDEX,FOLLOW">
<meta name="Revisit-after" content="31 Days">

<link rel="stylesheet" type="text/css" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.5/css/bootstrap.min.css">
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/font-awesome/4.4.0/css/font-awesome.min.css">
<link rel="stylesheet" type="text/css" href="/css/layout.css">
```

- **nav.erb**

```
<nav class="navbar navbar-default navbar-fixed-top" role="navigation">
    <div class="container-fluid">

        <div class="navbar-header">
            <button type="button" class="navbar-toggle collapsed" data-toggle="collapse" data-target="#bs
            -example-navbar-collapse-1">
            <span class="icon-bar"></span>
            <span class="icon-bar"></span>
            <span class="icon-bar"></span>
            </button>
            <a class="navbar-brand" href="/">
                <i class="fa fa-rocket"> <%= settings.app_name %></i>
            </a>
        </div>

        <div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1">
            <ul class="nav navbar-nav">
            <!-- current? is a helper method that returns
            'active' if route matches arg. nil otherwise.
            Done to show active link in navbar -->
                <li class= <%= current? %> >
                    <a href="/">Home</a>
                </li>
                <li class=<%= current?( "/about") %> >
                    <a href="/about">About</a>
                </li>
                <li class=<%= current?( "/scores") %> >
                    <a href="/scores">Scores</a>
                </li>
            </ul>
        </div>
    </div>
</nav>
```

- **game.erb**

```
<!DOCTYPE html>
<html>
<head>
    <title><%=@title %></title>
    <meta name="viewport" content="width=device-width, initial-scale=1.0, user-scalable=0">
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/font-awesome/4.4.0/css/font-awesome.min.css">
    <link rel="stylesheet" type="text/css" href="/css/main.css">
</head>
<body>
    <p class="hidden" id="lives"><%= @lives %></p>
    <p class="hidden" id="score"><%= @score %></p>

    <canvas id="game-canvas"></canvas>

    <!-- Display number of lives -->
    <div class="lives">
        <% @lives.to_i.times do %>
            <i class="fa fa-rocket" id="life-1"></i>
        <% end %>
    </div>

    <div class="hidden">
        <img id="rocket-normal" src="/images/rocket_normal.png"></img>
    </div>

    <div class="hidden">
        <img id="rocket-moving" src="/images/rocket_fired.png"></img>
    </div>

    <div class="hidden">
        <img id="null" src="/images/null.png"></img>
    </div>
```

```
    <div class="hidden">
        <img id="asteroid" src="/images/meteor.png"></img>
    </div>

    <div class="hidden">
        <img id="bullet" src="/images/bullet.png"></img>
    </div>

    <div class="celestial-bodies">
        <img id="planet-1" src="/images/blueplanet.png">
        <img id="planet-2" src="/images/planet.png">
        <img id="planet-3" src="/images/earth.png">
    </div>

    <script async type="text/javascript" src="/js/game.js"></script>
</body>
</html>
```

- **game_over.erb**

```
<link rel="stylesheet" type="text/css" href="/css/game_over.css">
<div class="game-over">
    <div class="container">
        <h1>Game Over</h1>
        <h3>Please fill details for scoreboard</h3>

        <form id="form" method="POST" action="/players" role="form" autocomplete="on">

            <div class="input-group">
                <span class="input-group-addon" id="score-sign">
                    <i class="fa fa-bolt"></i>
                </span>
                <input class="form-control" type="text" name="player[score]" id="player-score" value=<%= @score %>
                aria-describedby="score-sign" readonly="readonly" >
            </div>

            <div class="input-group">
                <span class="input-group-addon" id="name-sign">
                    <i class="fa fa-user"></i>
                </span>
                <input type="text" class="form-control" name="player[name]" id="player-name" required aria-
                describedby="name-sign" placeholder="Name *" autofocus >
            </div>

            <div class="input-group">
                <span class="input-group-addon" id="email-sign">
                    <i class="fa fa-envelope"></i>
                </span>
                <input type="email" class="form-control" name="player[email]" id="player-email" required aria-
                describedby="email-sign" placeholder="Email *" >
            </div>

            <button type="submit" class="btn btn-default" >
                <i class="fa fa-paper-plane"></i> Enter
            </button>
        </form>
    </div>
</div>
```

- **home.erb**

