

THE MAHARAJA SAYAJIRAO UNIVERSITY OF BARODA

FACULTY OF SCIENCE



A PROJECT REPORT

On

Planet Guardian

Submitted by

Vrushali Gaurang Patel

Seat No: 300022

In Partial fulfillment for the award of the degree

Of

BACHELOR OF COMPUTER APPLICATIONS

In

The Department of Computer Applications

December, 2018

Guide by:

Mrs. Poonam Yadav

Team members:

Parth Piyush Patel – 300021

Bansi Amish Shah – 300025

Table of Contents

Abstract	2
Acknowledgement	3
Project charter.....	4
❖ Project Definition	4
❖ Purpose.....	5
❖ Scope of Project	6
Project Plan	7
❖ Objective.....	7
❖ Software Model-Traditional (Prototype Model).....	8
❖ Work division (w.r.t time)	9
❖ Work division (w.r.t team members).....	10
Hardware and Software Requirements	11
❖ Hardware Requirements:.....	11
❖ Software Requirements:	11
Tools and Technologies.....	12
System Analysis	13
❖ State Diagram	13
System Design	16
❖ Interface Design (Pen Sketch)	16
❖ Screenshots	20
Testing	24
❖ Testing Plan:	25
❖ Testing Methods:	26
❖ Test Scenarios:	27
❖ Bug fixing /Test report	29
Coding Standard Followed	30
Limitations and Future Enhancements.....	31
Experience and Learning.....	32
Bibliography.....	33

Abstract

In the last few years, there has been a growing interest in the application of the game elements to the real-life goals and tasks.

Planet Guardian is a Desktop/Android Application Game that could be played only by a single-player.

Planet Guardian is a game that involves three major objects namely:

- 1) A Planet that revolves around its own axis
- 2) Meteorites that targets the Planet to destroy it
- 3) A Spaceship that revolves around the planet into its orbit to shoot the meteorites coming towards the planet.

The aim of the player is to save the Planet from being destroyed by the Meteor-shower. The meteorite could be destroyed by firing the bullets through the spaceship.

Score will be calculated, according to the survival time.

Planet has its health bar. It will be destroyed until five meteor fall on it and game would be over at this stage.

Acknowledgement

Planet Guardian Game (dummy project) is created using Prototype methodology. It has been a great experience for our team. Practical application of the studies as well as exploring new tools and technologies like Unity3D, Autodesk Maya, Autodesk 3dsMax and Adobe Photoshop was undertaken. This knowledge will help us forever.

The results received from this mini project (M.Sc.IT. Semester III) are due to the active guidance and assistance received from many individuals. We are grateful towards them. We could never have successfully completed our work if they hadn't contributed their part. We are thankful to **Prof. R.S. Srivastava** (Head, Department of Computer Application) for providing us with this platform and encouragement.

Very special thank you to our project guide **Mrs. Poonam Yadav** who portrayed extreme efforts towards providing appropriate guidance at all times.

We also want to thank other faculties also for helping us in the doubts aroused during project documentation.

We would also like to extend our heartfelt thanks to the staff of Department of Computer Applications helped us in completing our project.

Project charter

Project Definition

Planet Guardian is a Desktop/Android Application Game.

Planet Guardian is a single-player game. The aim of the player is to save the Planet from being destroyed by the Meteor-shower. The meteorite could be destroyed by firing the bullets through the spaceship.

The game also enables a user to browse through the rules of the game, and allows the player to play the game whenever interested also he could pause/play and quit accordingly.

Purpose

In the last few years, there has been a growing interest in the application of the game elements to the real-life goals and tasks. Game would allow the user to get a break from the tedious day to day life and could provide entertainment in the leisure time.

The main purpose of developing a game was the keenness to learn new software like Unity and gain experience from it.

Scope of Project

- A Planet that rotates on its own axis
- A Spaceship that revolves around the Planet
- A Meteor generation at random position.
- Spawning of Meteor
- The Planet pulls the Meteorite towards it
- The revolving Spaceship that fires a bullet on click
- Play/Pause controls (on pressing “P” key and play/pause button)
- Giving particular graphics and texture to each object
- Score will be calculated, according to the survival time.
- Displaying Highest score
- Destroying Meteorite on contact with bullet
- Generating a Health-Bar for a Planet
- Destroying Planet on 5 Meteorite fall (Game Over)
- Applying media (sound)

