# VISVESVARAYA TECHNOLOGICAL UNIVERSITY

**"JnanaSangama", Belagavi-590018, Karnataka**


***A Mini Project Report on***

## "RAIN WATER HARVESTING"

***Submitted in partial fulfilment of the requirement for the award of degree of***

**Bachelor of Engineering**

In

**Computer Science and Engineering**

***Submitted by***

**P SAI SHREYA (4NN20CS034 )**

**BHAVYA T (4NN20CS015)**

Under the Guidance of

**Ms. SHEEBAN E TAMANNA**

**Assistant Professor,**

**Dept. of CSE**


ESTD-2008

**Department of Computer Science and Engineering**

**NIE Institute of Technology**

Mysore -570018

## DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

## NIE Institute of Technology, Mysore

ESTD-2008

# *CERTIFICATE*

This is to certify that the mini project work entitled ***"RAIN WATER HARVESTING"*** is carried out by ***P SAI SHREYA*** bearing **4NN20CS034** and ***BHAVYA T*** bearing ***4NN20CS015*** in the partial fulfilment for the SIXTH semester of **Bachelor of Engineering degree** in **Computer Science and Engineering** of the **Visvesvaraya Technological University, Belagavi** during the academic year **2022-23**. The project report has been approved as it satisfies the academic requirements with respect to project work prescribed for the Bachelor of Engineering.

**Signature of the guide**
**Ms. SHEEBAN E TAMANNA**
Asst. Professor,
Dept. of CSE
NIEIT, Mysuru

**Signature of the HOD**
**Dr. Usha M.S**
Associate Professor and Head
Dept. of CSE
NIEIT, Mysuru

## External Viva

| Name of the examiners | Signature with Date |
|---|---|
| **1**…….... | **1**……. |
| **2**……. | **2**……. |

# ACKNOWLEDGEMENT

We sincerely owe our gratitude to all the persons who helped and guided us in completing this mini project work.

We are thankful to **Dr. ROHINI NAGAPADMA , Principal, NIEIT Mysuru**, for having supported us in our academic endeavors.

We are thankful to **Dr. Usha M S**, Associate Professor & Head, Department of Computer Science and Engineering NIEIT for providing us timely suggestions, encouragement and support to complete this mini project.

We would like to sincerely thank our guide **Ms. Sheeban E Tammana,** Assistant Professor, Department of Computer Science and Engineering for providing relevant information, valuable guidance and encouragement to complete this mini project.

We are grateful to all teaching and non-teaching staff members of Department of Computer Science and Engineering, for the valuable information provided by them in their respective fields. We are thankful for their co-operation during the period of our mini project.

Lastly, we thank almighty, our parents and friends for their constant encouragement without which this mini project would not be possible.

Yours sincerely,

**BHAVYA T (4NN20CS015)**

**P SAI SHREYA (4NN20CS034)**

# ABSTRACT

Main aim of this project is to illustrate the concepts of working of a Rain water harvesting in OpenGL. The word "Rain water harvesting" refers to a collection and storage of rain, rather than allowing it to run off. Rainwater is collected from a roof-like surface and redirected to a tank, cistern, deep pit (well, shaft, or borehole), aquifer, or a reservoir with percolation, so that it seeps down and restores the ground water.

Rainwater harvesting, the small-scale collection and storage of runoff for irrigated agriculture, is recognized as a sustainable strategy for ensuring food security, especially in monsoonal landscapes in the developing world.

Harvesting rainwater allows the collection of large amounts of water and mitigates the effects of drought. Most rooftops provide the necessary platform for collecting water.
Rainwater is mostly free from harmful chemicals, which makes it suitable for irrigation purposes. The rainwater harvesting process includes the collection and storage of collected rainwater with the help of artificially designed systems. Catchment: Used to collect and store the captured rainwater. Conveyance system: It is used to transport the harvested water from the catchment to the recharge zone.

Tamil Nadu is the first and the only state in India which has made roof top rainwater harvesting structure compulsory to all the houses across the state. The belief in Rain-gods in ancient times, in almost every culture of the world, is the sign of the significance of rain, for the earth. We all are aware of the strength of the rainwater and have learned to respect it in all possible ways. Harnessing the rain with rainwater harvesting systems, or any other known storage method is our way of using each drop of the rain to its full potential. It is the need of today's time to conserve the offerings of nature; conserving rain is one of that precious upkeep. This blog intends to summarise the quality use of rainwater and the basics of rainwater harvesting.

# TABLE OF CONTENTS

**CHAPTER 1**

# INTRODUCTION

First Come First Serve Algorithm is OS based Computer graphics Project. This Project as name suggest demonstrate the working of First Come First Serve Algorithm or FCFS Algorithm. The objects are drawn using the GLUT functions. This project has been developed using Code::Blocks IDE on Windows 10 operating system with OpenGL package.

