

SCIENCE FOR GAMING DOCUMENT

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

July 2013

Anngad Shyam Prasad

2nd Year UG – Game Development

2011UG03028



ICAT, Design and Media College,

Bangalore

Contents

Overview:	3
Marketing:.....	3
Gameplay:	3
Game Controls:	4
Game Story:	4
Props with Images:.....	4
Level Description:.....	5
Technical Specification:.....	5
Use Case:.....	7
Class Diagram:.....	8
Activity Diagram:.....	9
State Diagram:	10
Sequence Diagram:	11
Source Code:	Error! Bookmark not defined.
Screenshots:.....	12

Overview:

The main objective of the game is to collect as much data as possible on the moon. Data is collected by visiting the data check posts and collecting their information. To collect data from the check post, all the player has to do is navigate the moon buggy and make sure it passes through the various check posts found on the level. Once the player has collected as much data as possible, the player can then move towards the end of the level. The player's score will be calculated depending on the amount of check posts visited. Compete with your friends and see who can collect the highest score by the end of the game.

Marketing:

Target Audience:

Age: This game aims at entertaining children of ages 5 and above.

Gender: This game can be played by both genders.

Geographic Target: This game is aimed at audiences world-wide.

Platform:

Windows OS with XNA and C# runtime installed.

Gameplay:

The player has to navigate the moon buggy around the level and collect data from check posts; the player's score is determined by the number of check posts visited by him. The player is given three lives or chances to complete the level and has to do it within a given time.

Game Levels (no. of levels): 1

Game Rules:

1. The main objective of the game is to collect as much of data from the check posts as possible.
2. There are many check posts scattered throughout the level, the player has to explore the level to find them.
3. If the timer reaches zero, the player loses the game and has to retry the level.
4. The gravity is similar to that of the moon, the player has to keep this in mind not to overshoot the jumps.
5. If the player's lives reaches zero, the player has to restart the level from the beginning.

Game Controls:



: Decelerate the moon buggy.

: Accelerate the moon buggy.

Releasing the keys causes the moon buggy to brake.

Left ctrl can be pressed to give the buggy a small upward boost.

Game Story:

The plot of the game revolves around a moon buggy being sent to the moon. The player has to navigate the moon buggy throughout the moon and collect the data retrieved by the check posts on the moon.

Props with Images:

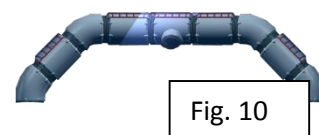
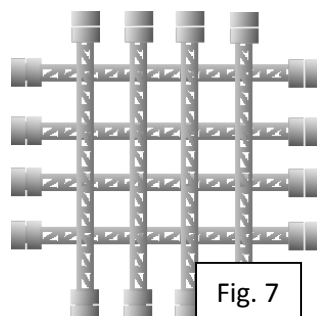
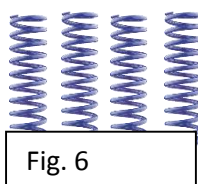
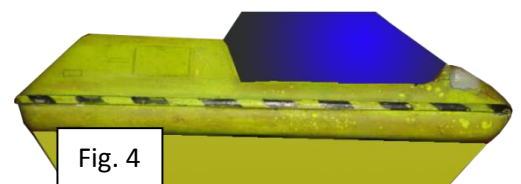
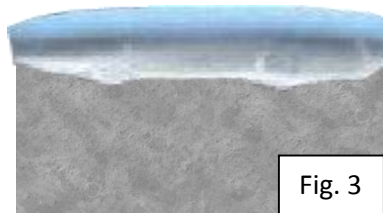
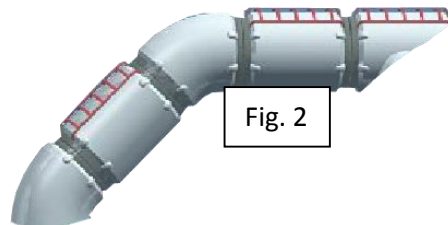


Fig. 1 : This is the prop used for the check posts throughout the game.