Project Plan

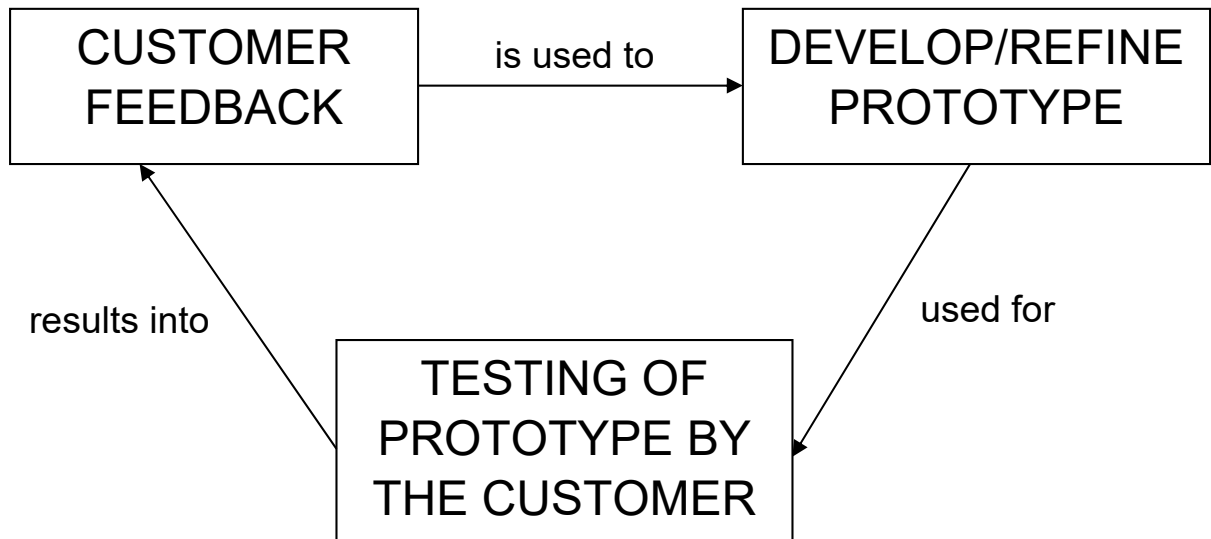
Objective

The world is moving towards a digital life and due to hard-working life, people don't get much time for entertainment and so aim is to provide a little fun and diverting environment in the leisure time to get rid from monotonous work.

There aroused a requirement of a system for amusement from the daily chore.

Planet Guardian game creates a hypothetical situation to save the planet from being destroyed by shooting the meteorites which is easy to play.

Software Model-Traditional (Prototype Model)



Reason for Selecting the Model:

- Increased user involvement in the product even before its implementation.
- Since a working model of the system is displayed, the users get a better understanding of the system being developed.
- Reduces time and cost as the defects can be detected much earlier.
- Quicker user feedback is available leading to better solutions.
- Missing functionality can be identified easily.
- Confusing or difficult functions can be identified.
- The customers get to see the partial product early in the life cycle. This ensures a greater level of customer satisfaction and comfort.
- New requirements can be easily accommodated as there is scope for refinement.
- Missing functionalities can be easily figured out.
- Flexibility in design.

Work division (w.r.t time)

Month	Work Division
September	Defining the Scope Deciding frameworks and software Designing and Modeling the objects Planet Rotation Spaceship Rotation Single Meteorite Generation Meteor pull towards the Planet Firing Bullet from the spaceship
October	Destroying Meteorite on contact with Planet Random generation and Spawning of Meteorites. Meteorite Rotation Designing the Home-Screen Designing the End-Screen Adding specific controls
November	Destroying Meteorite on contact with Bullet Generating Health-bar Calculating the score and high-score
December	Adding Effects Applying specific Media Developing for Android Platform

Note: Integration was done timely along with error correction.

Work division (w.r.t team members)

Member Name	Work Distribution
Parth Patel	Designing and Modeling the objects Setting camera-controls and lighting Firing Bullet from the spaceship Spawning of Meteorites Designing the Home-Screen Generating Health-bar Adding Effects Appling specific Media Developing for Android Platform
Vrushali Patel	Planet Rotation Spaceship Revolution Meteorite Rotation Designing the End-Screen Adding specific controls Calculating the score and high-score Generating Health-bar Adding Effects Applying specific Media Developing for Android Platform
Bansi Shah	Random Meteorite Generation Meteor pull towards the Planet Destroying Meteorite on contact with Planet Spawning of Meteorites Destroying Meteorite on contact with Bullet Calculating the score and high-score Adding Effects Appling specific Media Developing for Android Platform

Hardware and Software Requirements

Hardware Requirements:

Personal Computer/ Laptop	Mobile
I/O port	Touch Screen
Minimum 512MB RAM	Minimum 100MB Storage
Processor (Intel Dual core 1.3GHz and above)	Processor
Minimum 100MB Storage	Minimum 512MB RAM

Software Requirements:

Personal Computer/ Laptop	Mobile
Windows (XP SP2 and above)	Android Operating System (5.0 and above)
Sound Driver	
Display Driver	

Tools and Technologies

Tools	Technologies
Unity3D 2018.2	C#
Adobe Photoshop CS6	
Adobe 3Ds Max 15	

System Analysis

State Diagram

Main Menu State Diagram

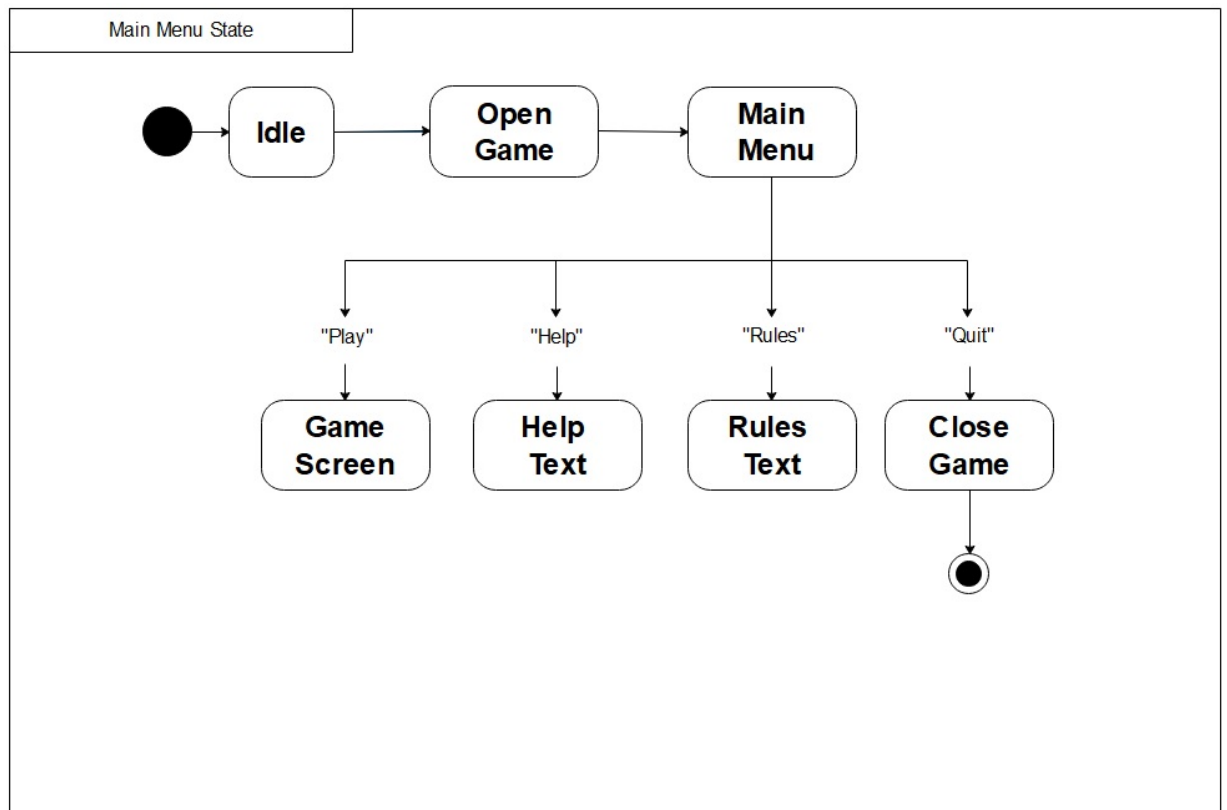


Figure 1.1 Main Menu State

Play Level State Diagram

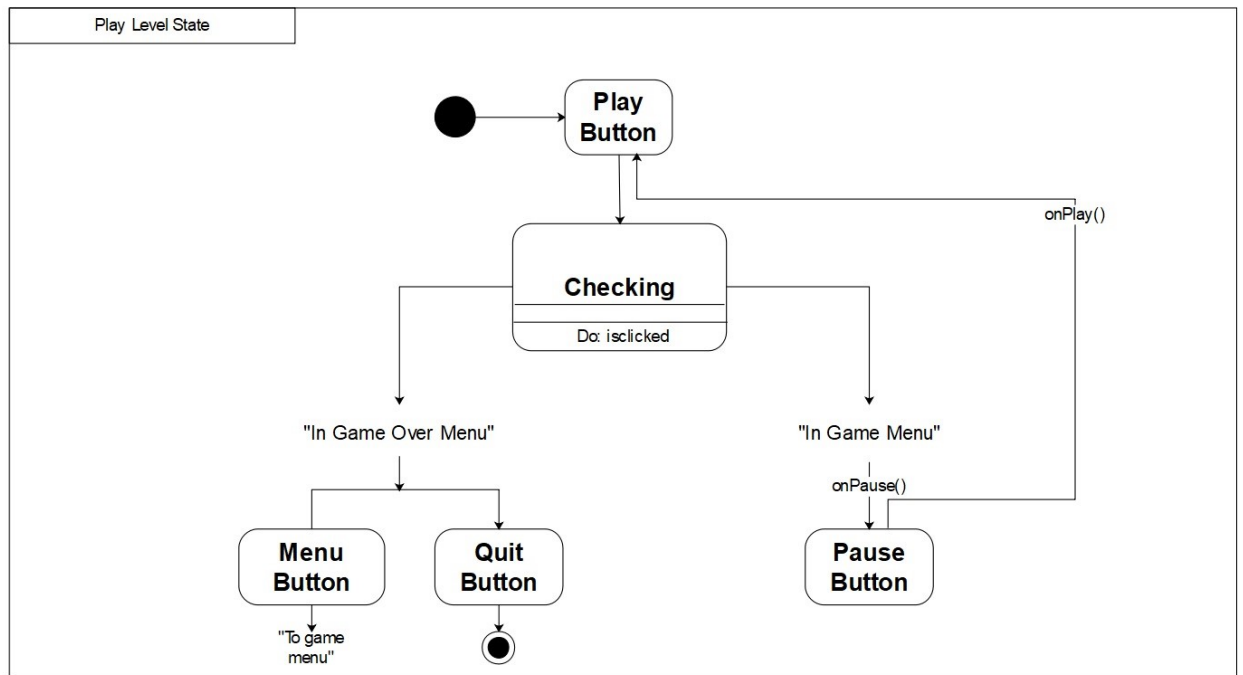


Figure 1.2 Play Level State

Planet Guardian System Flow Diagram

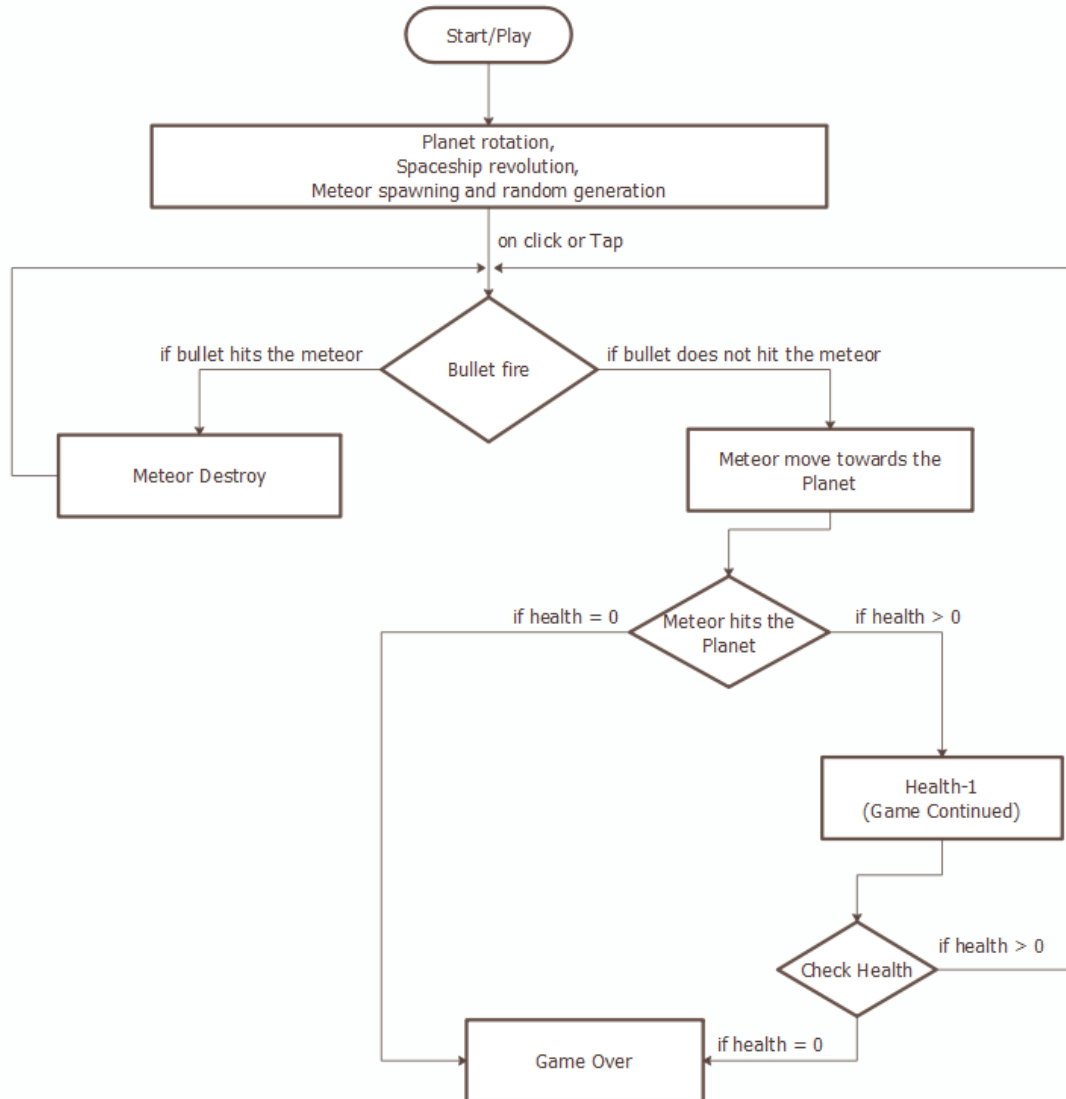


Figure 1.3 Planet Guardian System Flow Diagram

System Design

Interface Design (Pen Sketch)