## 1.1 Computer Graphics

Graphics provides one of the most natural means of communicating within computer since our highly developed 2D and 3D pattern-recognition abilities allow us to perceive and process pictorial data rapidly and effectively. Interactive computer graphics is the most important means of producing pictures since the invention of photography and television. It has the added advantage that, with the computer, we can make pictures not only of concrete real-world objects but also of abstract, synthetic objects, such as mathematical surfaces and of data that have no inherent geometry, such as survey results.

Computer graphics started with the display of data on hard copy plotters and cathode ray tube screens soon after the introduction of computers themselves. It has grown to include the creation, storage, and manipulation of models and images of objects. These models come from a diverse and expanding set of fields, and include physical, mathematical, engineering, architectural, and even conceptual structures, natural phenomena, and so on. Computer graphics today is largely interactive. The user controls the contents, structure, and appearance of the objects and of their displayed images by using input devices, such as keyboard, mouse, or touch-screen. Due to close relationships between the input devices and the display, the handling of such devices is included in the study of computer graphics. The advantages of the interactive graphics are many in number. Graphics provides one of the most natural means of communicating with a computer, since our highly developed 2D and 3D pattern-recognition abilities allow us to perceive and process data rapidly and efficiently. In many design, implementation, and construction processes today, the information pictures can give is virtually indispensable. Scientific visualization became an important field in the 1980s when the scientists and engineers

realized that they could not interpret the prodigious quantities of data produced in supercomputer runs without summarizing the data and highlighting trends and phenomena in various kinds of graphical representations.

# 2. OpenGL Interface

OpenGL is an application program interface (API) offering various functions to implement primitives, models and images. This offer functions to create and manipulate render lighting, colouring, viewing the models. OpenGL offers different coordinate system and frames. OpenGL offers translation, rotation and scaling of objects.

Most of our applications will be designed to access OpenGL directly through functions in three libraries. They are:

- Main GL: Library has names that begin with the letter gl and are stored in a library usually referred to as GL.
- OpenGL Utility Library (GLU): This library uses only GL functions but contains code for creating common objects and simplifying viewing
- OpenGL Utility Toolkit (GLUT): This provides the minimum functionality that should be accepted in any modern windowing system.

## 1.2.1 OpenGL Pipeline Architecture

Many OpenGL functions are used specifically for drawing objects such as points, lines, polygons, and bitmaps. Some functions control the way that some of this drawing occurs (such as those that enable antialiasing or texturing). Other functions are specifically concerned with frame buffer manipulation. The topics in this section describe how all of the OpenGL functions work together to create the OpenGL processing pipeline. This section also takes a closer look at the stages in which data is actually processed, and ties these stages to OpenGL functions.

The following diagram details the OpenGL processing pipeline. For most of the pipeline, you can see three vertical arrows between the major stages. These arrows represent vertices and the two primary types of data that can be associated with vertices: color values and texture coordinates. Also note that vertices are assembled into primitives, then into fragments, and finally into pixels in the frame buffer. This progression is discussed in more detail in Vertices, Primitives, Fragments and Pixels.

