GAME ENGINE

A final project submitted to the University of Wales in partial fulfilment of the requirements of BA (Hons) Digital Media in Game Development

June 2013

Planet Shift

Pavankumar Chopra

2nd Year UG – Game Development

11UG03040

ICAT, Design and Media College,

Bangalore

Table of Contents

High Concept

One Liner

Trapped on an alien planet called Zing, Jeff has to make his escape from the planets magnetic shifts in order to escape from the planet and avoid getting caught by the aliens chasing after him. To do this he needs to reach the south of the planet were the magnetic force is low giving him the ability to burst of and leave the planet.

Core Tenets

Planet shift is all about the planets magnetic pulls and shifts. Jeff uses an alien space jet having the ability to take off from the planet but cannot do so due to the planets magnetic field.

- Magnetic disruptor
- Planet shifts
- Speed boost
- After burners

Game Genre

This is an Adventure/Arcade game.

Re-Playability

The games re-playability depends on the player. He can play the game again if he or she isn't satisfied with their score and wants to get a higher score in-order to beat their friends score. A player would also want to play the game again if she or he likes the UI, the games audio or even the visual appeal of the game. The player can collect coins along the way and unlock other space jets and use them in game. All these options give the player a reason to play again and thus making the game re-playable.

Target Audience

The game is targeted to players of the age 6 and above. The games adventurous, challenging and is having a very cartoonish appeal. The game is a lot fun and interesting and challenges players to try getting a top score.

Game Play

Jeff found him-self trapped on a planet called Zing. He crashed into the planet due to the planets strong magnetic pulls. After finding himself surrounded by alien space ships and crafts. He got into one of the jets which alerted the aliens. Jeff now has to make his way out of the planet buy following the long paths around the planet which leads to the south of the planet. But Jeff doesn't realize that there is a catch to the planet. Over time the planet shifts every time Jeff completes one rotation on a path. Shifting him to the next path. As Jeff gets closer to the south the space craft gets faster, which is because there is less magnetic pull on the south of the planet. Jeff has to make sure he doesn't crash into the obstacles that come forth to him or he will be captured by the aliens chasing him. Along the way there are coins that Jeff can collect in-order to purchase upgrades or a newer Space craft. There are also other power-ups along the way to help Jeff, like a magnetic disruptor, After-burners and other bonuses.

The player playing Jeff can collect the coins along the way in order to add up with the time taken to complete one path and then add up to give the players score.

If the player hits into obstacles he will lose the game.

Game Mechanics

Here consists the abilities, movement for Jeff's Space craft.

Movement:

Jeff will be using an alien space craft in-order to help him move along the planet and escape from it. He can acquire power ups to help him boost his movement along the planet.

Sway: Jeff can make the craft sway left or right to avoid in-coming obstacles.

Roll: Jeff can roll the craft left or right in-order to take sharp or big hard turns.

Traits

Magnetic disruptors: cause disruption with the planets magnetic field allowing the space craft to move faster.

After-Burners: After- burners build up thrust giving the space craft more speed even though the magnetic force is slowing the crafts speed. Double tapping the screen activates this power up after acquiring it.

Controls

Actions	Keys
Sway Left	Tilt Left
Sway Right	Tilt Right
Up	Tilt Down
Down	Tilt Up
Roll Right	Swipe Right
Roll Left	Swipe Left
After-Burners	Double tap Screen

Back Story

Jeff was on a space adventure when suddenly his space ships control systems were disrupted and went out of controlled. He tried a lot to get the ships systems back online or he would crash into a planet. He found him-self getting pulled towards an unknown alien planet which seems to be made up of metal. After try so much to get his ship back running he crashed onto the planet and found him-self surrounded by alien space crafts. There was an alert that around the area and aliens came to inspect the crash site. Jeff ran towards an alien space craft and hid there. Once he got in and started the craft an alert was set out which alarmed all the aliens around that area. Jeff was so afraid that he just wanted to get out of the planet so he tried to start the craft. It was his luck that he managed to control the space craft and could finally escape the planet. But the aliens came after him in their mini space crafts to capture Jeff. On the Guide map looked like an exit sign which showed the south pole of the planet. Jeff then just decided to follow the path till he reached the South Pole.

Camera

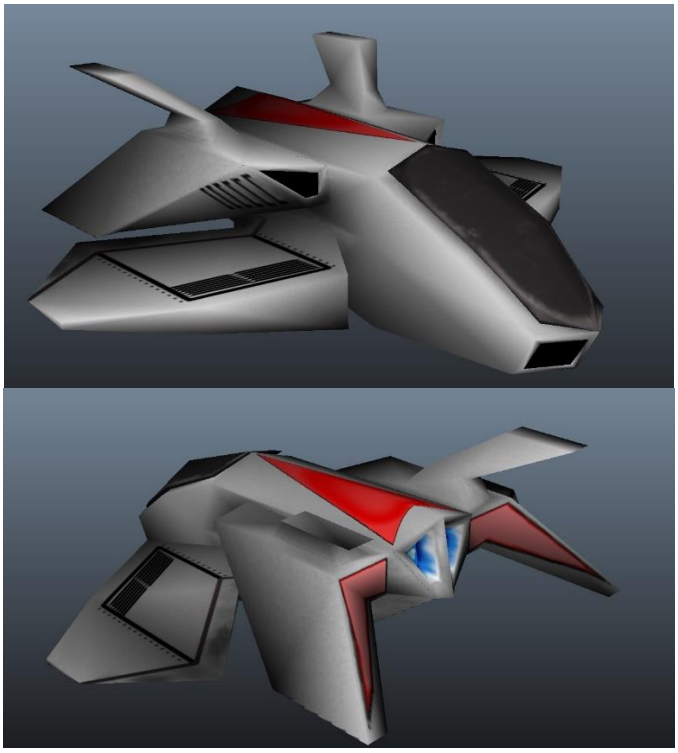
Third Person View. The player will be on the Centre of the frame and the camera is fixed. It follows the player controls. In Certain circumstances the camera angle will change like during the free fall from the water fall.