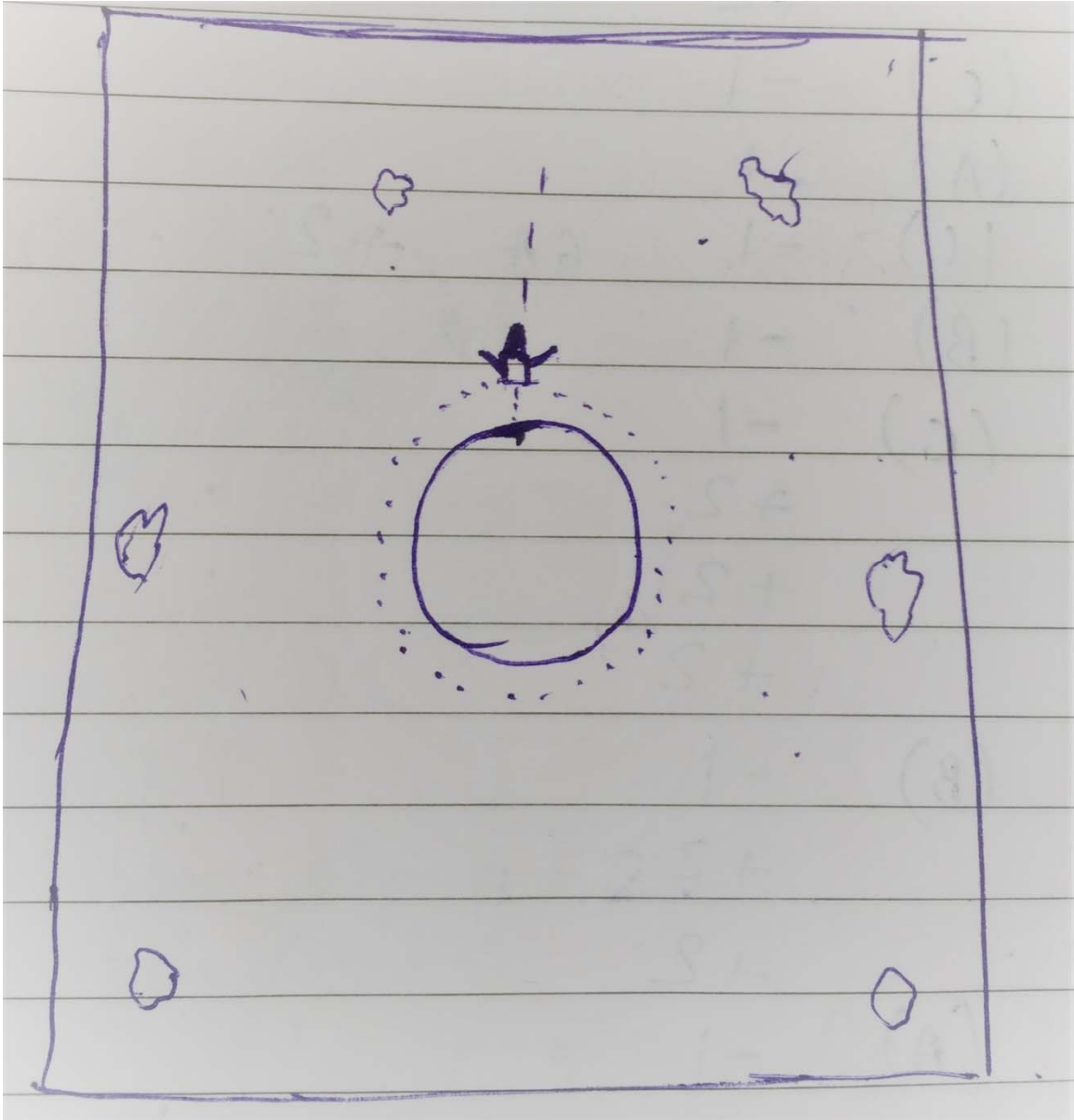


Figure 2.1 Interface Design

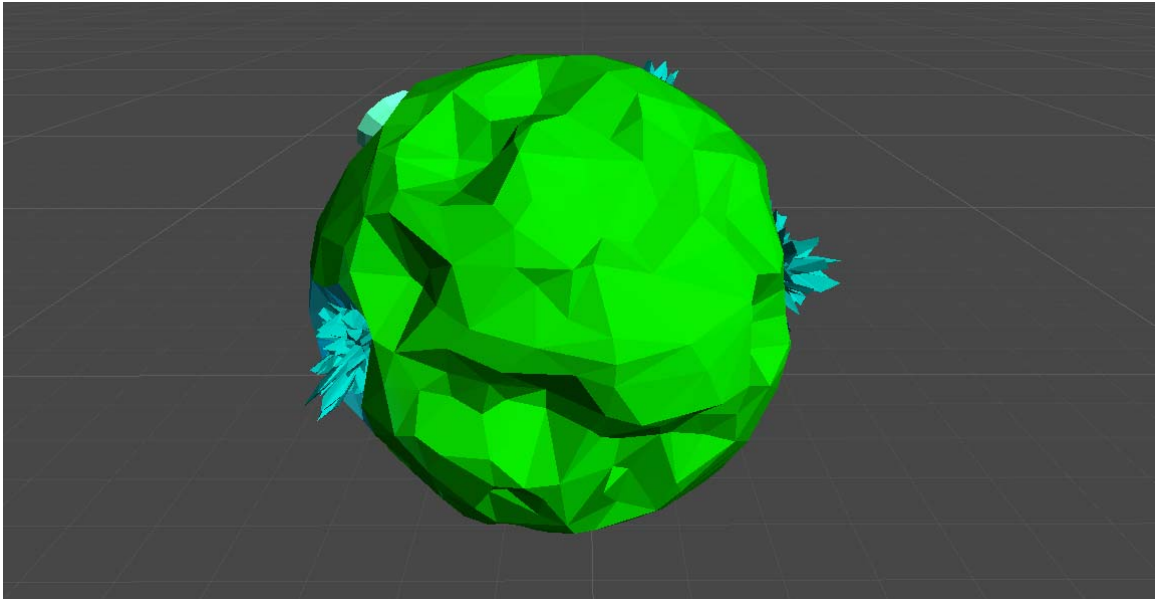


Figure 2.2 Old Planet



Figure 2.3 New Planet



Figure 2.4 Old Space Ship

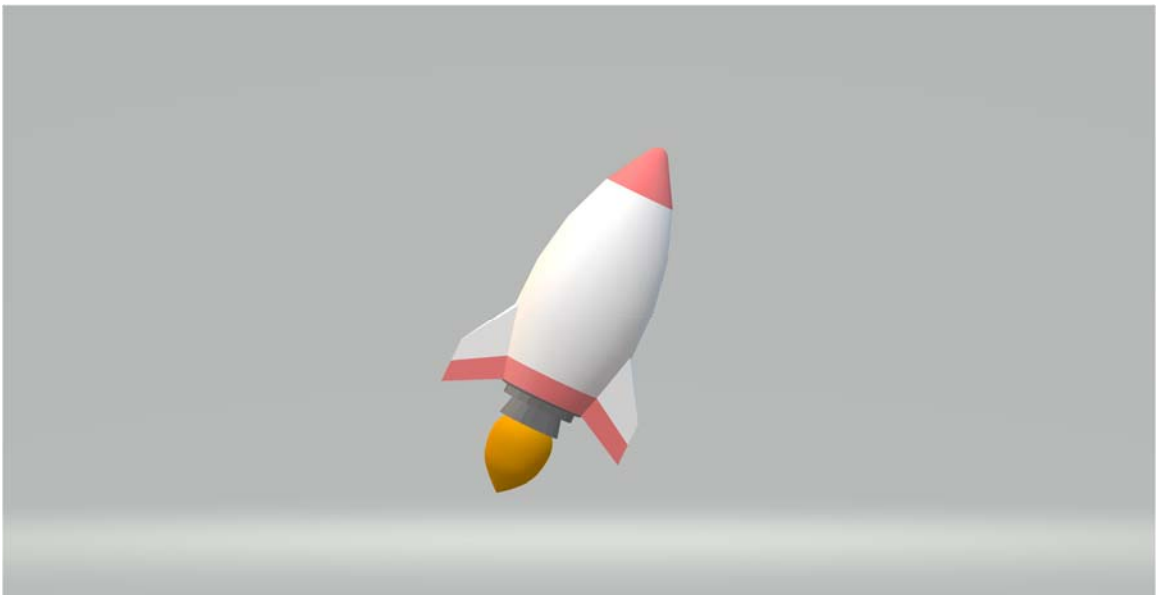


Figure 2.5 New Space Ship