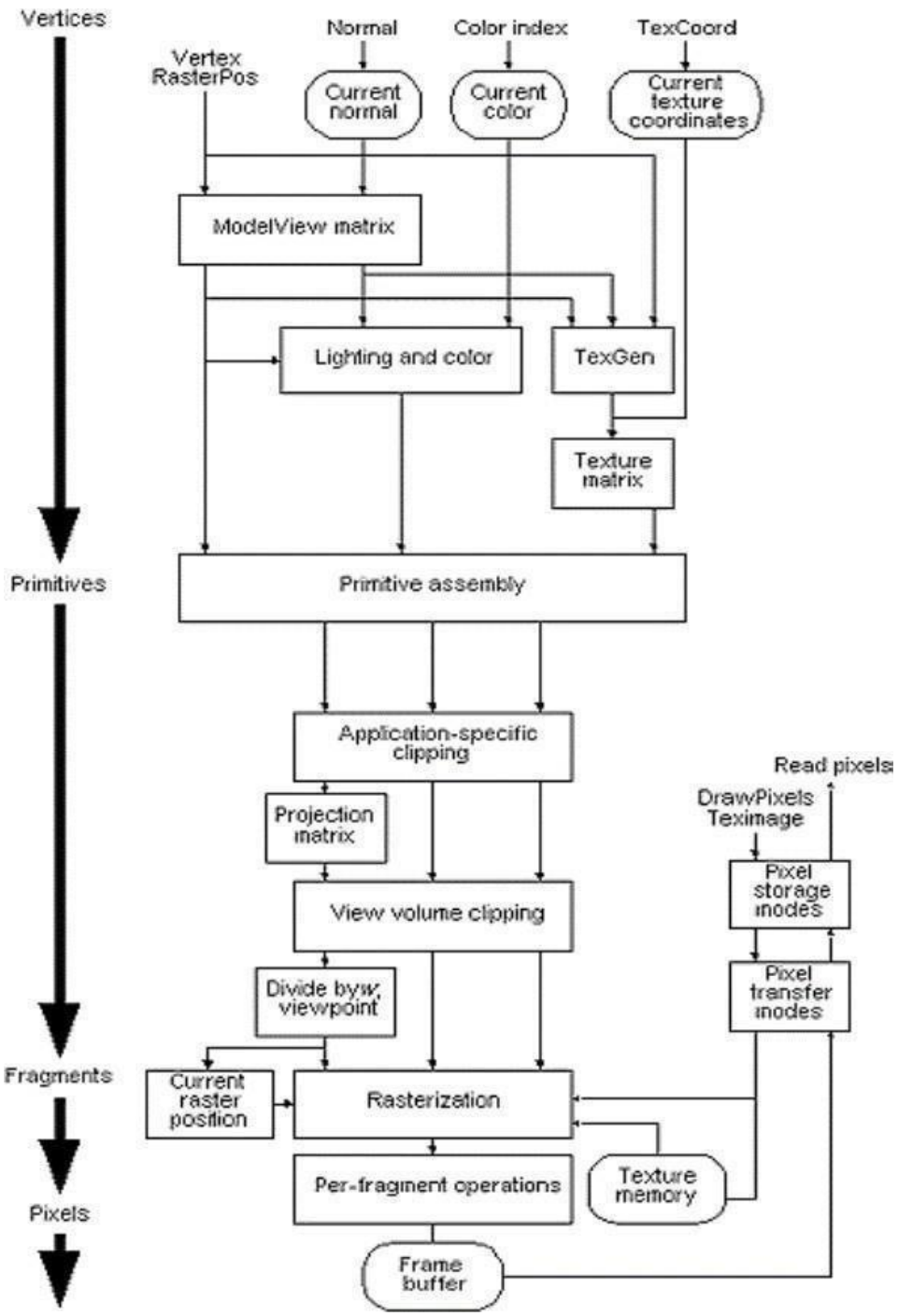


Figure. 1.1 OpenGL Pipeline Architecture

## 2. OpenGL Overview

- OpenGL (Open Graphics Library) is the interface between a graphic program and graphics hardware. It is streamlined. In other words, it provides low-level functionality. For example, all objects are built from points, lines and convex

polygons. Higher level objects like cubes are implemented as six four-sided polygons.

- OpenGL supports features like 3-dimensions, lighting, anti-aliasing, shadows, textures, depth effects, etc.
- It is system-independent. It does not assume anything about hardware or operating system and is only concerned with efficiently rendering mathematically described scenes. As a result, it does not provide any windowing capabilities.
- It is a state machine. At any moment during the execution of a program there is a current model transformation.
- It is a rendering pipeline. The rendering pipeline consists of the following steps:
    - Define objects mathematically.
    - Arranges objects in space relative to a viewpoint. ○ Calculates the color of the objects.
    - Rasterizes the objects.

Graphics provides one of the most natural means of communicating with computer, since our highly developed 2D and 3D pattern-recognition abilities allow us to perceive and process pictorial data rapidly and efficiently. Interactive computer graphics is the most important means of producing pictures since the invention of photography and television. It has the added advantage that, with the computer, we can make pictures not only of concrete real-world objects but also of abstract, synthetic objects, such as mathematical surfaces and of data that have no inherent geometry, such as survey results.

OpenGL (Open Graphics Library) is a standard specification defining a cross language cross platform API for writing applications that produce 2D and 3D computer graphics. OpenGL was developed by silicon graphics Inc. (SGI) in 1992 and is widely used in CAD, virtual reality, scientific visualization, information visualization and flight simulation. It is also used in video games.

OpenGL serves two main purposes:

- To hide the complexities of interfacing with different 3D accelerators, by presenting programmer with a single, uniform API
- To hide the differing capabilities of hardware platforms, by requiring that all Implementations support the full OpenGL, feature set.

- OpenGL has historically been influential on the development of 3D accelerator, promoting a base level of functionality that is now common in consumer level hardware:

- Rasterized points, lines and polygons are basic primitives.
- A transform and lighting pipeline.
- Z buffering.
- Texture Mapping.
- Alpha
- Blending.