```
<link rel="stylesheet" type="text/css" href="/css/home.css">

<div class="jumbotron" id="home">
    <div class="container">
        <h1>
            <%= settings.app_name %>
        </h1>
        <p>A redesigned arcade game</p>
        <a href="/game" class="btn btn-default">Play Game</a>
    </div>
</div>
```

- **scores.erb**

```
<link rel="stylesheet" type="text/css" href="/css/scores.css">

<div class="scores" id="scores">
    <div class="container">
        <h1>
            Top space explorers
        </h1>
        <% if @player.any? %>
            <ul id="player">
                <% @player.each do |player| %>
                    <li>
                        <span class="player-name"><%= "#{player.name}" %></span>
                        <span class="player-score"><%= "#{player.score}" %></span>
                    </li>
                <% end %>
            </ul>
        <% else %>
            <p>Nobody has dared to explore space!</p>
        <% end %>

        <a href="/game" class="btn btn-default">Play Game</a>
    </div>
</div>
```

- **not_found.erb**

```
<link rel="stylesheet" type="text/css" href="/css/not_found.css">

<div class="not_found">
    <div class="container">
        <h1>Oops! Looks like a broken link!</h1>
    </div>
</div>
```

7.3 Styles

CSS3 is the latest evolution of the Cascading Style Sheets language.

The views are styled using CSS3. The files are:

```
public/css/
|-- about.css
|-- error.css
|-- game_over.css
|-- home.css
|-- layout.css
|-- main.css
|-- not_found.css
`-- scores.css
```

- **about.css**

```
.about {
    background-image:url('/images/space3.jpg');
    height: 100%;
    background-repeat: no-repeat;
    background-size: cover;
    background-attachment: fixed;
}

.about .container {
    position: relative;
    top: 20%;
    background: rgba(0, 0, 0, 0.5);
}

.about h1 {
    color: #ffffff;
    font-size: 4em;
}

.about p {
    color: #ffffff;
    font-size: 1.5em;
}

.about .btn-default {
    color: #ffffff;
    background-color: transparent;
    font-size: 1.2em;

    margin-top: 5%;
    margin-bottom: 5%;
}
.about .btn-default:hover {
    background-color: #5a5aad;
}
```

- **error.css**

```
.error {
    background-image:url('/images/space5.jpg');
    height: 100%;
    background-repeat: no-repeat;
    background-size: cover;
    background-attachment: fixed;
}

.error .container {
    position: relative;
    top:20%;
}

.error h1 {
    color: #ffffff;
    font-size: 4em;
}

.error h2 {
    color: #ffffff;
    font-size: 2.5em;
}
```

- **not_found.css**

```
.not_found {
    background-image:url('/images/space4.jpg');
    height: 100%;
    background-repeat: no-repeat;
    background-size: cover;
    background-attachment: fixed;
}

.not_found .container {
    position: relative;
    top:20%;
    background: rgba(0, 0, 0, 0.5);
}

.not_found h1 {
    color: #ffffff;
    font-size: 4.5em;

    text-align: center;
}
```

- **main.css**

```css
* {
    /* to remove the top and left whitespace */
    margin:0;
    padding:0;
}

html, body {
    /* just to be sure these are full screen*/
    width:100%;
    height:100%;
    background-color: #000000;
}

canvas {
    /* To remove the scrollbars */
    display:block;
    background-image: url('/images/background.png');
    background-repeat: no-repeat;
    background-size: cover;
}

.hidden {
    display: none;
}

.lives {
    position: absolute;
    top: 0;
    right: 0;

    margin-top: 2%;
    margin-right: 2%;
}
```

```
.fa-rocket {
    color: #ffffff;
    font-size: 1.8em;

    margin-left: 0.2em;
}


body .celestial-bodies #planet-1 {
    width: 8%;

    position: absolute;

    margin-top: -30%;
    margin-left: 10%;
}


body .celestial-bodies #planet-2 {
    width: 12%;

    position: absolute;

    margin-top: -45%;
    margin-left: 70%;
}

body .celestial-bodies #planet-3 {
    width: 7%;

    position: absolute;

    margin-top: -10%;
    margin-left: 70%;
}
```

- **game_over.css**

```
.game-over {
    background-image:url('/images/space2.jpg');
    height: 100%;
    background-repeat: no-repeat;
    background-size: cover;
    background-attachment: fixed;
}