Visual Style

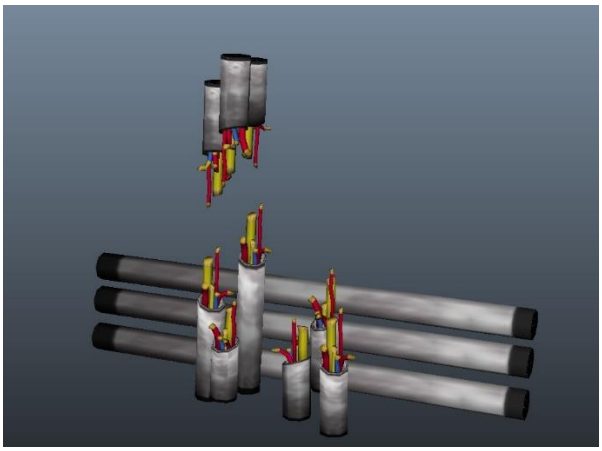
The game play is entertaining and fun to play. The colors used are all cool. Although most of the props and visuals are metal and hence having more of silver and grey gradients. Planet side has the similar graphical representation of the game mini-ninjas, sub way surfers and temple run.

Props

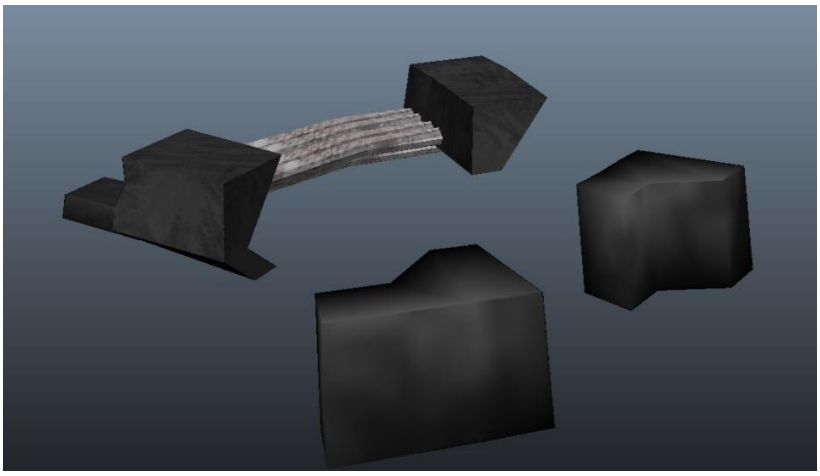
Space Craft:



Metal Pipes and Rods:



Metal Walls and Blocks:



Laser walls

{Prop yet to be build}

Steam Holes

{Prop yet to be build}

Player Character

Name: Turbug Mark II

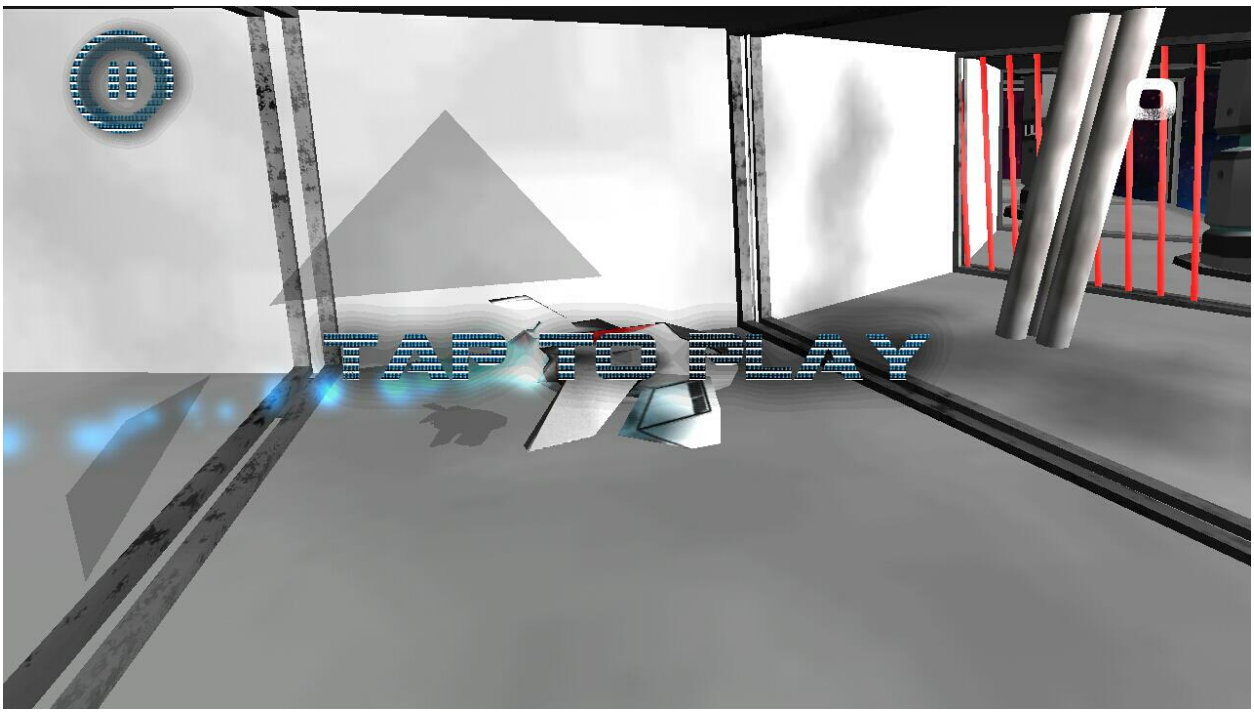
Control Mode: Single Player (SP)

Body: Aero dynamic, Turtle Shaped body.