**CHAPTER 2**

# SYSTEM REQUIREMENTS SPECIFICATIONS

## 2.1 Hardware Requirements

- Microprocessor: 1.0 GHz and above CPU based on either AMD or INTEL Microprocessor Architecture.
- Main memory: 2 GB RAM
- Hard Disk: 40 GB
- Hard disk speed in RPM: 5400 RPM
- Keyboard: QWERTY Keyboard
- Mouse: 2 or 3 Button mouse
- Monitor: 1024 x 768 display resolution

## 2. Software Requirements

- Programming language – C/C++ using OpenGL
- Operation system – Windows 8 or above
- Compiler/IDE – Code::Blocks IDE
- Graphics library – GL/glut.h
- OpenGL 2.0

## 3. Functional Requirements

### ➢ OpenGL APIs:

If we want to have a control on the flow of program and if we want to interact with the window system then we use OpenGL API'S. Vertices are represented in the same manner internally, whether they are specified as two-dimensional or three-dimensional entities, everything that we do are here will be equally valid in three dimensions. Although OpenGL is easy to learn, compared with other APIs, it is nevertheless powerful. It supports the simple three-dimensional programs and also supports the advanced rendering techniques.

### ➢ GL/glut.h:

We use a readily available library called the OpenGL Utility Toolkit (GLUT), which provides the minimum functionality that should be expected in any modern windowing system. The application program uses only GLUT functions and can be recompiled with

the GLUT library for other window system. OpenGL makes a heavy use of macros to increase code readability and avoid the use of magic numbers. In most implementation, one of the include lines.

## ➢ Code::Blocks Integrated Development Environment (IDE):

Code::Blocks is a free, open-source cross-platform IDE that supports multiple compilers including GCC, Clang and Visual C++. It is developed in C++ using wxWidgets as the GUI toolkit. Using a plugin architecture, its capabilities and features are defined by the provided plugins. Currently, Code::Blocks is oriented towards C, C++, and FORTRAN. It has a custom build system and optional Make support. Code::Blocks is being developed for Windows, Linux, and MacOS and has been ported to FreeBSD, OpenBSD and Solaris.

# CHAPTER 3

# ABOUT THE PROJECT

Main aim of this project is to illustrate the concepts of working of rain water harvesting in OpenGL. The objects are drawn using the GLUT functions. This project has been developed using Code::Blocks IDE on Windows 10 operating system with OpenGL package.

## 3.1 Overview

Rain water harvesting is collection and storage of rain water that runs off from roof tops, parks, roads, open grounds, etc. This water run off can be either stored or recharged into the ground water.

No, we have not and it is hard to imagine. We all use water for different kinds of day to day activities, such as cleaning, washing, bathing, cooking, drinking and other domestic and industrial uses.

Water is a precious, essential and an abiotic component of the ecosystem. Today we all are heading toward the scarcity of water, and this is mainly because of the lack of water conservation and pollution of water bodies. So, let us not waste a drop of water and start conserving water for further use.


**Fig 3.1 Rain water harvesting overview.**

Rainwater harvesting systems consists of the following components:

- Catchment- Used to collect and store the captured rainwater.
- Conveyance system – It is used to transport the harvested water from the catchment to the recharge zone.
- Flush- It is used to flush out the first spell of rain.
- Filter – Used for filtering the collected rainwater and removing pollutants.

- Tanks and the recharge structures: Used to store the filtered water which is ready to use.

The process of rainwater harvesting involves the collection and the storage of rainwater with the help of artificially designed systems that run off naturally or man-made catchment areas like- the rooftop, compounds, rock surface, hill slopes, artificially repaired impervious or semi-pervious land surface.

Several factors play a vital role in the amount of water harvested. Some of these factors are:

- The quantum of runoff
- Features of the catchments
- Impact on the environment
- Availability of the technology
- The capacity of the storage tanks
- Types of the roof, its slope and its materials
- The frequency, quantity and the quality of the rainfall
- The speed and ease with which the rainwater penetrates through the subsoil to recharge the groundwater.

**Fig 3.2 Real time example of rain water harvesting.**

## 2. User Interface

- ❖ Mouse and key board to interact with program.
- ❖ User can view both artificial satellites revolving around Earth as well as by making use of keyboard user can interact with program.

## 3. Objective

Reducing groundwater contamination. Reduce flooding and erosion. Avoiding flooding of roads. Avoiding flooding of roads. The main purpose of the rainwater harvesting is to use the locally available rainwater to meet water requirements throughout the year without the need of huge capital expenditure. This would facilitate the availability of uncontaminated water for domestic, industrial, and irrigation needs.

## 4. Benefits

Three main uses of rain water harvesting are:

- Less cost.
- Helps in reducing the water bill.
- Decreases the demand for water.
- Reduces the need for imported water.
- Promotes both water and energy conservation.
- Improves the quality and quantity of groundwater.
- Does not require a filtration system for landscape irrigation.