.game-over .container {
    position: relative;
    top: 20%;
    background: rgba(0, 0, 0, 0.5);
}

.game-over h1 {
    color: #ffffff;
    font-size: 3em;
}

.game-over h3 {
    color: #ffffff;
    font-size: 1.5em;
}

.game-over #form {
    margin-top: 5%;
}

.game-over .input-group {
    width: 30%;
    margin-bottom: 2%;
}

.game-over .fa {
    color: #ffffff;
    font-size: 1em;
}
```

```
.game-over #form input, .game-over #form .input-group-addon {
    background-color: transparent;
    color: #ffffff;
    font-size: 1em;
}

.game-over #form .btn, .game-over #form .btn .fa {
    background-color: transparent;
    color: #ffffff;
    font-size: 1.1em;

    margin-top: 2%;
    margin-bottom: 5%;
}

.form-control::-webkit-input-placeholder { color: white; font-size: 1em; }
.form-control:-moz-placeholder { color: white; font-size: 1em; }
.form-control::-moz-placeholder { color: white; font-size: 1em; }
.form-control:-ms-input-placeholder { color: white; font-size: 1em; }
```

- **scores.css**

```
.scores {
    background-image:url('/images/space3.jpg');
    min-height: 100%;
    background-repeat: no-repeat;
    background-size: cover;
    margin-bottom: 0%;
    background-attachment: fixed;
}

.scores .container {
    position: relative;
    padding-top: 10%;
}

.scores h1 {
    color: #ffffff;
    font-size: 4.5em;
}

.scores ul {
    margin-top: 5%;
    margin-bottom: 5%;
    padding-top: 2%;
    padding-bottom: 2%;

    list-style-type: none;

    background: rgba(0, 0, 0, 0.5);
}

.scores li {
    color: #ffffff;
    font-size: 1.8em;
    margin-bottom: 1%;
}
```

```
.scores {
    background-image:url('/images/space3.jpg');
    min-height: 100%;
    background-repeat: no-repeat;
    background-size: cover;
    margin-bottom: 0%;
    background-attachment: fixed;
}

.scores .container {
    position: relative;
    padding-top: 10%;
}

.scores h1 {
    color: #ffffff;
    font-size: 4.5em;
}

.scores ul {
    margin-top: 5%;
    margin-bottom: 5%;
    padding-top: 2%;
    padding-bottom: 2%;

    list-style-type: none;

    background: rgba(0, 0, 0, 0.5);
}

.scores li {
    color: #ffffff;
    font-size: 1.8em;
    margin-bottom: 1%;
}
```

- **home.css**

```
.jumbotron {
    background-image:url('/images/space1.jpg');
    min-height: 100%;
    background-repeat: no-repeat;
    background-size: cover;
    margin-bottom: 0%;
}

.jumbotron .container {
    position: relative;
    padding-top: 10%;
}

.jumbotron h1 {
    color: #ffffff;
    font-size: 4.5em;
}

.jumbotron p {
    color: #ffffff;
    font-size: 1.8em;
}

.jumbotron .btn-default {
    color: #ffffff;
    background-color: transparent;
    font-size: 1.2em;
}
.jumbotron .btn-default:hover {
    background-color: #5a5aad;
}
```

- **layout.css**

```
html {
    height: 100%;   /*So that the 100% divs are full height*/
}

body {
    height: 100%;   /*So that the 100% divs are full height*/
}

* {
    font-family: Verdana;
}

.fa {
    color: #ffffff;
    font-size: 1.5em;
}


.navbar {
    width: 100%;
    z-index: 1;

    background-color: transparent;
    border-color: transparent;
}

.navbar-brand {
    margin-right: 2em;
}

.navbar > .navbar-collapse {
    text-align: center;
}
```

```
.nav.navbar-nav li a {
    color: #ffffff;
    font-size: 1.2em;
}
.nav.navbar-nav li a:hover {
    color: #ffffff;
    border-bottom: 1px solid #efefef;
    padding-bottom: 4px;
}

.navbar-default .navbar-collapse .navbar-nav > .active a {
    color: #ffffff;
    background-color: transparent;
    border-bottom: 2px solid #efefef;
    padding-bottom: 3px;
}