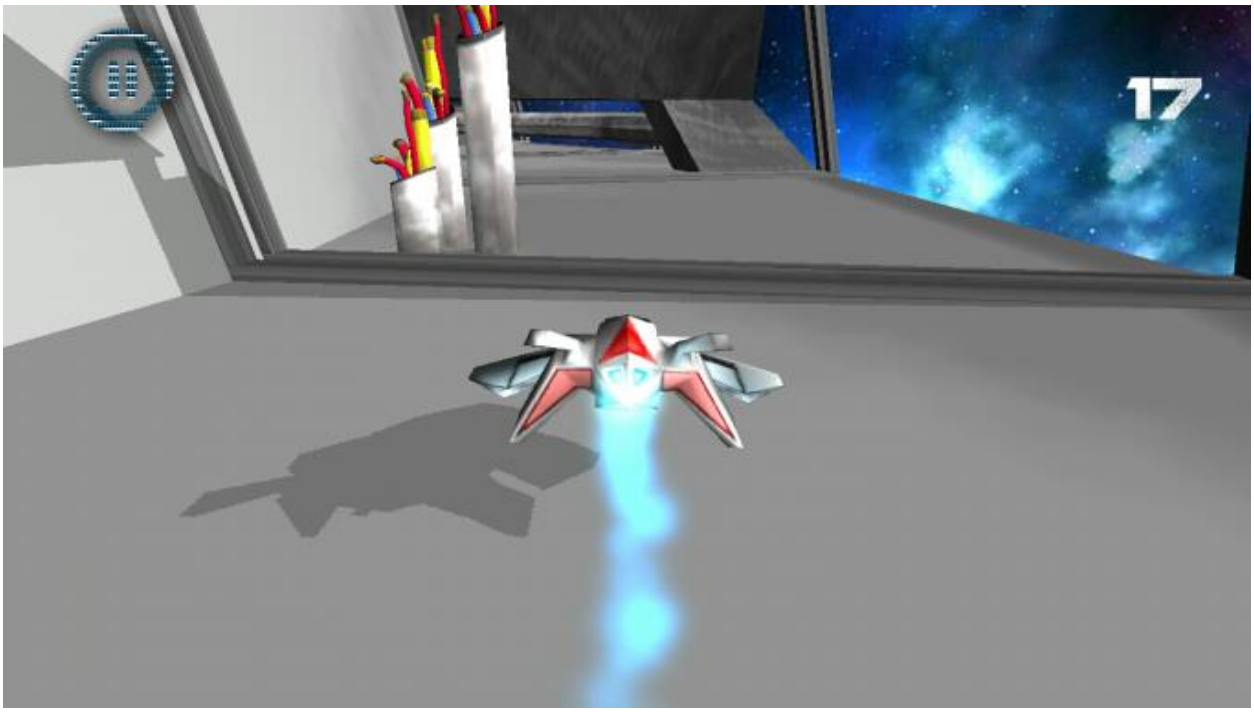
Colour: Grey, Red, blue and Black

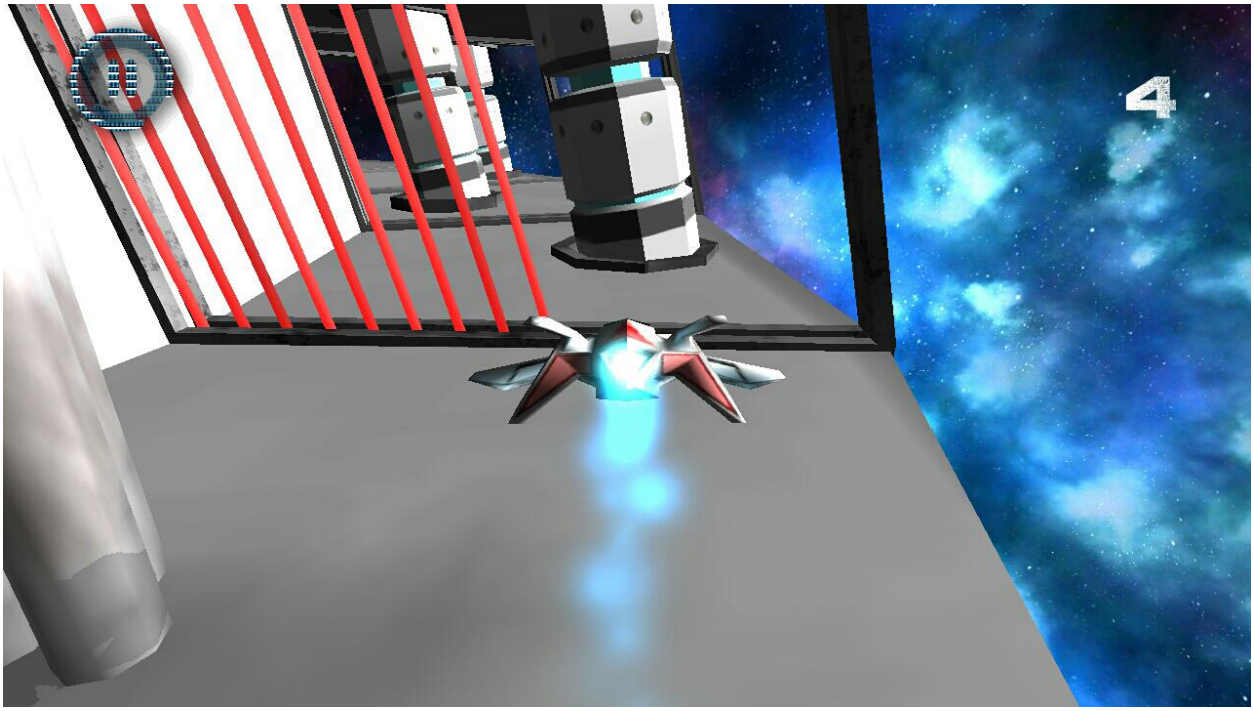
Ability: After burners (When Power ups is activated) And Normal Thruster.