Fig. 2, Fig. 7, Fig. 8. Fig. 9 and Fig. 10 are used as background props throughout the game and have no effect on the gameplay. These are used purely for visual purposes.

Fig. 3: This prop is used as the base ground/platforms of the levels.

Fig. 4: This is the prop used for the body of the buggy.

Fig. 5: These are the wheels used to control the buggy.

Fig. 6: These are the springs/shock absorbers which connect the wheels to the buggy.

Level Description:

Level 1:

In the first level, there are five checkpoints found in five different locations, the player is given a three lives and a time limit to find as many as possible and complete the level. This level is very easy and is more like a tutorial; the player is shown the controls and is given a chance to get used to the game play mechanics and the flow of the game.

Technical Specification:

XNA Game Studio

Visual Studio 2010 C#

Photoshop

Illustrator

Farseer Physics Engine

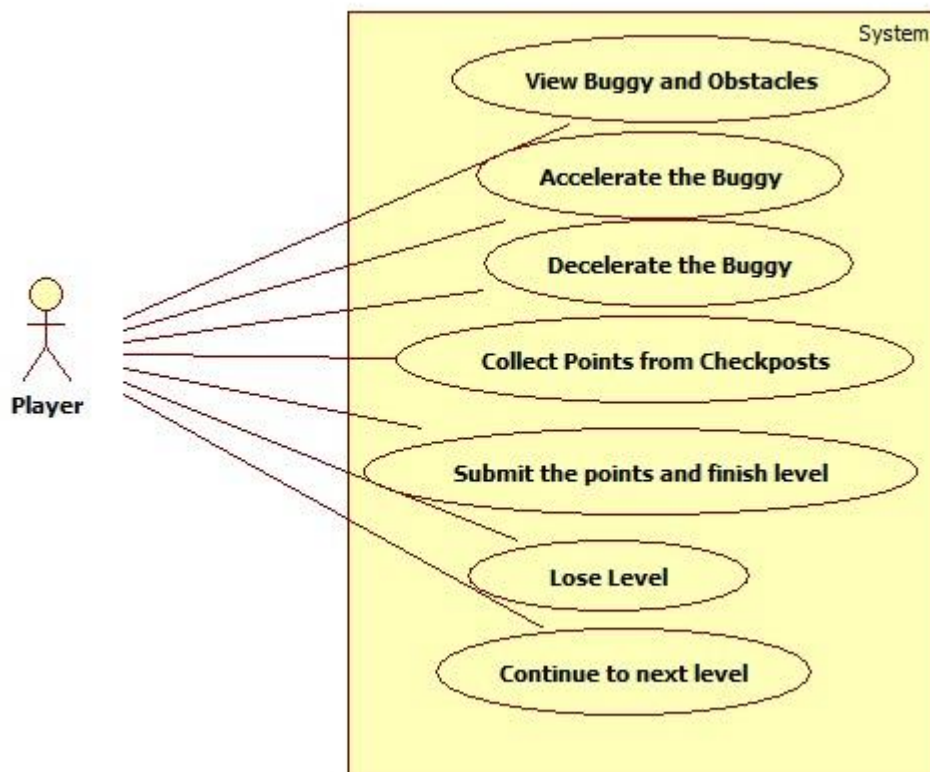
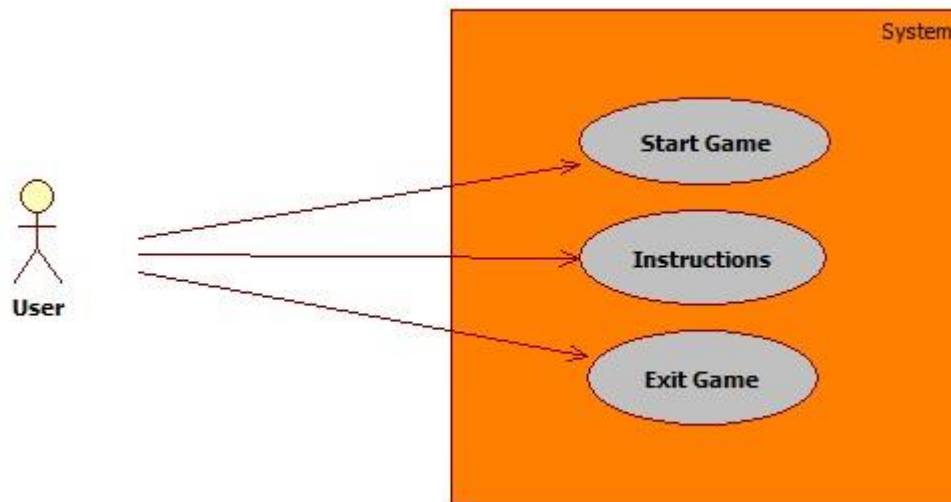
Mercury Particle Engine

