VISVESVARAYA TECHNOLOGICAL UNIVERSITY

BELAGAVI - 590018

A PROJECT REPORT

ON

"IMPLEMENTING THE CONCEPT OF SOLAR POWER AND ENERGY"

BY

ALLAN JOSTON FERNANDES	**YASH KORADIYA**
4SF18CS009	**4SF18CS067**

In the partial fulfillment of the requirement for VI Sem. B. E. (CSE)

COMPUTER GRAPHICS LABORATORY WITH MINI PROJECT (18CSL67)

Under the guidance of

Ms. Megha Rani R
Assistant Professor, Dept. of CS&E

Department of Computer Science & Engineering

SAHYADRI
COLLEGE OF ENGINEERING & MANAGEMENT
Adyar, Mangaluru-575007
2020-2021

SAHYADRI
COLLEGE OF ENGINEERING & MANAGEMENT

(Affiliated to Visvesvaraya Technological University, BELAGAVI)

Adyar, Mangaluru – 07

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that the project entitled **“IMPLEMENTING THE CONCEPT OF SOLAR POWER AND SOLAR ENERGY”** is submitted in partial fulfillment for the requirement of VI Sem. B.E. (Computer Science & Engineering), **“COMPUTER GRAPHICS LABORATORY WITH MINI PROJECT”** during the year 2020 – 2021 is a result of bonafide work carried out by

ALLAN JOSTON FERNANDES	**4SF18CS009**
YASH KORADIYA	**4SF18CS067**

…………………………

Ms. Megha Rani R
Asst. Prof .Dept. of CS&E
SCEM, Mangaluru

..………………………

Dr. Pushpalatha K
HOD ,Dept. of CSE&E
SCEM, Mangaluru

Signature of the Examiners

1. **…………………………..**

2. **…………………………..**

ABSTRACT

Computer graphics is an art of drawing pictures, lines, charts using computer with the help of programming. Computer graphics is made up of number of pixels. Pixel is the smallest graphical picture or unit represented on the computer screen. Basically, there are two types of computer graphics namely. Usually, the term refers to computer generated image data created with the help from specialized graphical hardware and software. It is a vast and recent area in computer science.

Our objective is to simulate the working of solar energy and power in real life situation using OpenGL library. In this simulation, we try to show normal events that happen everyday in real life. We are using different OpenGL transformations to implement these events.

We have made use of Code::Blocks, C language and OpenGL library functions to implement our project. This is because it allows graphics to run and manage on various applications and multiple platforms. It is easier to draw objects, assign colors, zoom in and zoom out, transforming objects. This allows us to simulate our project.

ACKNOWLEDGEMENT

In the accomplishment of this project successfully, many people have best owned upon us their blessings and their heart pledged support, this time we are utilizing to thank all the people who have been concerned with this project.

Primarily we would thank god for being able to complete this project with success. Then we would like to thank our project guide **Ms. Megha Rani R, Assistant Professor, Department of Computer Science and Engineering,** Sahyadri College Of Engineering And Management for their constant encouragement and support extended throughout.

We owe our deep gratitude to **Dr. Pushpalatha K, Head of the Department, Computer Science and Engineering** for her invaluable support and guidance.

We express our sincere gratitude to **Dr. Rajesha S, Principal, Sahyadri College Of Engineering And Management** who have also been a great source of inspiration.

We would also like to thank all our non-teaching staffs who also were very much supportive to us in building this project.

Last but not the least we want to extend our gratitude to our parents for their continuous love and support for which we are always indebted to them and also thank each and every one of those who helped or provided a helping hand in this project to be carried out.

ALLAN JOSTON FERNANDES	**4SF18CS009**
YASH KORADIYA	**4SF18CS067**

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1. COMPUTER GRAPHICS

Computer graphics is a sub-field of computer science and is concerned with digitally synthesizing and manipulating visual content. Although the term refers to three-dimensional computer graphics, it is also encompasses two-dimensional graphics and image processing. Computer graphics is often differentiated from field of visualization, although two have some similarities. Graphics are visual presentation on some surface like wall, canvas, computer screen. Graphics often combine text, illustration and color.