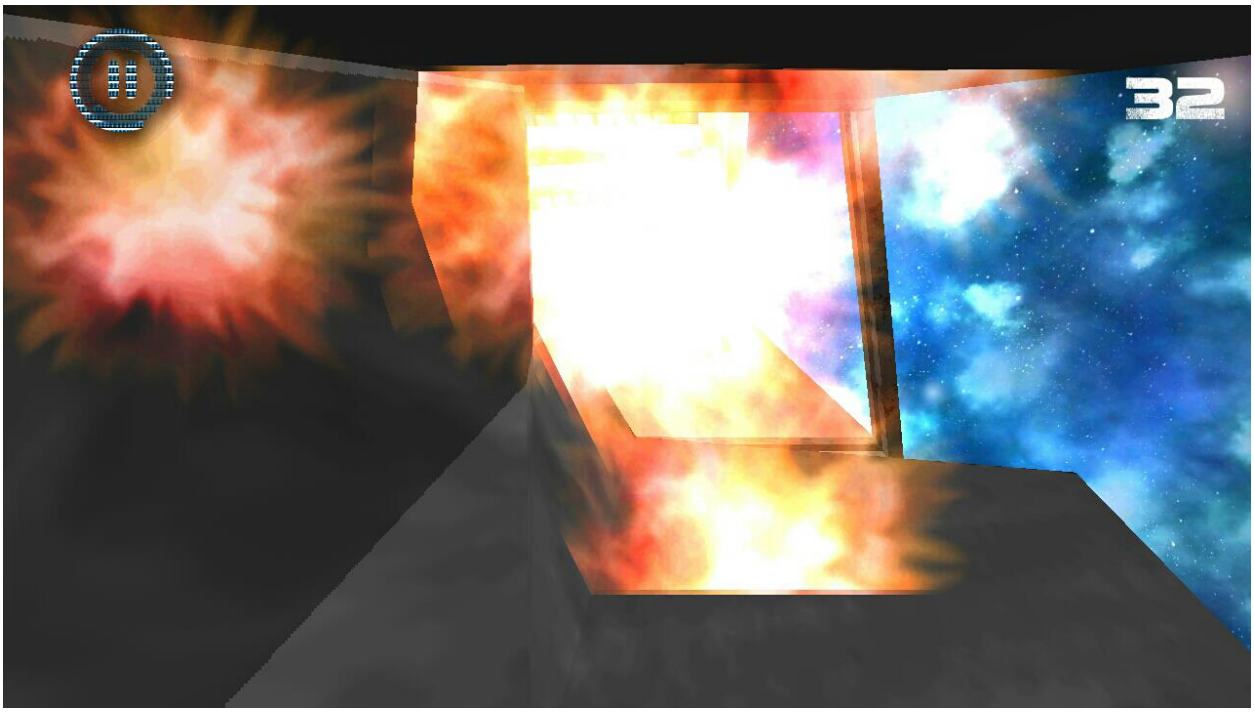
POV SCREEN



This is the Point of view that the player will see during game play. The above Image shows the location of the score and number of coins collected bar along with the power up equipped bar.