.navbar-btn {
    color: #ffffff;
    background-color: transparent;
    font-size: 1.2em;
    margin-right: 2em;
}

.footer {
    background-color: #222222;
    min-height: 10%;
}

.footer .container {
    padding-top: 2%;
    padding-bottom: 2%;
}
```

7.4 Player Class:

The player class is defined as:

```
class Player
        include DataMapper::Resource
        property :id, Serial
        property :email, String
        property :name, String
        property :score, Integer
end
```

7.5 Audio

Audio files are stored in /public/audio/

7.6 Images

Image files are stored in /public/images/

7.7 Core Game

7.7.1 Objects

- Rocket Object

```
var rocket = {
    spriteNormal: null,
    spriteMoving: null,
    x: 0,
    y: 0,
    width: 20,
    height: 30,
    speed: 0,
    angle: 90,
    move: false,
    accelerate: false,
    rotating: false,
    clkwise: true,
    draw : null,
    up: null,
    down: null,
    left: null,
    right: null,
    update: null
};
```

- Meteor Object

```
var Meteor = function(meteorSize) {
    this.sprite = document.getElementById('asteroid');
    this.x = Math.random() * global.width;
    this.y = Math.random() * global.height;
    this.width = meteorSize;
    this.height = meteorSize;
    this.speed = 0;
    this.xmovement = Math.random() * 2;
    this.ymovement = Math.random() * 2;
    this.angle = 30;
    this.rotating = false;
    this.clkwise = true;
};
```

- Bullet Object

```
var Bullet = function() {
    this.sprite = document.getElementById('bullet');
    this.x = rocket.x + 1 ;
    this.y = rocket.y + 6.5 ;
    this.width = 5 ;
    this.height = 5 ;
    this.speed = 7 ;
    this.angle = rocket.angle;
    this.hit = false;
};
```

7.7.2 Functions

- Initialization Function

  It is the first function called. It initializes all the data we need.

```
function init () {
    canvas = document.getElementById('game-canvas');
    context = canvas.getContext('2d');
    global.width = canvas.width = window.innerWidth;
    global.height = canvas.height = window.innerHeight;
    global.left = 37, global.up = 38, global.right = 39, global.down = 40;
    global.space = 32;

    rocket.x = (rocket.width + global.width)/2;
    rocket.y = (rocket.height + global.height)/2;

    rocket.spriteNormal = document.getElementById('rocket-normal');
    rocket.spriteMoving = document.getElementById('rocket-moving');

    for (let i = 0; i < global.initialMeteorNumber; i++) {
        meteor.push(new Meteor(global.meteorSizeLarge));
    };

    document.addEventListener('keydown', keydown);
    document.addEventListener('keyup', keyup);

    console.log( "Lives left: " + global.lives);
    console.log( "Score: " + global.score);
};
```

- Step Function

  It is called 65 times a second by requestAnimationFrame. It calls global update and draw functions followed by a recursive call to itself. This form of recursion is called tail call optimization and ensures linear complexity without a stack.

```
var step = function (timestamp) {
    update();
    draw();
    window.requestAnimationFrame(step);
};
window.requestAnimationFrame(step);
```

- Global Update Function

  Calls the update function of each object.

```
function update () {
    rocket.update();

    for (let met of meteor) {
        met.update();
    };

    for(let shot of bullet) {
        shot.update();
    };

};
```

- Global Draw Function

  Calls the draw function of each object.

```
function draw () {
    // clear full context to redraw new stuff
    context.clearRect(0, 0, global.width, global.height);
    rocket.draw();

    for (let met of meteor) {
        met.draw();
    };

    for (let shot of bullet) {
        shot.draw()
    };
};
```

- Meteor – Draw & Update Functions

```
Meteor.prototype.draw = function() {
    context.drawImage(this.sprite, this.x, this.y, this.width, this.height);
};

Meteor.prototype.update = function() {

    this.x += this.xmovement;
    this.y += this.ymovement;
    //to wrap the asteroids
    if (this.x > global.width) {
        this.x = 0;
    } else if (this.x < 0) {
        this.x = global.width;
    } else if (this.y > global.height) {
        this.y = 0;
    } else if (this.y < 0) {
        this.y = global.height;
    };