Computer graphics started with the display of data on hardcopy plotters and cathode ray tube (CRT) screens soon after the introduction of computers. It has grown to include the creation, storage and manipulation models and images of objects. These models come from a diverse and expanding set fields, and include physical, mathematical, engineering, architectural and even conceptual structures. Computer graphics today is largely interactive. The user controls the contents, structure and appearance of objects and of their displayed images by using devices such as keyboard, mouse or touch-sensitive panel on screen.

1.2 OpenGL

Open Graphics Library (OpenGL) is cross-language, cross-platform application programming interface (API) for rendering 2D and 3D vector graphics. The API is typically used to interact with a graphics processing unit (GPU), to achieve hardware-accelerated rendering.

Silicon Graphics Inc, (SGI) started developing OpenGL in 1991 and released it in January 1992, applications use it extensively in the fields of computer-aided design (CAD), virtual reality, scientific visualization, information visualization, flight simulation, and video games. Since 2006 OpenGL has been managed by the non-profit technology consortium Khronos Group.

One aspect of OpenGL that suits it so well for use in computer graphics course is its device independence or probability. A student can develop and run program on any available computer.

OpenGL offers rich and highly usable API for 2D graphics and image manipulation, but its real power emerges with 3D graphics.

A sophisticated library that provides these features could certainly be build on top of OpenGL. The OpenGL utility library (GLU) provides many of the modeling features, such as quadric surfaces and the NURBES curves. GLU is a standard part of every OpenGL implementation.

OpenGL is designed to be independent of the windowing system or operating system. GLUT is needed to interact with the operating system. GLUT commands start with a prefix of "glut". GLUT is platform independent, which is built on top of platform -specific OpenGL extension such as GLX for X windowing system. GLUT is designed to construct small and medium sized OpenGL programs.

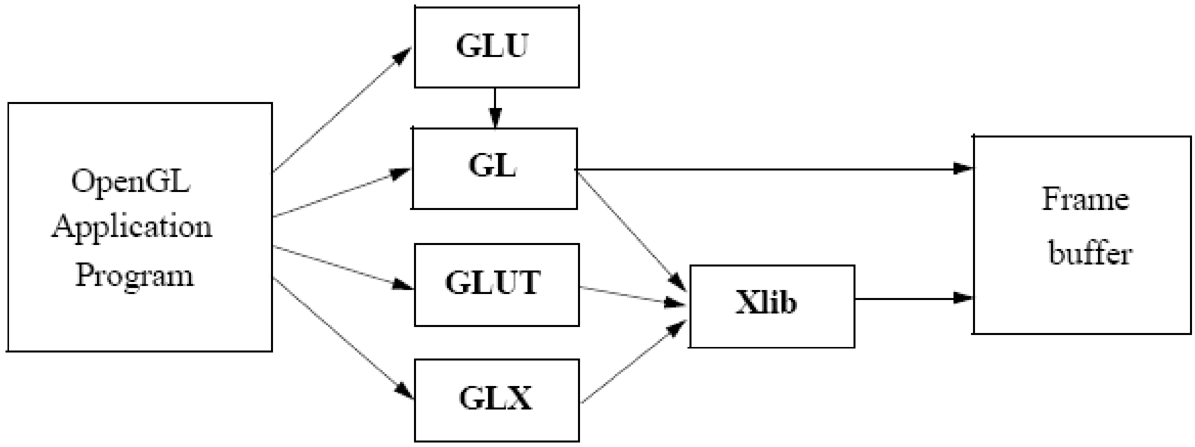


Fig 1.2 OpenGL Architecture

1.3 ABOUT PROJECT

In this project we try to simulate the working of how solar energy is stored and then converted to electrical energy using OpenGL library. In this simulation, we try to show the benefits of using solar energy and how we can reduce electricity consumption costs. We are using different OpenGL transformations to implement this.

This project mainly focuses on how solar energy and power can be used effectively and efficiently in our daily lives. People living in rural areas or remote areas have the mindset that it is costly to implement solar in their houses. We want to make a common person understand what its uses are and how it works actually. By understanding how it works, we can think of implementing it within our houses so that the advantage we can have is that it’s very cost effective and also the power it stores in its panel is the sunlight which is a free natural resource available to us. This project overall shows how solar energy is stored and converted from heat energy to electrical energy and also how it is useful especially during the night.

OpenGL is a low-level graphics library specification. It makes available to the programmer a small set of geometric primitives - points, lines, polygons, images and bitmaps. OpenGL provides a set of commands that allow the specification of geometric objects in two or three dimensions, using the provided primitives, together with commands that control how these objects are rendered (drawn).