Figure 2.6 Meteorite

Screenshots



Figure 3.1 Home Screen



Figure 3.2 Help Button

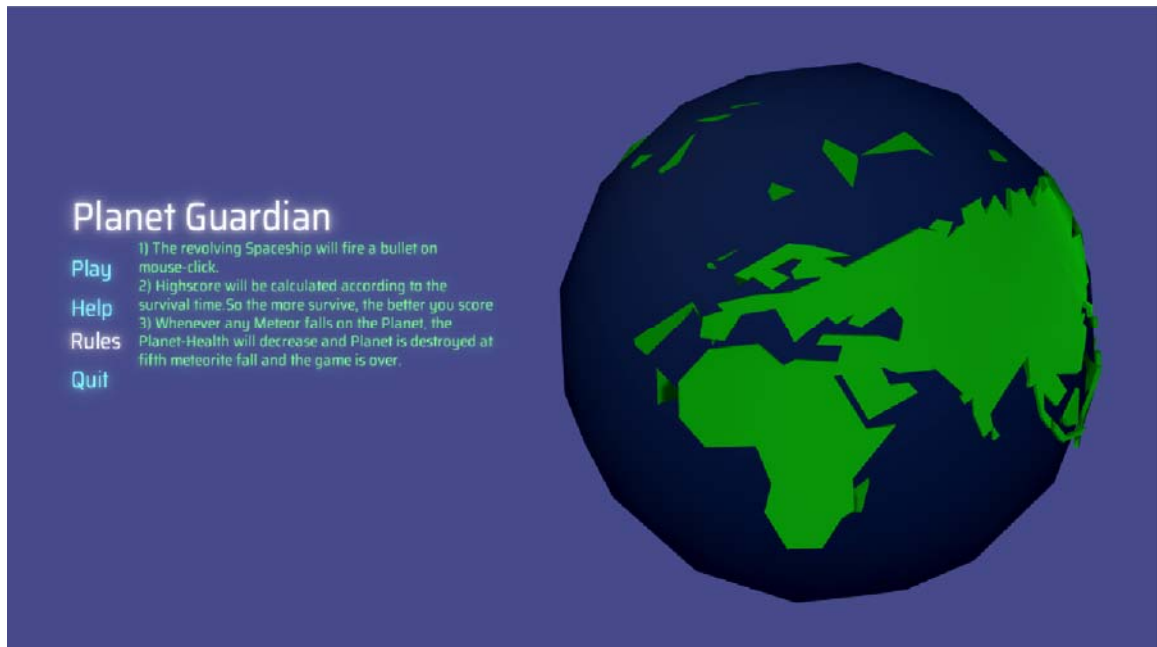


Figure 3.3 Rules Button



Figure 3.4 Game Screen



Figure 3.5 Bullet Fire



Figure 3.6 Destroying Meteor



Figure 3.7 Planet Heath



Figure 3.8 Game over Screen



Figure 3.9 Mobile View of Game

Testing

Testing Plan:

Testing is the process of evaluating a system or its component(s) with the intent to find whether it satisfies the specified requirements or not.