UML DIAGRAMS

Use Case:

Description: The below Diagram represents the actions of the User performed in the game. This diagram helps us to know which all are the actions will be there in the game.

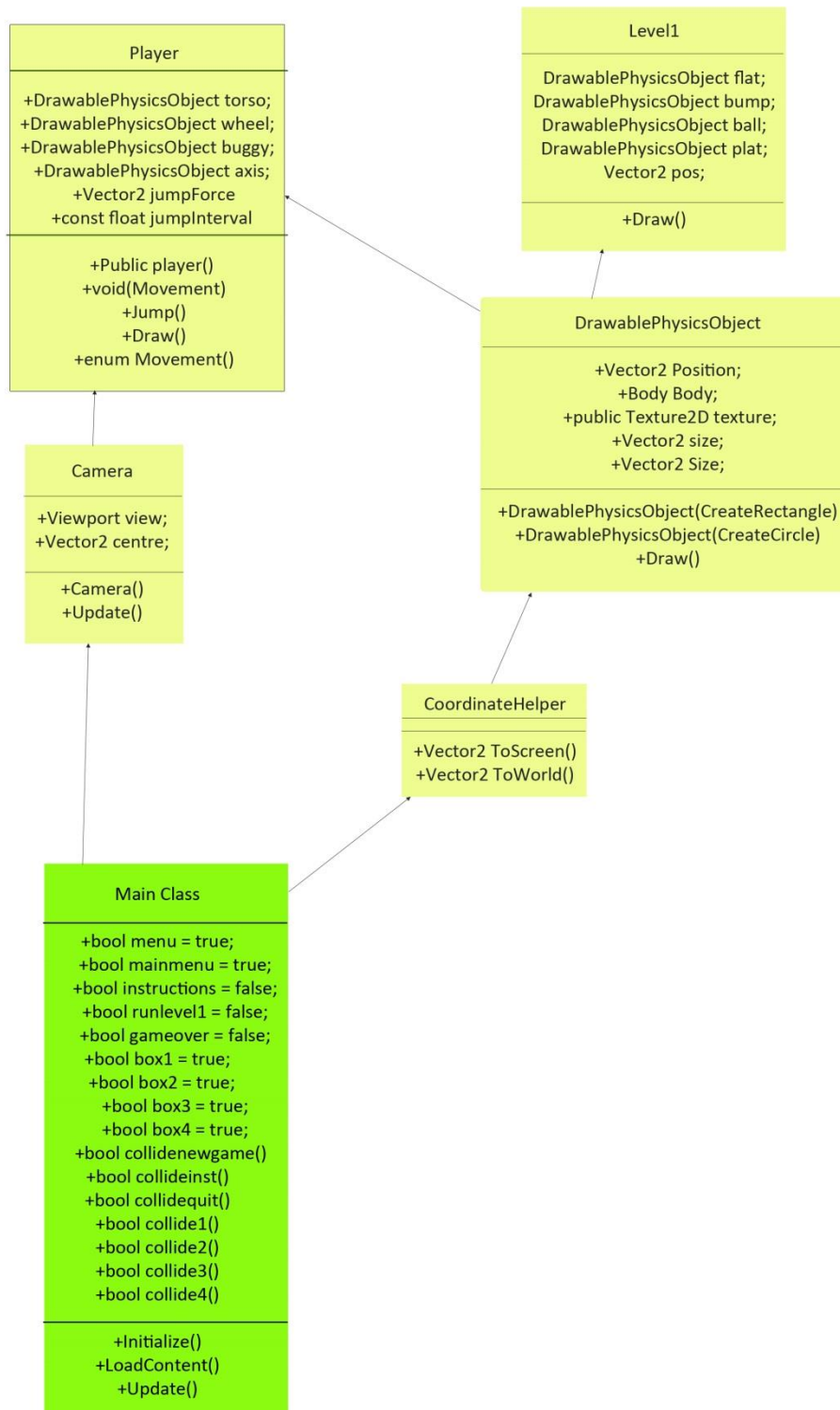
Diagrams:



Class Diagram:

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

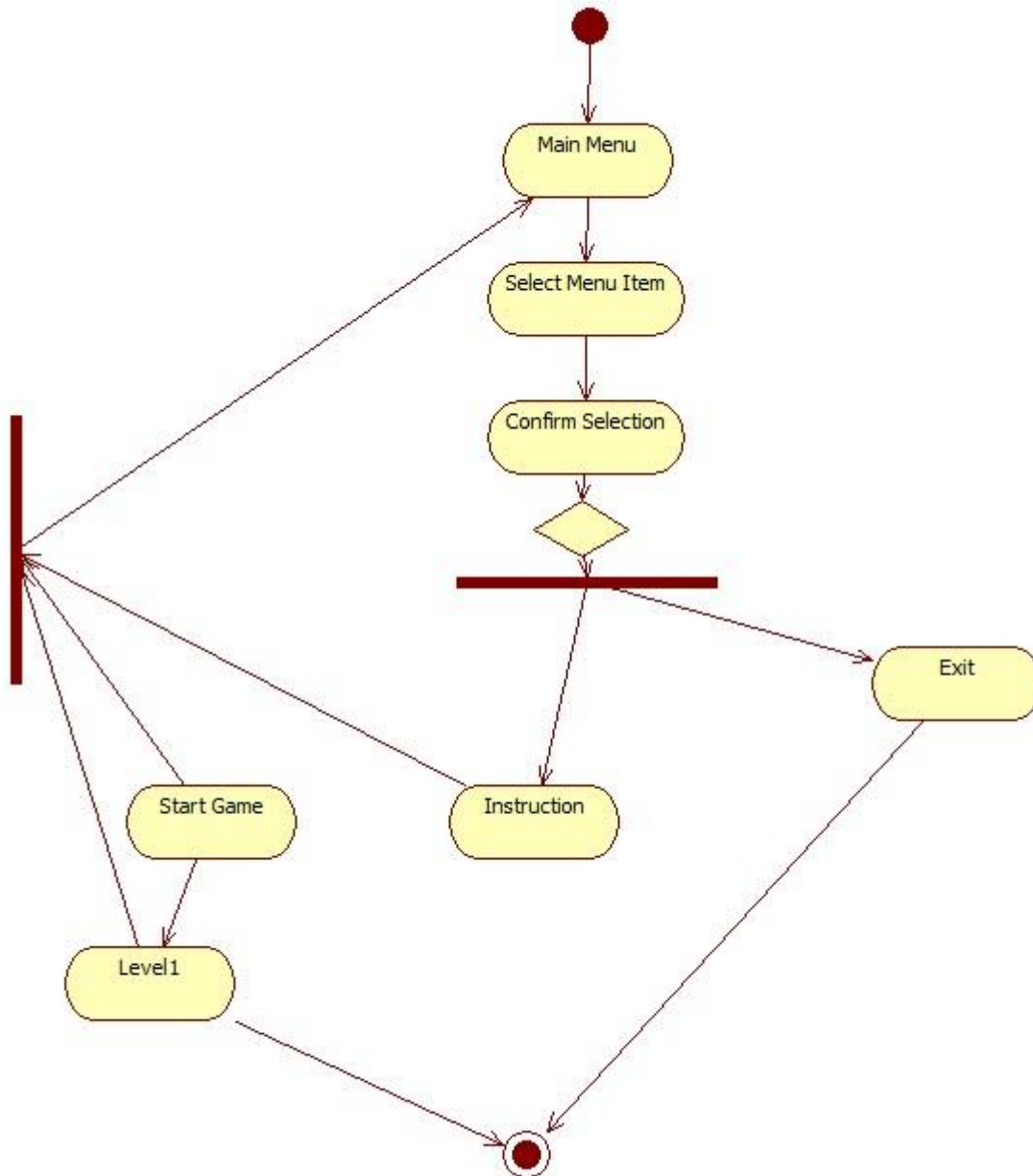
Diagram:



Activity Diagram:

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

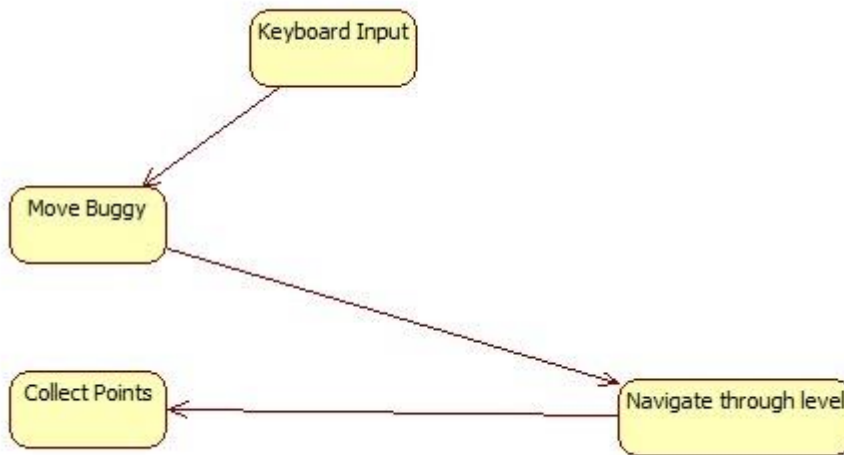
Diagram:



State Diagram:

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

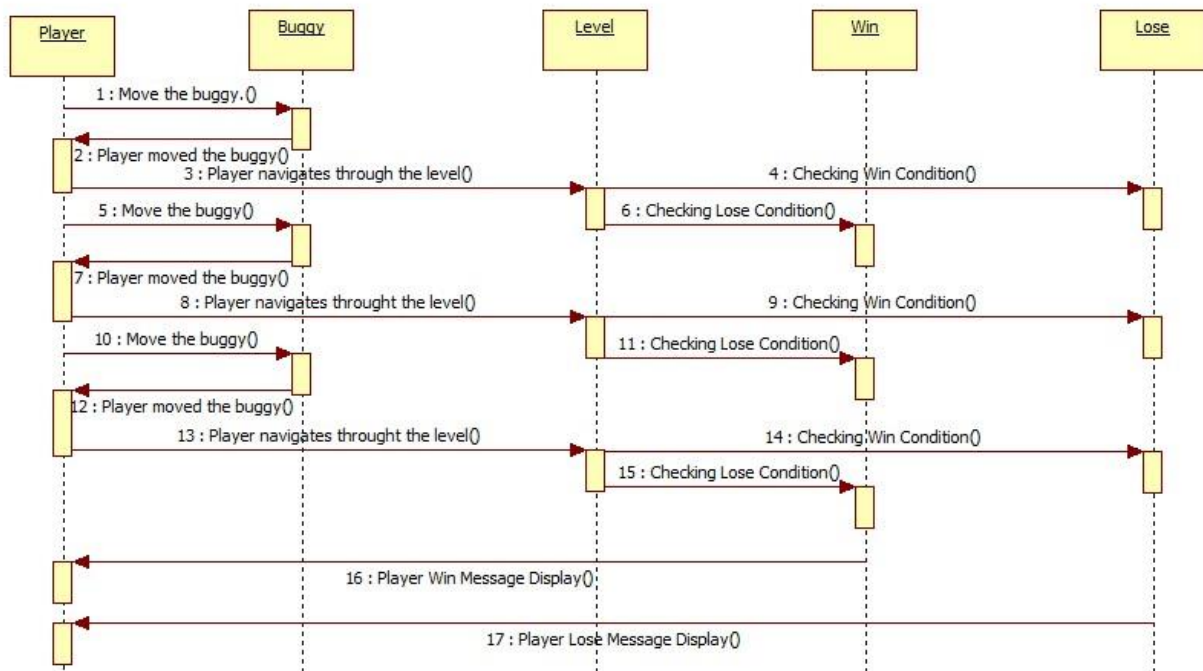
Diagram:



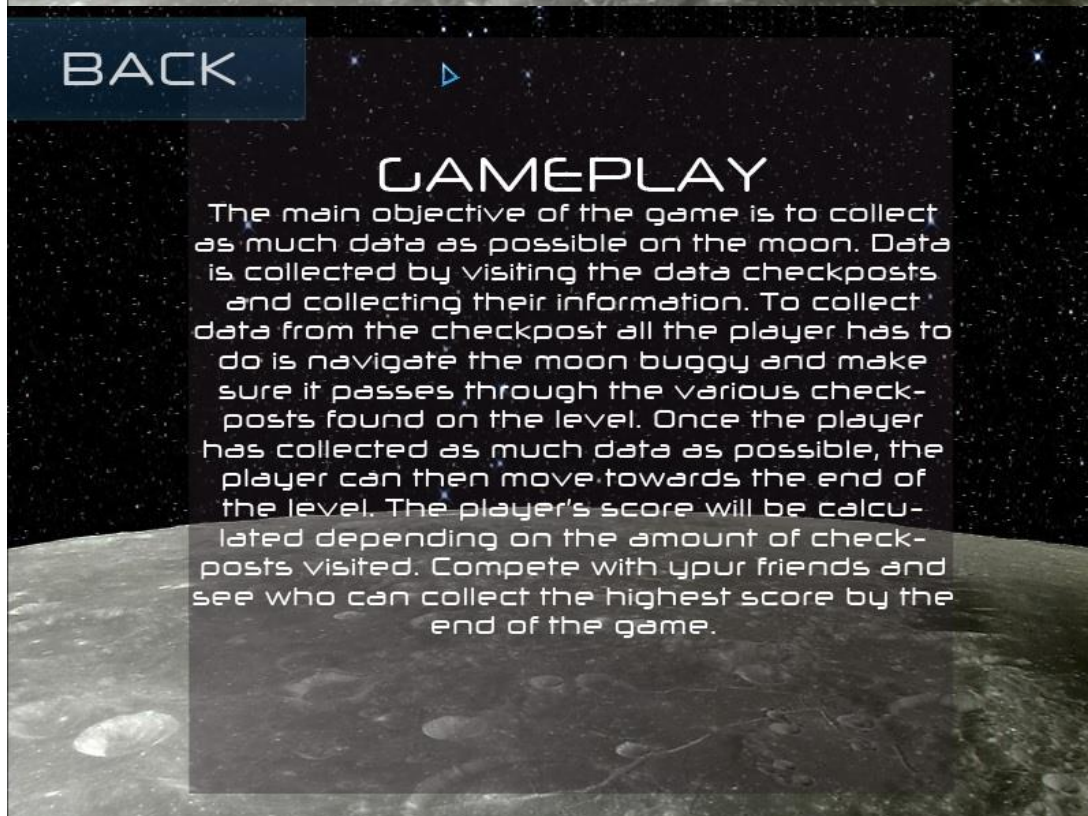
Sequence Diagram:

Description: Sequence Diagram helps user to know the interaction between Player, Place and Props.