In this project we use OpenGL library to demonstrate the use of solar energy. We implement it using C language. Here we intend to show what is solar energy, its uses and how it works.

The project will contain movement options via keyboard. The user will land in the project name page when he executes the project from where he can navigate to other screens and the view the simulation. In the last screen, at a certain point the execution stops automatically.

CHAPTER 2

REQUIREMENT SPECIFICATION

The basic purpose of software requirement specification (SRS) is to bridge the communication between the parties involved in the development project. SRS is the medium through which the user’s needs are accurately specified; indeed, SRS forms the basis of the software development. Another important purpose of developing an SRS is helping the users understand their own needs.

Now we will be discussing the requirement analysis of the project. This report gives the description of the roles of users, the functional overviews of the project, input and output characteristics and also the hardware and software for the project.

2.1 Hardware Requirements

- Intel Pentium CPU 2.6 GHz or AMD Athlon 64 (K8) 2.6 GHz or higher
- RAM – 512MB (minimum)
- Hard Disk – 1MB (minimum)
- Mouse
- Keyboard 108 standard
- Monitor resolution 800×600

2.2 Software Requirements

- Programming Language: C / C++ using OpenGL
- Operating System: Windows / Ubuntu
- Compiler: C / C++ Compiler (GCC)
- IDE: Code Blocks / Visual Studio 2019 or 2013
- Functional Requirements: GLUT

CHAPTER 3

SYSTEM DESIGN

3.1 FLOWCHART

A flowchart is a common type of chart that represents an algorithm or process showing the steps as boxes of various kinds and their order by connecting these with arrows. Flowcharts are used in analyzing, designing, managing a process or program in various fields. Flowchart used to be a popular means for describing computer algorithms. They are still used for the purpose, modern techniques such as UML activity diagrams.