**CHAPTER 4**

# SYSTEM DESIGN

The System Design Document describes the system requirements, operating environment, system and subsystem architecture, files and database design, input formats, output layouts, human-machine interfaces, detailed design, processing logic, and external interfaces.

## 1. Documentation

- **Purpose and Scope**

  This section provides a brief description of the Systems Design Document's purpose and scope.

- **Project Executive Summary**

  This section provides a description of the project from a management perspective and an overview of the framework within which the conceptual system design was prepared. If appropriate, include the information discussed in the subsequent sections in the summary.

- **System Overview**

  This section describes the system in narrative form using non-technical terms. It should provide a high-level system architecture diagram showing a subsystem breakout of the system, if applicable. The high-level system architecture or subsystem diagrams should, if applicable, show interfaces to external systems. Supply a high-level context diagram for the system and subsystems, if applicable. Refer to the requirements trace ability matrix (RTM) in the Functional Requirements Document (FRD), to identify the allocation of the functional requirements into this design document.

- **Design Constraints**

  This section describes any constraints in the system design (reference any trade-off analyses conducted such, as resource use versus productivity, or conflicts with other systems) and includes any assumptions made by the project team in developing the system design.

- **Future Contingencies**

  This section describes any contingencies that might arise in the design of the system that may change the development direction. Possibilities include lack of

interface agreements with outside agencies or unstable architectures at the time this document is produced. Address any possible workarounds or alternative plans.

- **Document Organization**
  This program includes interaction with keyboard.
  a) s/S – To Start the project.
  b) t/T – To transmit and receive signals.
  c) q/Q – Quit