    /*Collision detection with rocket*/
    if ( (Math.abs(this.x - rocket.x) < 30) && (Math.abs(this.y - rocket.y) < 30) ) {
        rocketCrash();
    };
};
```

Each time the update function is called:

- It first increments the *x* & *y* co-ordinates of the meteor to show 'movement'.
- Then the co-ordinate positions are checked to determine if the meteor is moving out of context.
- If it is, the co-ordinates are updated so that the meteor wraps around and re-enters the context.
- Then it checks if a collision between the current meteor and the rocket has occurred.
- If collision has occurred it calls the *rocketCrash* function to decrement the number of lives and restart the game at initial position.
- The draw function draws the meteor sprite at the new updated position.

- Bullet – Draw & Update Functions (folded code)

```
Bullet.prototype.draw = function() {
    context.drawImage(this.sprite, this.x, this.y, this.width, this.height);
};

Bullet.prototype.update = function() {

    this.x += Math.sin(this.angle*(Math.PI/180))*this.speed;
    this.y += Math.cos(this.angle*(Math.PI/180))*this.speed*-1;

    /*Collision detection between bullet & meteor*/
        for (let met of meteor) {
        /*if 'in' is used, met => index no. With 'of', met => values(object)*/
            if ( (Math.abs(this.x - met.x) < 25) && (Math.abs(this.y - met.y) < 25) && (this.hit === false) ) {
                explosionAudio.play(); // meteor break-up audio

                // Get hit meteor's metadata
                let meteorSize = met.width;

                // Bullet & Meteor vanish
                this.sprite = document.getElementById('null');
                met.sprite = document.getElementById('null');

                // Remove bullet & meteor from their arrays
                bullet.splice(bullet.indexOf(this), 1);
                meteor.splice(meteor.indexOf(met), 1);

                // Break Meteor based on size
                switch(meteorSize) {...
                }

                function addMeteors (newMeteorSize) {...
                };
                // Making sure one bullet's hit is registered only once
                this.hit = true;
            };
        };
};
```

Each time the update function is called:

- It first updates the co-ordinates to emulate movement of the bullet.
- It checks if the bullet has collided with any of the meteors.
- If it has, the meteor is replaced
  - Large meteor is replaced with Medium meteor.
  - Medium meteor is replaced with Small meteor.
  - Small meteors vanish when shot.
- The bullet vanishes after successful shot to ensure multiple hits are not registered.

- Rocket- Draw Function

```
rocket.draw = function() {
    context.translate(rocket.x, rocket.y);
    context.rotate(rocket.angle*(Math.PI/180)); /*Rotate context to angle required*/

    if (rocket.move) {
        context.drawImage(rocket.spriteMoving, 0, 0, rocket.width, rocket.height+4);
    } else {
        context.drawImage(rocket.spriteNormal, 0, 0, rocket.width, rocket.height);
    };

    context.rotate(rocket.angle*(Math.PI/180) * -1);
    context.translate(-rocket.x, -rocket.y);
};
```

  - The context is translated to the co-ordinates of the rocket. This places the context's origin at the center of the rocket.
  - The context is rotated to the angle of the rocket - in relation to the origin of the context.
  - The rocket is drawn onto the context.
  - The context is then rotated back to the normal orientation and translated back to its original position.

void Context.drawImage(image, sx, sy, sWidth, sHeight, dx, dy, dWidth, dHeight);

- ***dx* & *dy* :** The X & Y coordinates in the destination canvas at which to place the top-left corner of the source `image`.
- **dWidth** & **dHeight :** The width and height to draw the image in the destination canvas. This allows scaling of the drawn image. If not specified, the image is not scaled in width when drawn.
- **sx** & **sy :** The X & Y coordinates of the top left corner of the sub-rectangle of the source image to draw into the destination context.
- **sWidth :** The width of the sub-rectangle of the source image to draw into the destination context. If not specified, the entire rectangle from the coordinates specified by sx and sy to the bottom-right corner of the image is used.
- **sHeight :** The height of the sub-rectangle of the source image to draw into the destination context.

- Rocket – Update Function

```
rocket.update = function() {

    if (rocket.move) {
        rocket.x += Math.sin(rocket.angle*(Math.PI/180))*rocket.speed;
        rocket.y += Math.cos(rocket.angle*(Math.PI/180))*rocket.speed*-1;