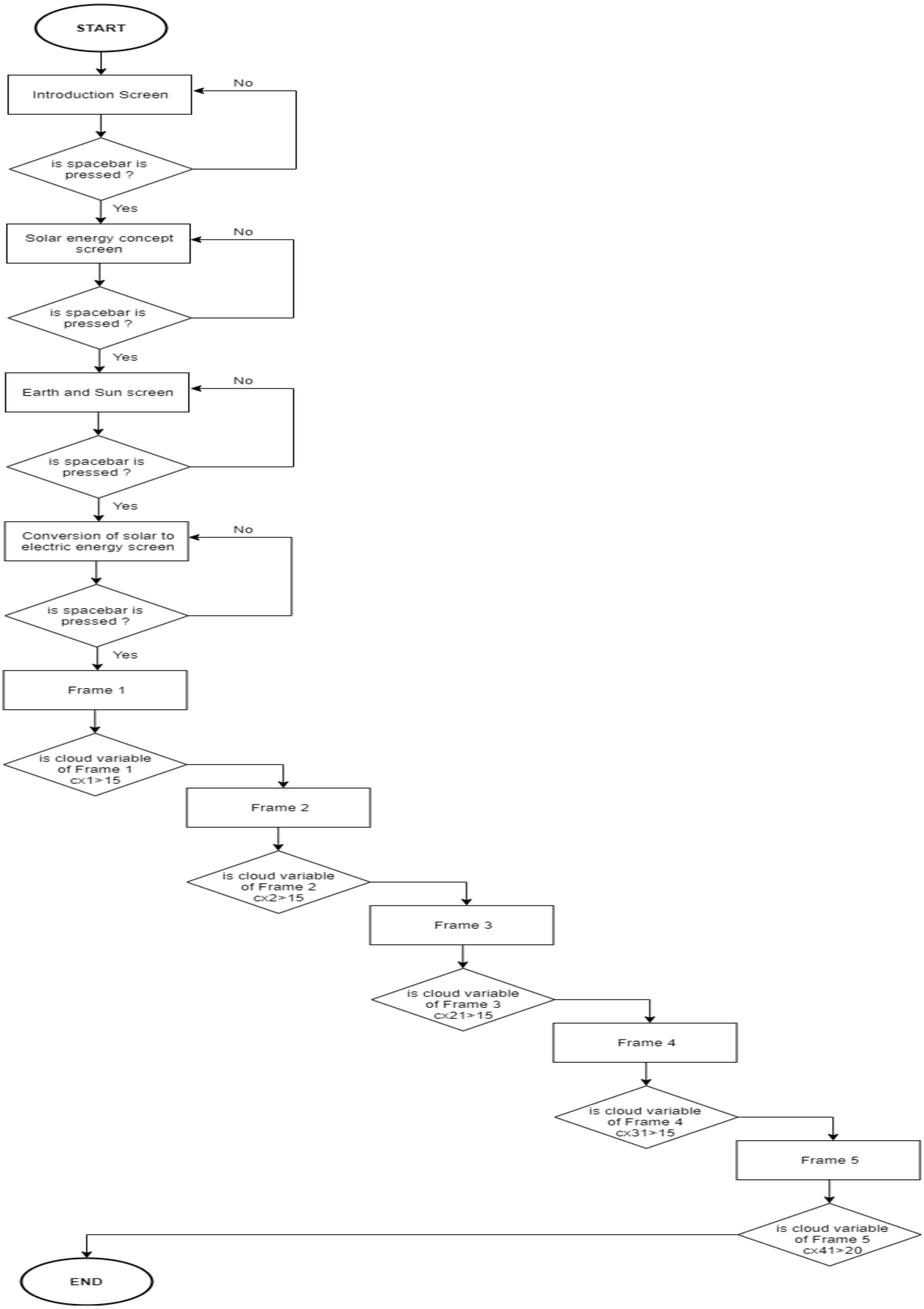

START
Introduction Screen
No
is spacebar is pressed ?
Yes
Solar energy concept screen
No
is spacebar is pressed ?
Yes
Earth and Sun screen
No
is spacebar is pressed ?
Yes
Conversion of solar to electric energy screen
No
is spacebar is pressed ?
Yes
Frame 1
is cloud variable of Frame 1 cx1>15
Frame 2
is cloud variable of Frame 2 cx2>15
Frame 3
is cloud variable of Frame 3 cx21>15
Frame 4
is cloud variable of Frame 4 cx31>15
Frame 5
is cloud variable of Frame 5 cx41>20
END

Fig 3.1 Flowchart

CHAPTER 4

IMPLEMENTATION

4.1 OpenGL Functions

4.1.1 Specifying Geometry

void glBegin (glEnum mode)

Initiates a new primitive of type mode and starts the collection of vertices. Values of mode include GL_POINTS, GL_LINES and GL_POLYGON.

void glEnd( )

Terminates a list of vertices.

4.1.2 Attributes

void glClearColor (GLclampf r, GLclampf g, GLclampf b, GLclampf a)

Sets the present RGBA clear color used when clearing the color buffer. Variables of GLclampf are floating-point numbers between 0.0 and 1.0.

void glPointSize(GLfloat size)

Sets the point size attribute in pixels.

4.1.3 Working with the window

void glFlush( )

Forces any buffered any OpenGL commands to execute.

void glutInit (int argc, char **argv)

Initializes GLUT. The arguments from main are passed in and can be used by the application.

int glutCreateWindow(char *title);

Creates a window on the display. The string title can be used to label the window. The return value provides a reference to the window that can be used where there are multiple windows.

void glutInitWindowSize(int width, int height)

Specifies the initial height and width of the window in pixels.

void glutInitWindowPosition(int x, int y)

Specifies the initial position of the top-left corner of the window in pixels

void glutMainLoop( )

Cause the program to enter an event processing loop. It should be the last statement in main.

void glutDisplayFunc(void (*func) (void))

Registers the display function func that is executed when the window needs to be redrawn.

void glutPostRedisplay()

Requests that the display callback be executed after the current callback returns.

4.1.4 Interactions

void glutKeyboardFunc(void *f(char key, int width, int height))

Registers the keyboard callback function f. The callback function returns the ASCII code of the key pressed and the position of the mouse.

4.1.5 Enabling Features

void glEnable(GLenum feature)

Enables an OpenGL feature. Features that can be enabled include GL_DEPTH, GL_LIGHTING, GL_TEXTURE_ID, GL_TEXTURE_2D, GL_TEXTURE_3D, GL_LINE_SMOOTH, GL_POLYGON_SMOOTH, GL_POINT_SMOOTH, GL_BLEND, GL_LINE_STIPPLE, GL_POLYGON_STIPPLE, GL_NORMALIZE.

void glDisable(GLenum feature)

Disables an OpenGL feature.