## 4.2 Flow Chart

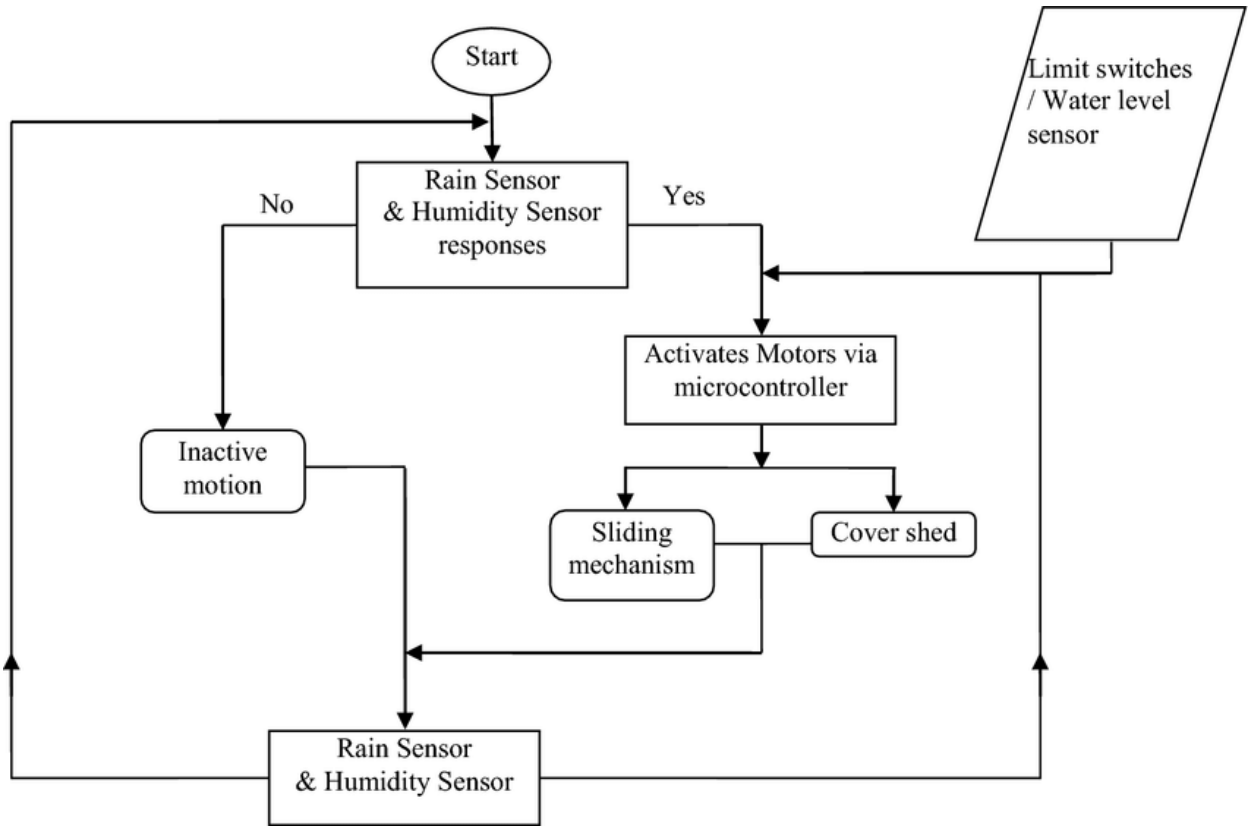


**Figure 1.2: flow chart of working of rain water harvesting**

# CHAPTER 5

# IMPLEMENTATION

## 1. OpenGL functions

- **glutInit():** Interaction between the windowing system and OpenGL is initiated.
- **glutInitDisplayMode():** Used when double buffering is required and depth information is required.
- **glutCreateWindow():** This opens the OpenGL window and displays the title at top of the window.
- **glutInitWindowSize():** Specifies the size of the window.
- **glutInitWindowPosition():** Specifies the position of the window in screen co-ordinates
- **glutKeyboardFunc():** This function handles normal ascii symbols.
- **glutSpecialFunc():** This function handles special keyboard keys.
- **glutReshapeFunc():** Setup the call back function for reshaping the window.
- **glutIdleFunc():** This handles the processing of the background.
- **glutMainLoop():** This starts the main loop, and it never returns.
- **glViewport():** Used to setup the viewport.
- **glVertex3fv():** Used to setup the points or vertices in three dimensions.
- **glColor3fv():** Used to render color to faces.
- **glFlush():** Used to flush the pipeline.
- **glutPostRedisplay():** Used to trigger an automatic redraw of the project.
- **glMatrixMode():** Used to setup the required mode of the matrix.
- **glLoadIdentity():** Used to load or initialize to the identity matrix.
- **glTranslatef():** Used to translate or move the rotation centre from one point to another in three dimensions.
- **glRotatef():** Used to rotate an object through a specified rotation angle.
- **glBegin() , glEnd() :** delimit the vertices of a primitive or a group of like primitives.

# CHAPTER 6

# TESTING

## Introduction to Testing

Verification and validation are a generic name given to checking processes, which ensures that the software conforms to its specifications and meets the demands of users.

- **Validation**
  Are we building the right product? Validation involves checking that the program has implanted meets the requirement of the users.
- **Verification**
  Verification involves checking that the program confirms to its specification.

## Stages in the Implementation of Testing

- **Unit testing**
  Each individual unit is tested for correctness. These individual components will be tested to ensure that they operate correctly.
- **Module testing**
  A module is a collection of dependent components such as a function. A module encapsulates related components so can test without other system modules.
- **Sub-system Testing**
  This phase involves testing collection of modules, which have been integrated into sub-systems. Sub-systems may be independently designed and implemented.
- **System testing**
  The sub-systems are integrated to make up the entire system. The errors that result from unanticipated interaction between sub-systems and system components are removed.
- **Acceptance testing**
  This is the final stage in the testing process before the system is tested for operational use. Any requirement problem or requirement definition problem revealed from acceptance testing are considered and made error free.
- **Test plan**
  Careful planning is needed to the most of testing and controlled testing cost.

# Results

Several errors were detected and rectified and the whole project is working as it should have to work with proper output and high efficiency.

| Test case ID | Test case | Steps to execute the test case | Expected result | Actual result | Pass/Fail |
|---|---|---|---|---|---|
| 1 | Front screen | Press any key | Remains in the front screen | Remains in the front screen | Pass |
| 2 | Front screen | Right click | Menu should appear | Menu appear | Pass |
| 3 | Front/Another screen | Select S option from menu | Should press enter to house | View of house with raining appears | Pass |
| 4 | Front/Another screen | Select R or r option from menu | Should open collecting rain water view | Opens the view of storage of water in container | Pass |
| 5 | Front/Another screen | Select q bar option from menu | Program terminated | Program terminated | Pass |