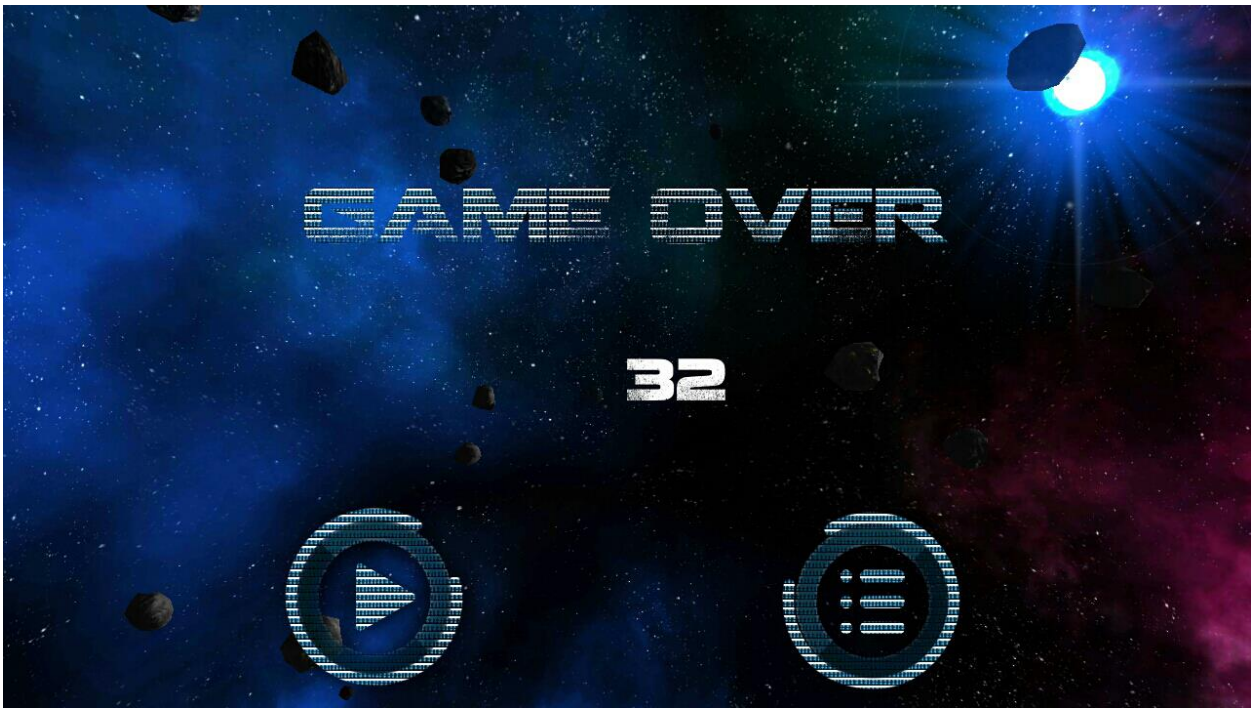

17


4

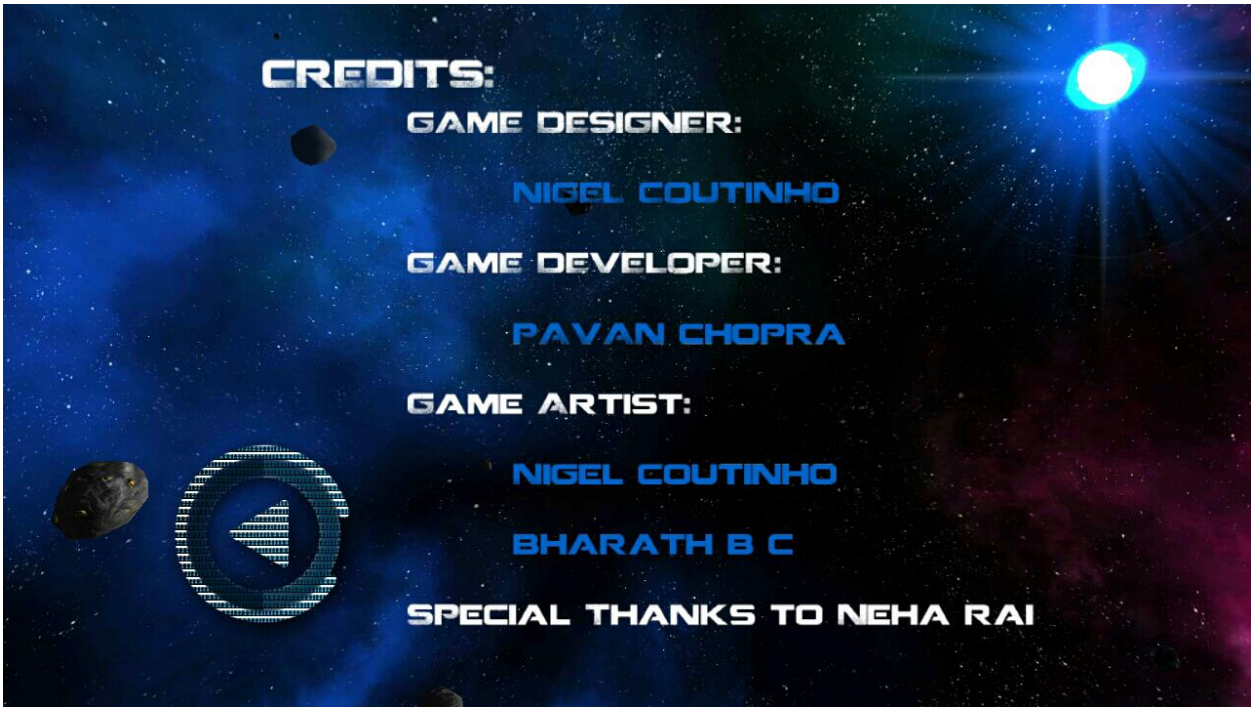


MAIN MENU AND UI SCREEN

Main Menu



Game over Screen



Credit Menu

The Above images are the Main menu and UI built for the game.

Level Design

The levels of planet shift is simple and is repetitive. The planet has 6 paths out of which each path will have 4 half's which will random as the player plays and completes one rotation. On completion of the rotation the planet will shift to the next path, gradually reaching towards the south pole of the planet which will be the 6th path of the planet. Every path has 4 defined paths having its own level of obstacles. As the player reaches the 6th path the level of obstacles will be more and so will the speed of the space jet.

Level 1 which is the 1st path which will be basic allowing the player to know and get used to the controls. And having less obstacles.

Level 2 is the 2nd path which will be having a little more obstacles and a slight speed increase.

Level 3 and 4 will be similar having the same speed increase and increased obstacles. With added moving obstacles such as lasers and hydraulics.

Level 5 will have all the obstacles from the old levels but with a slightly more speed increase.

Lastly, Level 6 will have a moderate amount of obstacles that are static and more obstacles which will be moving like lasers and steam holes. Here in this level the speed will be at its max, since the magnetic pull of the planet is the least.

Technical Design Consideration

The Game Engine that will be used to run this game will be Unity 3D and the other Soft-wares used to help create the props and other assets of the game is Auto Desk Maya 2011 and Photo Shop CS5. Most of the assets will be modeled in Maya itself and then ported to Unity. The game is designed specially to be playable on android and the Wii U.

Unique Selling Point

The Unique selling point of this game is the function and ability of the game, its back story and its game play. Although the game play may not be that unique, it still has a different feel and approach. The planets ability to shift its gravity and path is unique. Not to forget the back story of the game is also different from other stories out there in the market.

Development for Android mobile game:

The Game is currently developed for the android mobile devices. The game was designed and developed using the unity engine. And using the android support. The game support a maximum resolution of 1280x 720 which will automatically adjust the resolution on smaller android devices. Initially all models and the level was designed and modeled in Maya 2011 and was then exported to the unity engine. Java coding was used in the development of the game.

My point of view

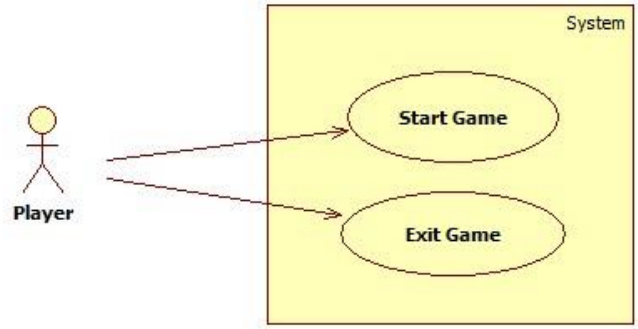
I love playing adventure game, not just adventure but a mix genre. As I just like playing games with attractive visuals, catchy tunes, interesting story lines that gets the player to be a part of the story. During the whole process to make the game I learned a lot from the modeling perspective to the engines and coding perspective. There were many more features I wanted to implement but due to the increase of draw calls that was forming I was restricted to add much more. The one thing I didn't like was unity's lighting conditions. It didn't bring the real space feel into the game but it managed to bring my idea out and let people who love simulation games to play it.

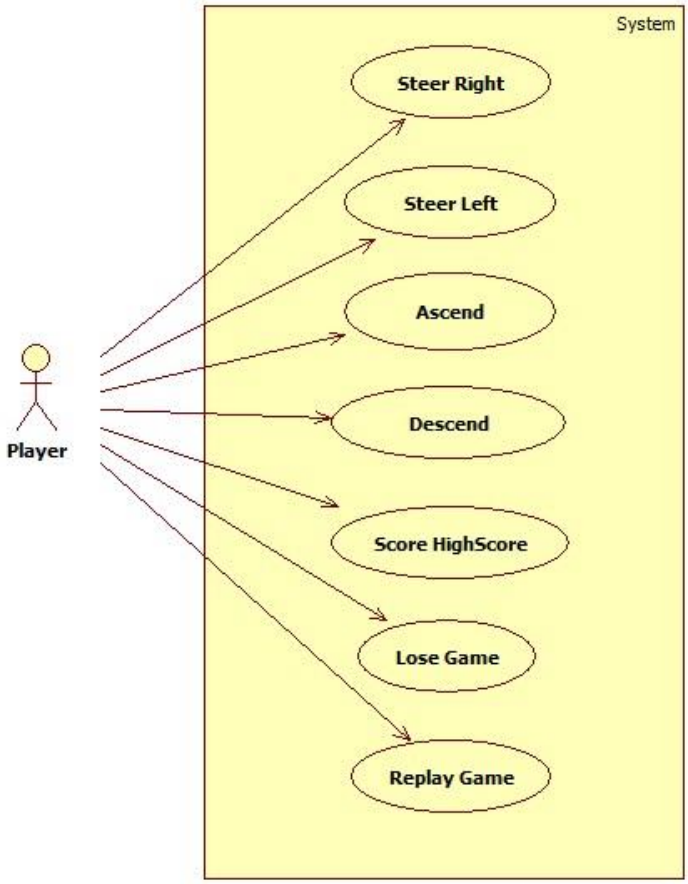
UML DIAGRAMS

USE CASE DIAGRAM:

Description: The below diagram represents the actions, user can performed in the game. This diagram helps us to know all the actions user can perform in the game.

Diagram:

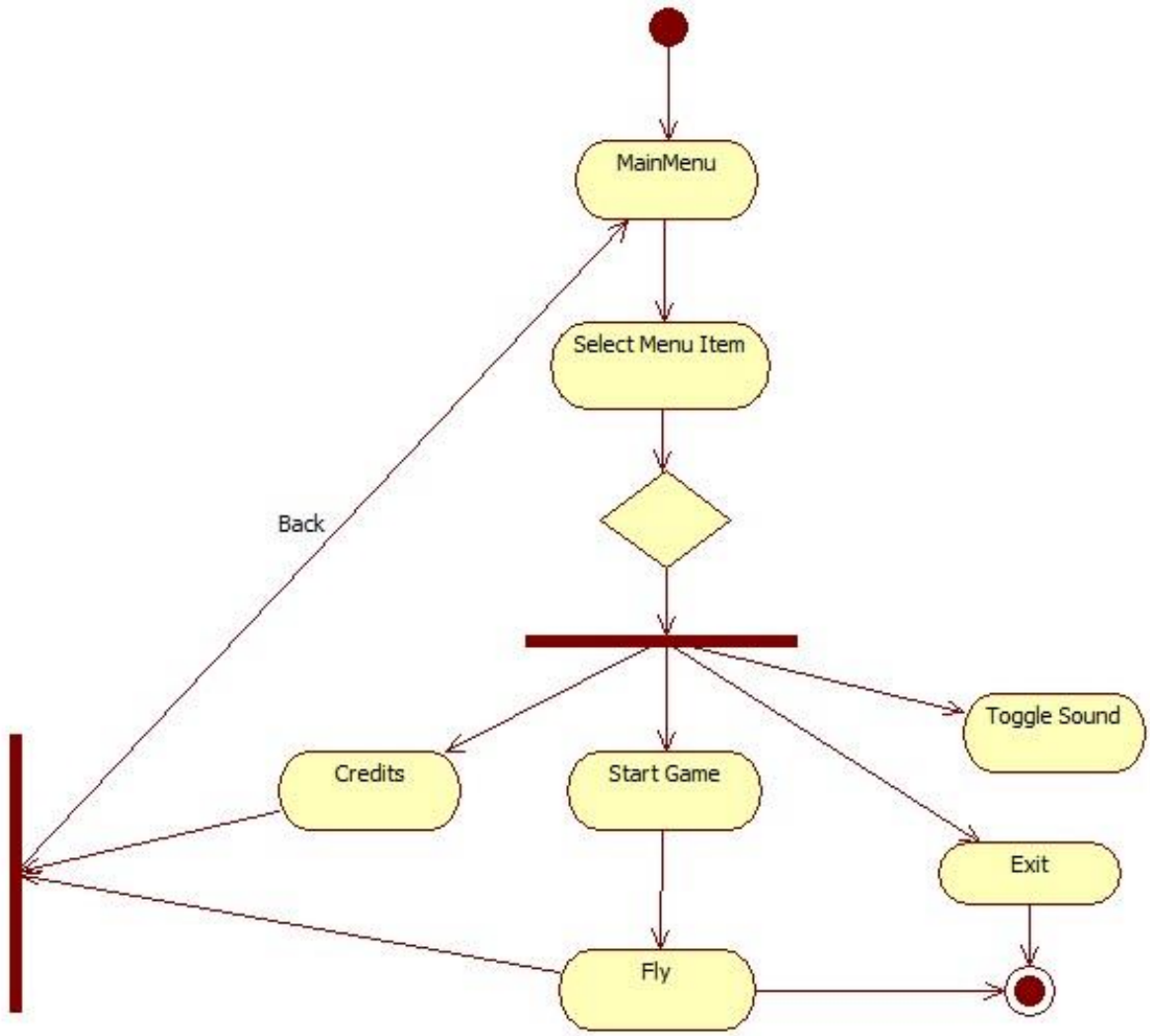


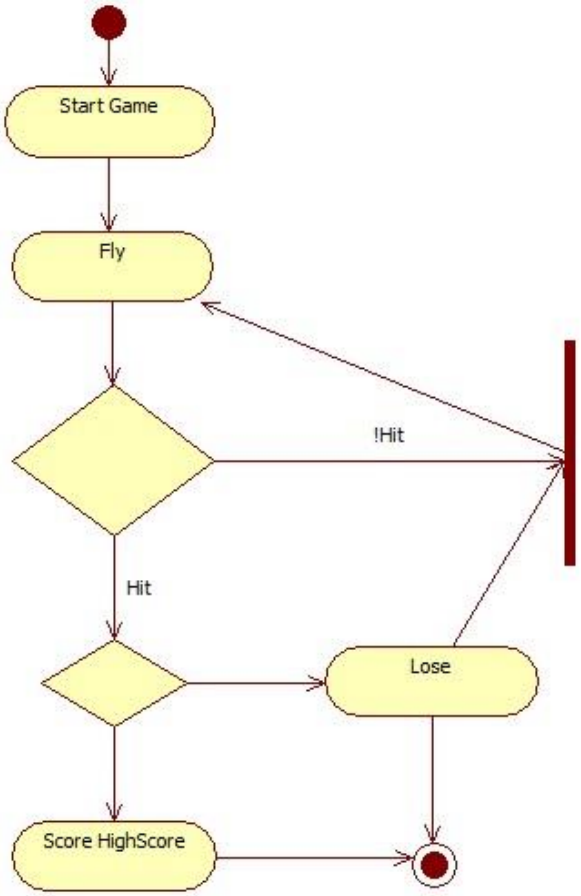


ACTIVITY DIAGRAM

Description: Activity Diagram is used to show the game-play mechanics working parallel.