Diagram:



Screenshots:



Time Remaining :170.5023

Checkpoints : 0/4

CONTROLS

 - This is used to accelerate

 - This is used to reverse

 Braking is done by releasing the keys.

 Lctrl - Activates the springs giving the moon buggy a small upward boost

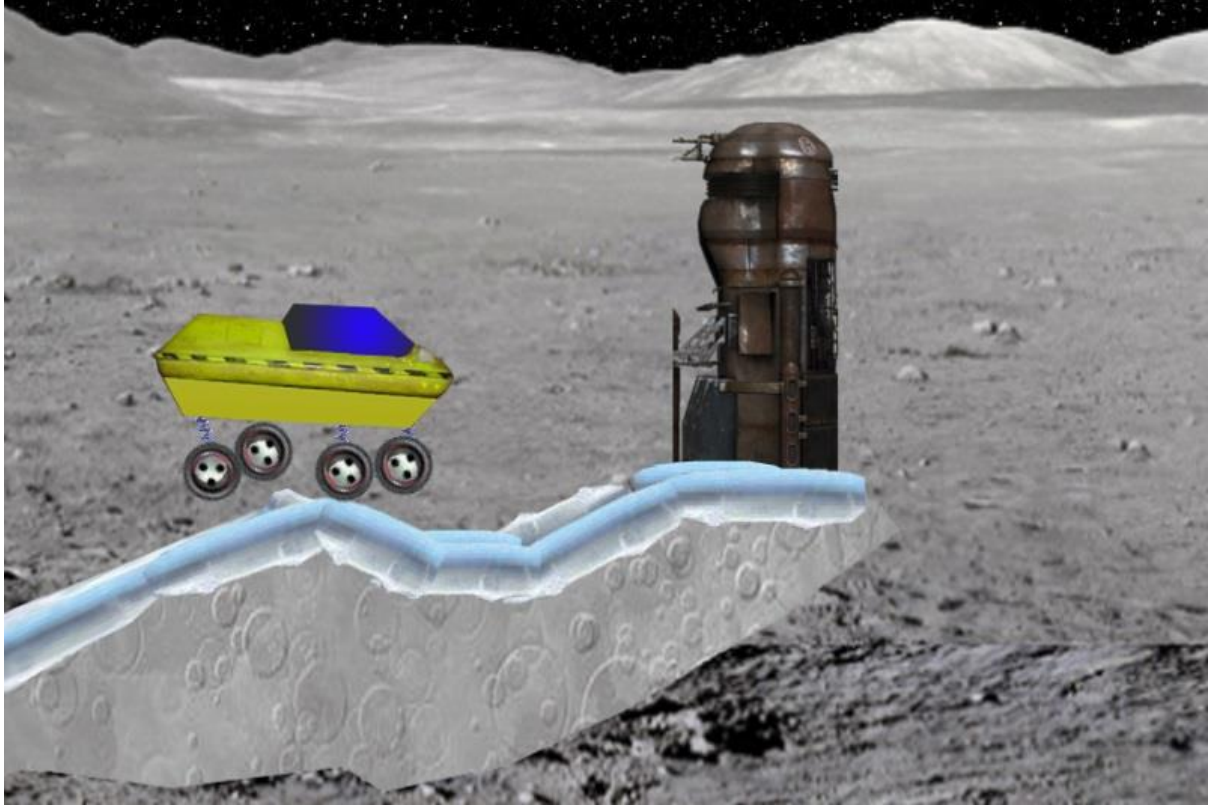
Time Remaining :165.9534

Checkpoints : 0/4



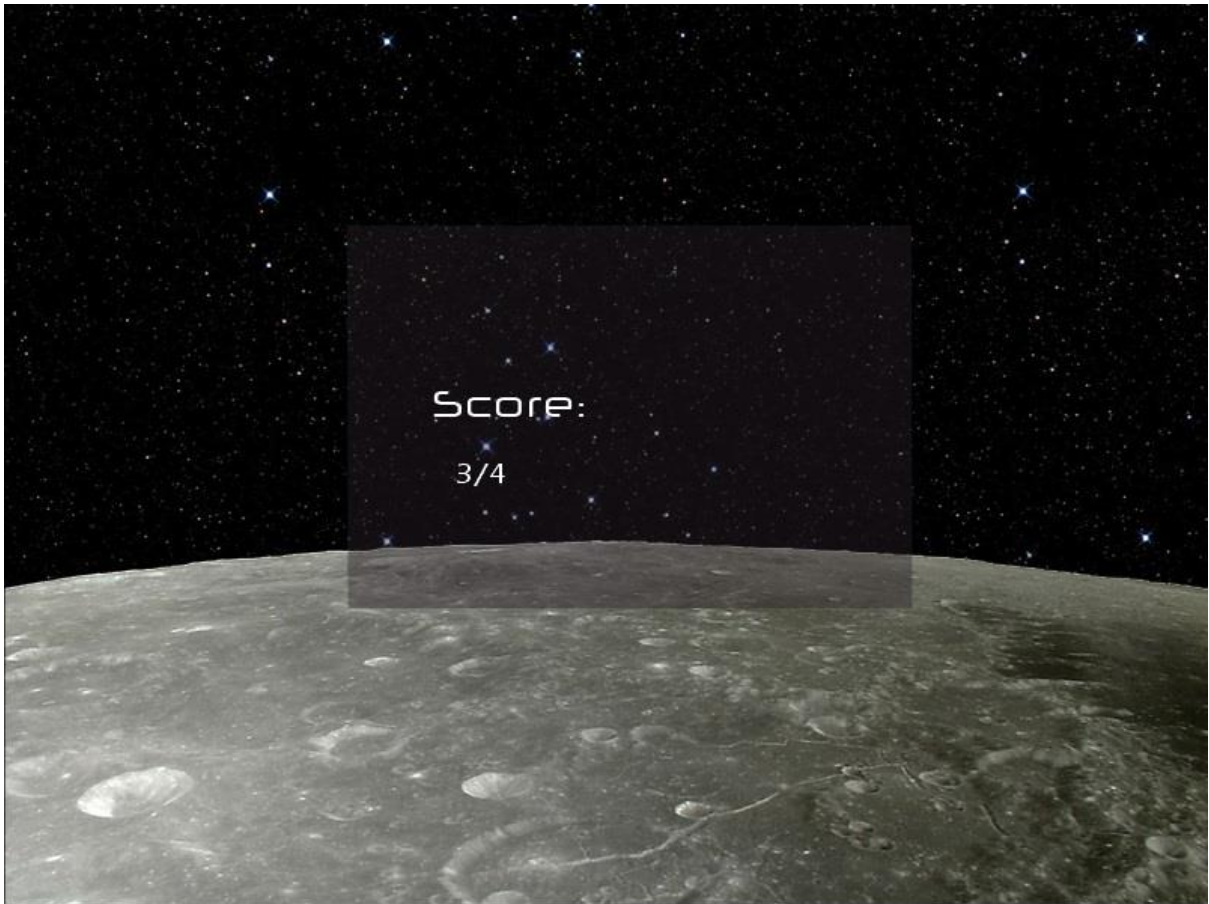
Time Remaining : 151.8569

Checkpoints : 1/4



Score:

3/4



Time Remaining :126.6624

Checkpoints : 2/4