Test planning, the most important activity to ensure that there is initially a list of tasks and milestones in a baseline plan to track the progress of the project. It also defines the size of the test effort.

Testing Methods:

Testing methodologies are the strategies and approaches used to test a particular product to ensure it is fit for purpose. Software testing methodologies are the different approaches and ways of ensuring that a software application in particular is fully tested. Software testing methodology such as Test Scenarios was prepared so as to improve product quality.

Test Scenarios:

Some of the test scenarios are listed below:

Sr. No.	Scenarios	Comments
1	Planet Rotation on its own axis	Perfect at once
2	Spaceship revolution around the Planet	Problem was the distance between the Planet and the spaceship The Spaceship was revolving at different position Also the spaceship was revolving statically
3	Spawning of Meteorites (Multiple Generation)	This stage was the toughest. It took many hours of research and hard work to make it possible as our need.
4	A Meteor generation at random position.	Meteor were generating on x and y axis. As well as the problem was they were generating also on the Z-axis, which was not required.
5	Planet that pulls the Meteorite towards it	Perfect at once
6	The revolving Spaceship that fires a bullet on click	Initially, size of the bullet was very small Eventually the bullets were modified, but then they were continuously generated.
7	Giving particular graphics and texture to each object	Initially the models were different than now
8	Home Screen	Perfect at once
9	Main Screen	Camera control from home screen to main screen made some issues
10	End Screen	Spaceship didn't stop firing
11	Implementing Health bar of planet	Designing was perfect but working wasn't at initial stage
12	Implementing Play/Pause controls	The time frame at once didn't stop

13	Time Elapsed Display	Perfect at once
14	Destroying Planet	Had many issues as the meteor just got stuck at the center
15	Destroying prefabs(Meteor, Bullet)	Perfect at once

Bug fixing /Test report

The main principal of Software testing of an application is to find important bugs in the software application & try to make software application bug free.

Once a bug is found, we should know conditions under which this bug is occurring, how many time it occurs & the expected result of the bug. The bug report should accurate & complete, so that we can get the exact failure reason. Based on this we can get exact idea of problem faced by user & it helps to resolve the problem accurately.

After rectifying the bugs, it has to be solved and system was re-tested until the bugs were fixed.

Some of the bug fixing issues are listed below:

- Spaceship revolution around the Planet
- Spawning of Meteorites (Multiple Generation)
- A Meteor generation at random position
- The revolving Spaceship that fires a bullet on click
- Giving particular graphics and texture to each object
- Main Screen Camera control
- End Screen spaceship fire and meteor spawning
- Implementing Health bar of planet
- Implementing Play/Pause controls
- Bullet size was very small initially
- Continuous bullet shower on click
- Health bar decrease on contact of meteorite with planet
- High Score storage issue

Coding Standard Followed

- Different scripts are made for proper separation of the codes
- Using appropriate conditions/ functions/ methods
- Comments have been included where ever necessary
- Built-in functions are used where needed
- Declaring objects and giving reference
- Required objects are referred to scripts

Limitations and Future Enhancements

- Defining levels
- Changing Planets for every new game
- Changing Spaceship “avatar” according to users choice
- Adding different types of meteorites and other Enemies making the level tough
- Adding power-ups like Boost speed, Decrease speed, Freeze, Shield and many more.
- IOS Application Development

Experience and Learning

We still remember the day, when we decided to develop Planet Guardian game project. We had faced many difficulties. But thanks to teachers and colleagues, who helped us in every way overcome the problems, this project has given us immense knowledge and valuable experience throughout. We as team members had greater understanding and coordination that made us stronger and hence there was also a social bonding with the project. Even though we were following a new technology the team took it as a challenge and worked very smoothly.

Throughout the project there were ups and downs but overcoming them was our aim and hence finally we got a fruitful result. During the span of time of this project we learned many lessons like developing convincing power and realized working in a team with feeling of togetherness is so important in such projects.

We are all very satisfied working as a team, undergoing the challenges and facing them with the right spirit. We have learned to be positive on our way and to never give up. This project has given us opportunity to enhance our development, project management and communication skills.

Bibliography

<https://unity.com/education>

https://www.youtube.com/channel/UCYbK_tjZ2OrIZFBvU6CCMiA

<https://stackoverflow.com/>

<https://gamedev.stackexchange.com>

<https://www.c-sharpcorner.com/>