Diagram:

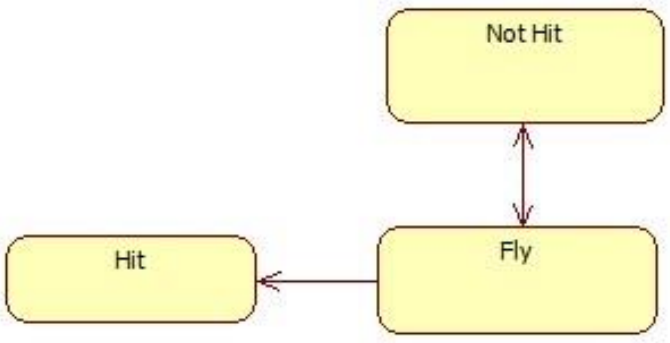




STATE DIAGRAM

Description: The diagram represents States of the Player. Activity Diagram helps us to know what activities can be performed by a Player or an AI.

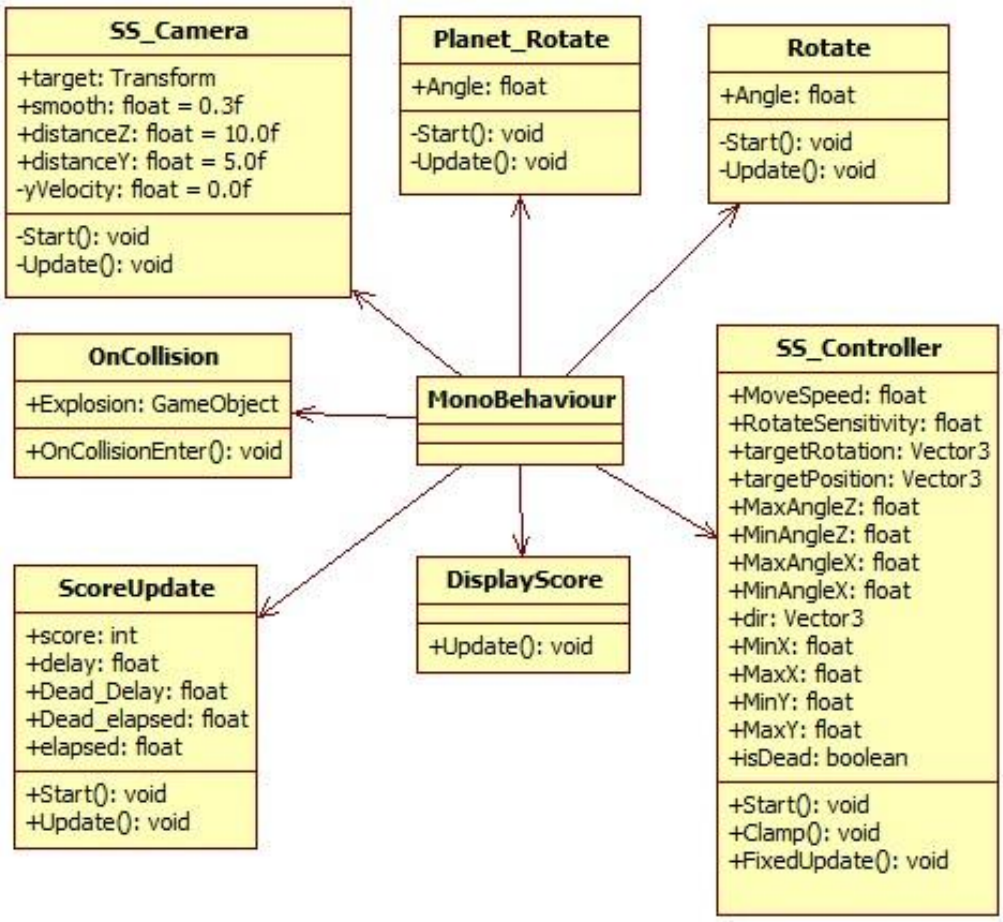
Diagram:



CLASS DIAGRAM

Description: The below diagram represents the class diagram. Then some important variables in the class and then functions declared in the class.