4.1.6 Transformations

void glMatrixMode(GLenum mode)

Specifies which matrix will be affected by subsequent transformations. Mode can be GL_PROJECTION, GL_MODELVIEW or GL_TEXTURE.

void glLoadIdentity()

Sets the current transformation matrix to an identity matrix.

void glPushMatrix & void PopMatrix()

Pushes and Pops from the matrix stack corresponding to the current matrix mode.

void glRotate[fd](TYPE angle, TYPE dx, TYPE dy, TYPE dz)

Alters the current matrix by a rotation of angle degrees about the axis (dx,dy,dz).

void glTranslate[fd](TYPE x, TYPE y, TYPE z)

Alters the current matrix by a displacement of (x,y,z).

void glScale[fd](TYPE sx, TYPE sy, TYPE sz)

Alters the current matrix by a scaling of (sx,sy,sz).

gluOrtho2D(GLdouble left, GLdouble right, GLdouble bottom, GLdouble top)

Defines a two-dimensional viewing rectangle in the plane Z=0.

4.1.7 Viewing

gluOrtho2D(GLdouble left, GLdouble right, GLdouble bottom, GLdouble top, GLdouble near, GLdouble far)

Defines a 3D orthographic viewing volume with all parameters measured from the center of the projection plane.

4.2 User defined functions

We have used many user defined functions for the convenience in translating and using other basic functions.

- **void circle()** To draw a normal circle. It is used to draw 2 wheels of a car as well as earth and sun.
- **void bitmap_output()** To display the character text into the display window. For eg: "SAHYADRI COLLEGE OF ENGINEERING AND MANAGEMENT".
- **void mydisplay()** To display the text from the "INTRODUCTION" screen on the display window.
- **void mydisplay1()** To display the text from "SOLAR POWER AND ENERGY CONCEPT" screen on the display window.
- **void mydisplay2()** To draw earth and the sun with the arrows on the display window.
- **void mydisplay3()** To draw conversion of solar energy into electric energy with the arrows on the display window.

The below functions are used for Frame 1.

- **void cloudB()** To draw left, right, top left and top right clouds which are of large size.
- **void a()** To draw left, right, top left and top right clouds which are of small size (mini clouds).
- **void car()** To draw a car which includes car bottom body part, car top body part, car top middle body part, window, car bottom part, left wheel and the right wheel.
- **void c()** To draw a single cloud.
- **void cloud()** To draw 3 single clouds together.
- **void road()** To draw a road which includes footpath below, road lower, road center, road upper and footpath above.
- **void tution()** To draw 3 buildings, the Office, Apartment and a House.

- **void display1()** To move the cloud and the car in the first frame.
- **void spindisplay1()** To control the speed of the cloud and the car in the first frame and then move to the next frame when one of the clouds reaches to some coordinate position.

The below functions are used for Frame 2.

- **void cloudB2()** To draw left, right, top left and top right clouds which are of large size.
- **void a2()** To draw left, right, top left and top right clouds which are of small size (mini clouds).
- **void car2()** To draw a car which includes car bottom body part, car top body part, car top middle body part, window, car bottom part, left wheel and the right wheel.
- **void c2()** To draw a single cloud.
- **void cloud2()** To draw 3 single clouds together.
- **void road2()** To draw a road which includes footpath below, road lower, road center, road upper and footpath above.
- **void tution2()** To draw 3 buildings, the Office, Apartment and a House.
- **void display2()** To move the cloud and the car in the second frame.
- **void spindisplay2()** To control the speed of the cloud and the car in the second frame and then move to the next frame when one of the clouds reaches to some coordinate position.

The below functions are used for Frame 3.

- **void road3()** To draw a road which includes footpath below, road lower, road center, road upper and footpath above.
- **void tution3()** To draw 3 buildings, the Office, Apartment and a House.
- **void display3()** To move the cloud and the car in the third frame.
- **void spindisplay3()** To control the speed of the cloud and the car in the third frame and then move to the next frame when one of the clouds reaches to some coordinate position.

The below functions are used for Frame 4.

- **void road4()** To draw a road which includes footpath below, road lower, road center, road upper and footpath above.
- **void tution4()** To draw 3 buildings, the Office, Apartment and a House.

The below functions are used for Frame 5.