        //place outside this 'if' to make it turn even when not moving
        if (rocket.rotating) {
            if (rocket.clkwise) {
                rocket.angle += 5;
            } else {
                rocket.angle -= 5;
            }
        };
    } else {
        rocket.speed = 0;
    };

    if (rocket.accelerate) {
        rocket.speed = 5;
    } else {
        rocket.speed *= friction;
    };

    /*Due to friction speed drops. When it goes below treshhold, move = false
        so that sprite image is changed*/
    if (rocket.speed <= 0.3) {
        rocket.move = false;
    };

    //rocket wrap around
    if (rocket.x > global.width) {
        rocket.x = 0;
    } else if (rocket.x < 0) {
        rocket.x = global.width;
    } else if (rocket.y > global.height) {
        rocket.y = 0;
    } else if (rocket.y < 0) {
        rocket.y = global.height;
    };
};
```

Each time the update function is called:

- The co-ordinates are updated to show motion of the rocket.
- The angle of rotation of the rocket is adjusted based on flag set by key-press listeners.
- The acceleration and speed is adjusted based on flags set by key-press listeners.
- Rocket wrap around is performed by checking co-ordinates *wrt* the context.

- Key Listener Functions

  Listen for key presses to perform actions.

```
function keydown (event) {
    let key = event.keyCode;

    switch (key) {
        case global.left:
            rocket.left('pressed');
            break;
        case global.up:
            rocket.up('pressed');
            break;
        case global.right:
            rocket.right('pressed');
            break;
        case global.down:
            rocket.down();
            break;
        case global.space:
            shotAudio.play(); // bullet fired audio
            bullet.push(new Bullet());
            break;
    }
};
```

```
function keyup (event) {
    let key = event.keyCode;

    switch (key) {
        case global.left:
            rocket.left('released');
            break;
        case global.up:
            rocket.up('released');
            break;
        case global.right:
            rocket.right('released');
            break;
    }
};
```

7.7.3 Global Data

```
var context = null, canvas, global = {};
var friction = 0.95;
var meteor = [], bullet = [];
var explosionAudio = new Audio('/audio/explosion.mp3');
var shotAudio = new Audio('/audio/shot.mp3');


global.initialMeteorNumber = 8;
global.meteorSizeLarge = 110;
global.meteorSizeMedium = 80;
global.meteorSizeSmall = 50;
// Points for breaking meteors
global.smallMeteorPoints = 5;
global.mediumMeteorPoints = 10;
global.largeMeteorPoints = 15;

global.lives = document.getElementById('lives').innerHTML;
global.score = parseInt(document.getElementById('score').innerHTML);
```

7.7.4 Integrated Code

All the functions and objects integrate to form a single script.

- The code has been folded to ensure brevity.

```
function loaded () {

    "use strict"    // So 'let' and other stuff can be used

    var context = null, canvas, global = {};
    var friction = 0.95;
    var meteor = [], bullet = [];
    var explosionAudio = new Audio('/audio/explosion.mp3');
    var shotAudio = new Audio('/audio/shot.mp3');


    global.initialMeteorNumber = 8;
    global.meteorSizeLarge = 110;
    global.meteorSizeMedium = 80;
    global.meteorSizeSmall = 50;
    // Points for breaking meteors
    global.smallMeteorPoints = 5;
    global.mediumMeteorPoints = 10;
    global.largeMeteorPoints = 15;

    global.lives = document.getElementById('lives').innerHTML;
    global.score = parseInt(document.getElementById('score').innerHTML);

    var rocket = {...
    };

    var Meteor = function(meteorSize) {...
    };

    var Bullet = function() {...
    };

    Bullet.prototype.draw = function() {
        context.drawImage(this.sprite, this.x, this.y, this.width, this.height);
    };

    Bullet.prototype.update = function() {...
    };
```

```
Meteor.prototype.draw = function() {
    context.drawImage(this.sprite, this.x, this.y, this.width, this.height);
};