Table 1. Testing for the “Working of a Rain water harvesting” Project

# CHAPTER 7

# SNAPSHOT


**Figure 7.1 Displaying the functions to start.**

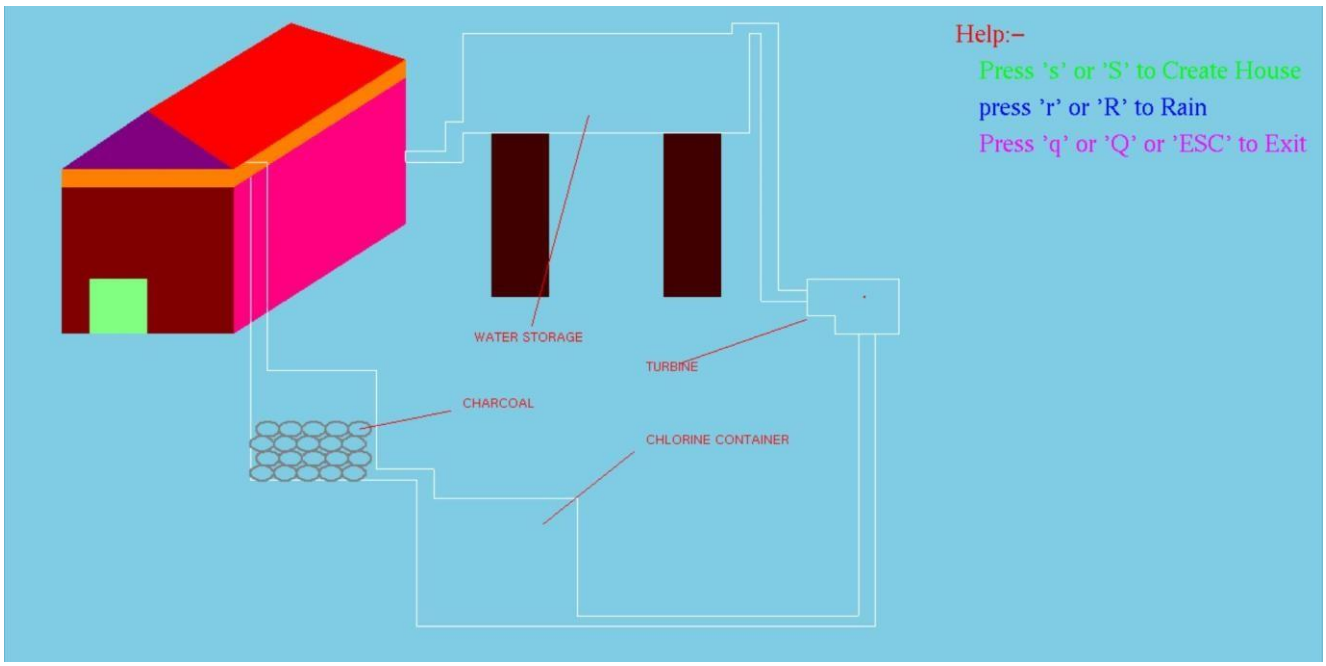


**Figure 7.2 Displaying the rain water harvesting setup.**

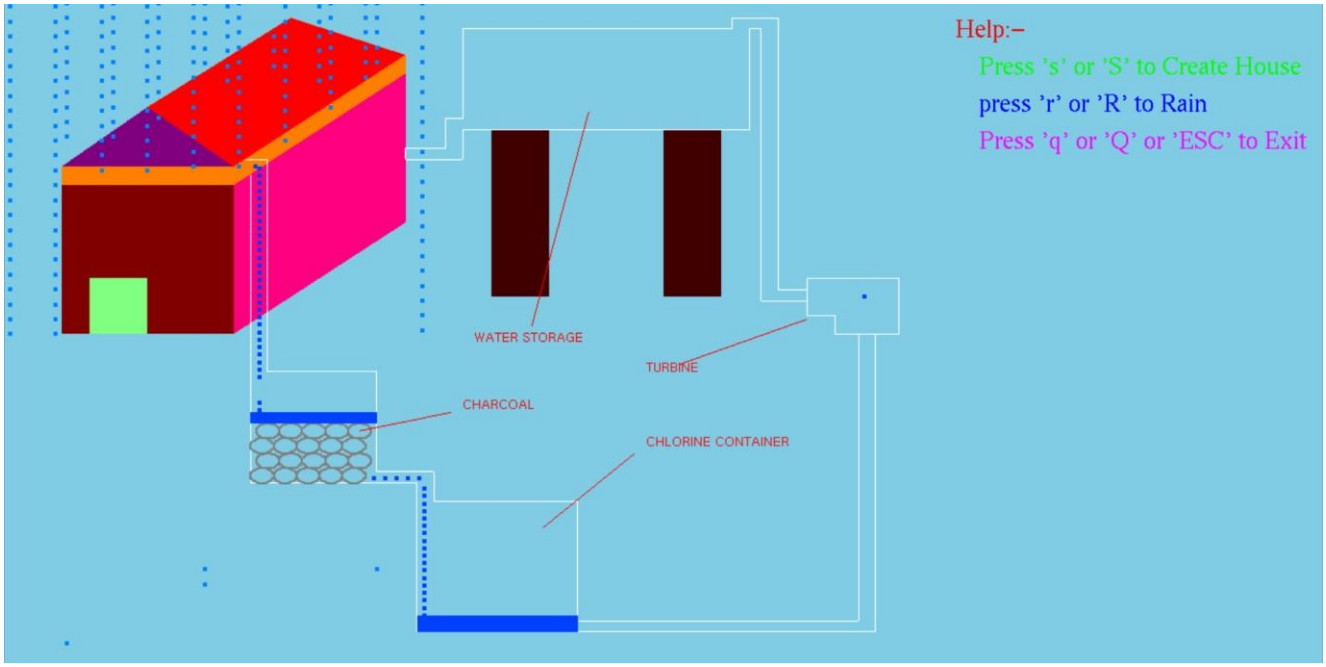


**Figure 7.3 Displaying the rain falling and collecting it in container.**

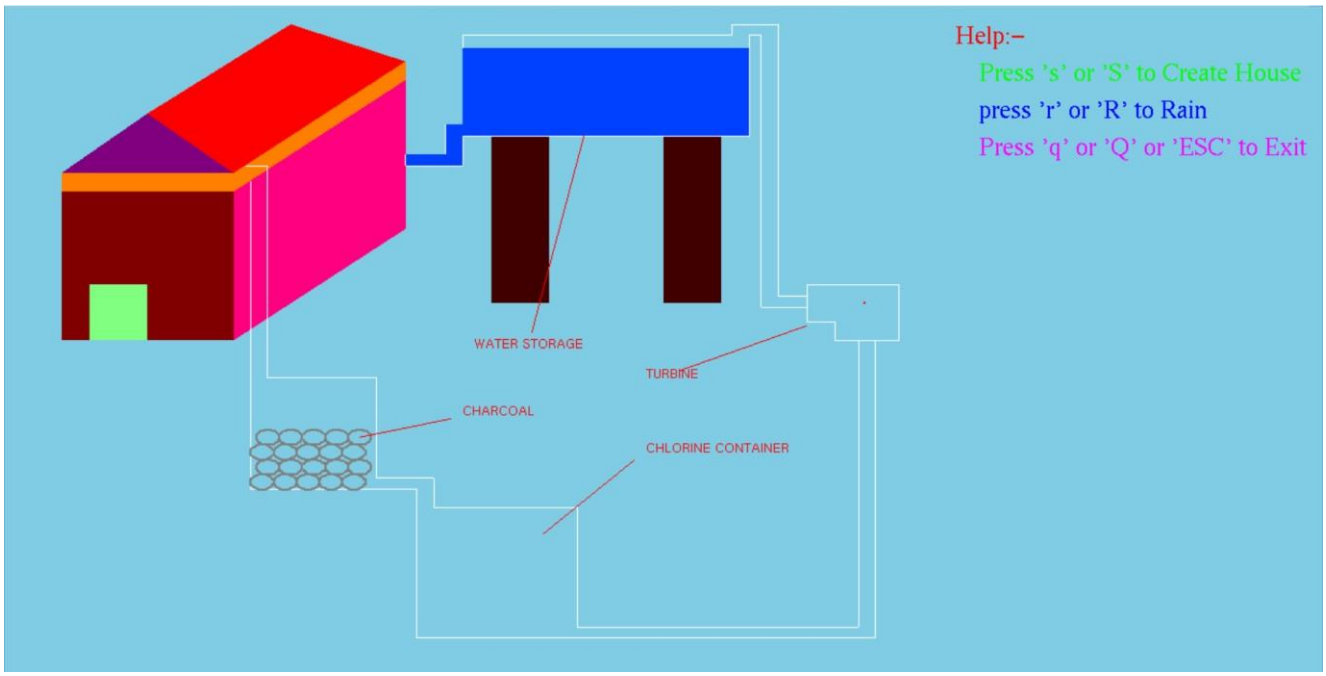


**Figure 7.4 Sending the collected water to storage for further us**

# CHAPTER 8
# CONCLUSION AND FUTURE ENHANCEMENT

## CONCLUSION

In conclusion, this project presents a visual implementation of working process of a Rain water harvesting. This project demonstrates how rain water harvesting is collection and storage of rain water that runs off from roof tops, parks, roads, open grounds.

This package is very useful for the user since it provides the basic information about various OpenGL functions and its component utilities. This is an interactive project which has user friendly interaction given through keyboard. Thus, this project meets the basic requirements successfully and is flexible in all respects to one and all.

## FUTURE ENHANCEMENT

The future enhancement that can be made include:

- Keyboard interaction can be improvised.
- The project can be further developed to change the colour of the objects and also, we can give the background effect as well.

## CHAPTER 9

# BIBLOGRAPHY

## Books:


- Edward Angel's Interactive Computer Graphics Pearson Education 5th Edition.
- Computer Graphics Using OpenGL – F.S. Hill, Jr. 2nd Edition, Pearson Education, 2001.
- Computer Graphics – James D Foley, Andries Van Dam, Steven K Feiner, John F Hughes, Addison-Wesley 1997.
- Computer Graphics - OpenGL Version – Donald Hearn and Pauline Baker, 2nd Edition, Pearson Education, 2003


## Websites for reference


- https://opengl.org
- https://stackoverflow.com
- https://wikipedia.org
- www.google.com
- www.geeksforgeeks.org