**SOFTWARE REQUIREMENT SPECIFICATION**

Software Requirement Specification document describes the intended purpose and that encompasses almost any graphical aspect and this document specifies generation of images of 3-dimensional scenes. The THIRD EYE application which would be developed will focus to bring Augmented reality(AR) using computer generated objects.

## Functional Requirements

This module includes the following stages of development:

* **Camera**
* **Marker**
* **Gesture Technology**
* **3D-Modeling**

**Camera:** Camera identifies and interprets a marker, often a black and white barcode image. The software analyses the marker and creates a virtual image overlay on the screen, tied to the position of the camera. This means the app works with the camera to interpret the angles and distance is away from the marker.

**Gesture Technology:** Gesture Technology consists of interpreting human gestures via mathematical algorithms. Gesture can originate from any bodily motion or state but commonly originate from the face or hand, here we make use of hand gesture.

**Marker:** Gesture Technology consists of interpreting human gestures via mathematical algorithms. Gesture can originate from any bodily motion or state but commonly originate from the face or hand, here we make use of hand gesture.

**3D-Modeling:**

The 3D modeling consists of the models of the solar system, all the planets models are developed in 3 dimensional views and project when marker is mapped by the camera.

**Hardware specifications**

* The system which is used to develop the application is compelled to have the following hardware requirements

**System**: Laptop or Personal Computer

**Processor**: Dual core i3 with 1.70 Gigahertz speed

**RAM** : 4 Gigabytes

**Memory** : 50 Gigabytes

**Software specifications**

* The developer will work on the following set of software tools listed below:

**Operating** **System** : window 98/higher, window xp with environment

**Compiler** : GNU GCC complier/C++ compiler

**Libraries** : Supporting glut32.h, opengl32.h, glu32.h, libARvideo.dll,

**Languages** : Open GL and open Cv.

## 1.2 Non Functional Requirements

 The Non Functional requirements for the application that will be developed will have the following factors:

**Performance**: To work properly, i.e. providing a “believable” augmented reality view to the user, the application should respond rapidly to the user’s movements. Moving the mobile device around changes the view of the user and thus, the displayed information should update accordingly.

**Maintainability**: Changing, adding and removing information and information sources should be facilitated. While users run the application base and make it available to the user’s afterwards.

**Usability:**It should be easy for the users to install and use the application. Functions should be clear, easy to find either self-explanatory or well-documented.

# Purpose

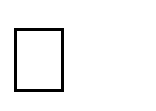
 The main purpose of this project is to demonstrate the use of augmented reality (AR) using computer graphics. The project helps to create AR object using computer vision algorithms by the help of some video tracking libraries calculate the real camera position and orientation relative to physical markers in real time and use the gathered date inside OpenGL to display and manipulate object.

# Scope

 This is basically an educational where we demonstrate use of virtual imagery on the real world and manipulating the computer generated object using OPENGL and other library.

# Technical Feasibility

 Technical feasibility is the study of the resource availability that may affect the ability to achieve an acceptable system. It is essential that the process of the analysis and definition be conducted in parallel with an assessment with technical feasibility. This project does not have technical difficulty in installing, does not have complex errors in installing. The errors are is easy to debug.

 **User interface**: It allows us to interact with the objects through mouse, keyboard, voice and gesture to manipulate the objects.

Mouse interactions

Keyboard interactions

# . IMPLEMENTATION

 . Graphical functions.

* AR tool function
* User defined function
* Video.h

# . Testing

 . The objective of this chapter is to introduce technique that may be used to test programs to discover program faults. In other words we can say that this chapter is mainly based on:

* Pose & Position Estimation
* Camera calibration
* Image Processing
* Pose Estimation
* Background Video Stream Display
* Rectangle Extraction

# . Testing process

* Unit testing
* Module testing
* System testing

# . Snapshots

# . List of figures

# . Conclusion

# Bibilography