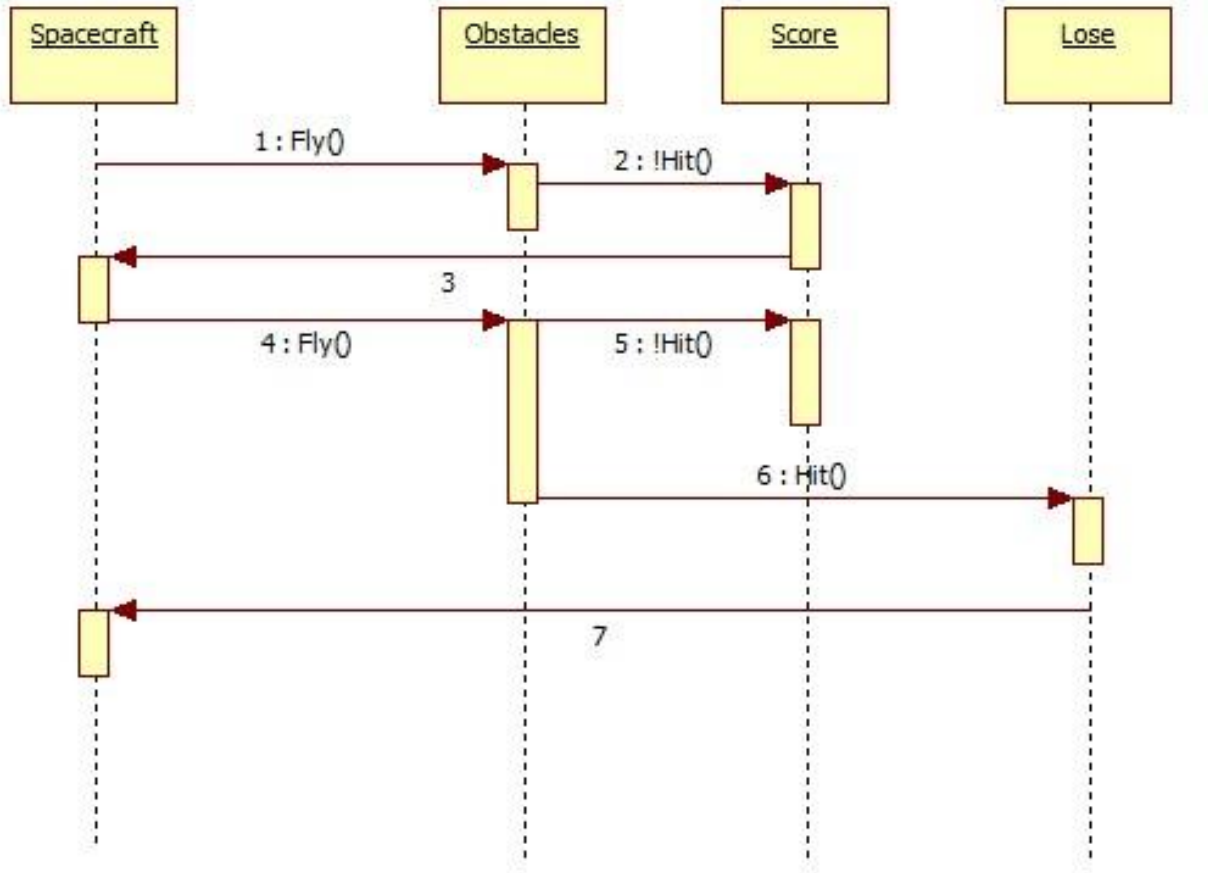
Diagram:



SEQUENCE DIAGRAM

Description: Sequence Diagram helps user to know the interaction between Player and the varies Puzzles in the game.

Diagram:



SOURCE CODE

SS_Camera.cs

```
public class SS_Camera : MonoBehaviour
{
    public Transform target;
    public float smooth = 0.3f;
    public float distanceZ = 10.0f;
    public float distanceY = 5.0f;
    private float yVelocity = 0.0f;

       // Use this for initialization
       void Start () {

       }

       // Update is called once per frame

    void Update()
       {
              if(target==null)
                     return;

        float yAngle = Mathf.SmoothDampAngle(transform.eulerAngles.y,
target.eulerAngles.y, ref yVelocity, smooth);
        Vector3 position = target.position;

        // Offset the Camera
        position += Quaternion.Euler(0, yAngle, 0) * new Vector3(0, distanceY - 1, -
distanceZ);
        transform.position = position;
        transform.LookAt(target);
    }
}
```

SS_Controller.js

```
       var MoveSpeed : float = 50f;
       var RotateSensitivity: float = 100f;
       var targetRotation : Vector3 = Vector3.zero;
       var targetPosition : Vector3 = Vector3.zero;
       var MaxAngleZ : float = 45;
       var MinAngleZ : float = -45;
       var MaxAngleX : float = 30;
       var MinAngleX : float = -30;
       var dir:Vector3;
       var MinX : float;
       var MaxX : float;
       var MinY : float;
       var MaxY : float;
       static var isDead : boolean;
       // Use this for initialization
```

```
	function Start () {
	isDead = false;
	Screen.sleepTimeout = SleepTimeout.NeverSleep;

	MinX = 62;
	MaxX = 86;
	MinY = -10;
	MaxY = 5;
	}


	// Clamp the Angle
	function Clamp()
	{
		this.transform.position.x = Mathf.Clamp(this.transform.position.x, MinX,
MaxX);
		this.transform.position.y = Mathf.Clamp(this.transform.position.y, MinY,
MaxY);
	}


	// Update is called once per frame
	function FixedUpdate ()
	{
	var deadZone : float = 0.0f;

		targetPosition = Vector3.zero;

		// Get Android Device's Accelerometer Values
		dir.x = Input.acceleration.x + Input.GetAxis("Horizontal");
		dir.y = -Input.acceleration.y + Input.GetAxis("Vertical");


		if(dir.x >= deadZone || dir.x <= -deadZone)
		{
			targetPosition.x = dir.x;
			targetRotation.z = Mathf.Clamp(-(dir.x * RotateSensitivity),
this.MinAngleZ, this.MaxAngleZ);
		}
		if(dir.y >= deadZone || dir.y <= -deadZone)
		{
			targetPosition.y = dir.y;
			targetRotation.x = Mathf.Clamp(-(dir.y * RotateSensitivity),
this.MinAngleX, this.MaxAngleX);
		}

		// Rotate and Translate the SpaceCraft
		this.transform.rotation = Quaternion.Slerp(this.transform.rotation,
Quaternion.Euler(this.targetRotation), 10f*Time.deltaTime);
		this.transform.Translate(targetPosition * MoveSpeed * Time.deltaTime,
Space.World);
		Clamp();
	}
```

ScoreUpdate.js

```
static var score : int;
var delay : float;
var Dead_Delay : float;
private var Dead_elapsed : float;
private var elapsed : float;

function Start () {
score = 0;
}

function Update () {
// Increment the Score
elapsed += Time.deltaTime;
this.guiText.text = ""+score;
if(this.elapsed > delay && !SS_Controller.isDead)
{
	elapsed = 0;
	score++;
}

if(SS_Controller.isDead)
{
	Dead_elapsed += Time.deltaTime;
	if(this.Dead_elapsed > this.Dead_Delay)
	{
		Dead_elapsed = 0;
		// Switch Scene if isDead Boolean is true
		Application.LoadLevel("Game_Over");
	}
}
}
```

RotateAround.js

```
function Start () {
}

function Update () {
// Rotate the GameObject along Y-axis
this.transform.RotateAround(Vector3.up, 0.001f);
}
```

OnCollision.js

```
var Explosion : GameObject;

// Use this for initialization

function OnCollisionEnter( collision : Collision )
{
	// Instantiate Explosion
	Instantiate(Explosion, transform.position, Quaternion.identity);
	SS_Controller.isDead = true;
	Destroy(this.transform.parent.gameObject);
}
```