Meteor.prototype.update = function() {
};

rocket.up = function(status) {
    if (status == "pressed") {
        rocket.move = true;
        rocket.accelerate = true;
    } else {
        rocket.accelerate = false;
    };
};

rocket.down = function() {
    rocket.accelerate = false;
    rocket.move = false;
};

rocket.left = function(status) {
    if (status == "pressed") {
        rocket.rotating = true;
        rocket.clkwise = false;
    } else{
        rocket.rotating = false;
    };
};

rocket.right = function(status) {
    if (status == "pressed") {
        rocket.rotating = true;
        rocket.clkwise = true;
    } else{
        rocket.rotating = false;
    };
};
```

```
    rocket.draw = function() {
    };

    rocket.update = function() {
    };

    function keydown (event) {
    };

    function keyup (event) {
    };

    function init () {
    };

    function update () {

    };

    function draw () {
    };

    init();

    var step = function (timestamp) {
    };
    window.requestAnimationFrame(step);

    function rocketCrash () {
        if (global.lives > 1) {
            global.lives = global.lives - 1;
            window.location = "/game?lives=" + global.lives + "&score=" + global.score;
            throw new Error("Thrown to stop script from continuing executing");
        } else if (global.lives == 1 || global.lives < 1) {
            window.location = "/gameover?score=" + global.score;
            throw new Error("Thrown to stop script from continuing executing");
        };
    };
};

window.onload = loaded;
```

7.9 Server Code

- Run the file as ***ruby main.rb***
- It is advisable to run ***bundle install*** - to install any missing gems - before running the file.

```ruby
require 'sinatra'
require './player'

# Reload if in development mode
# require 'sinatra/reloader' if development?

set :app_name, 'Astrofighter'
set :lives, 3   # Number of lives in the game
set :score , 0

# for development
configure :development do
    DataMapper.setup(:default, "sqlite3://#{Dir.pwd}/development.db")

    disable :show_exceptions
end

# DB for production
configure :production do
    DataMapper.setup(:default, ENV['DATABASE_URL'])
end


before do
    set_title
end

get '/' do
    erb :home
end

get '/about' do
    erb :about
end
```

```ruby
get '/game/?' do    # for /game & /game/
    @title = "Asteroid Game"
    @lives = params[:lives] || settings.lives # 3 lives at start of game
    @score = params[:score] || settings.score

    erb :game, :layout => nil   # So that default layout not used
end

get '/gameover/?' do
    @title = "Game Over"
    @score = params[:score] || 0

    erb :game_over
end

get '/scores' do
    @player = Player.all(:order => [ :score.desc ]) # Scores in descending order
    erb :scores
end

post '/players?' do
    player = params[:player] # From form submission

    # Get/Check player data using helper method
    player_id = get_player_id(player[:email])

    if player_id.nil?
        Player.create(player)
    else
        Player.get(player_id).update(player)
    end

    redirect to ('/scores')
end
```

```ruby
# All errors
error do
    @title = "Error"
    erb :error
end

# Not Found error
not_found do
    @title = "Not Found"
    erb :not_found
end

helpers do
    def current?(path = '/')
        (request.path == path || request.path == path+'/') ? "active" : nil
    end
    def set_title
        # if not given, set to app name. That is why the OR
        @title ||= settings.app_name
        # Done so that we don't have to put this login in the erb file like before
    end
    def get_player_id(email)
        player = Player.first(:email => email)
        if player.nil?
            return nil
        end
        return player.id
    end
end
```

7.9 Database Script

```ruby
# Run with bundle exec ruby populate_db.rb

require "./player.rb"

no_of_players = 10

# Clear db
Player.all.destroy

# Populate db
no_of_players.times do |var|
    player=Player.new
    player.email = "user#{var}@gmail.com"
    player.name = "user#{var}"
    player.score = rand(20)
    player.save
end

puts "Done"
```

- Run this script using ***bundle exec populate_db.rb*** from the root directory of the project.
- This will initialize the database with 10 players.
- Set the ***no_of_players*** variable to change the number of players the DB is populated with.

7.10 Gemfile

A Gemfile is a file we create which is used for describing gem dependencies for Ruby programs. A gem is a collection of Ruby code that we can extract into a “collection” which we can call later.

```
source "https://rubygems.org"
ruby '2.2.1'
gem "sinatra"
gem "slim"
gem "sass"
gem "dm-core"
gem "dm-migrations"
gem "thin"
gem "sinatra-flash"

gem "pg", :group => :production
gem "dm-postgres-adapter", :group => :production

gem "dm-sqlite-adapter", :group => :development
```

8. License