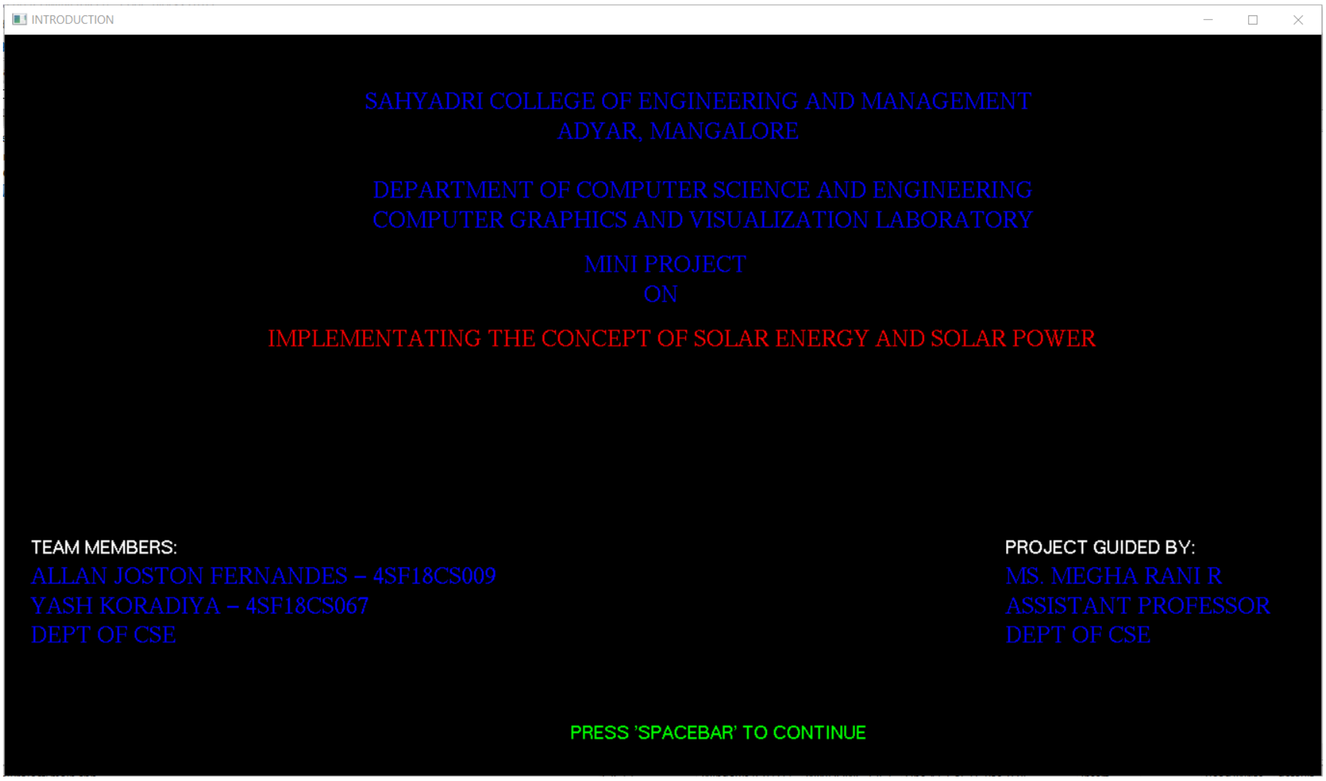
- **void road5()** To draw a road which includes footpath below, road lower, road center, road upper and footpath above.
- **void tution5()** To draw 3 buildings, the Office, Apartment and a House.
- **void solar5()** To draw a solar panel with all the borders.
- **void display4()** To move the cloud and the car in the fourth frame.
- **void spindisplay4()** To control the speed of the cloud and the car in the fourth frame and then move to the next frame when one of the clouds reaches to some coordinate position.

The below functions are used for Frame 5.

- **void road6()** To draw a road which includes footpath below, road lower, road center, road upper and footpath above.
- **void tution6()** To draw 3 buildings, the Office, Apartment and a House.
- **void solar6()** To draw a solar panel with all the borders.
- **void display5()** To move the cloud and the car in the fifth frame.
- **void spindisplay5()** To control the speed of the cloud and the car in the fifth frame and then when one of the clouds reaches to some coordinate position it exits from the program.

CHAPTER 5

RESULTS



SAHYADRI COLLEGE OF ENGINEERING AND MANAGEMENT
ADYAR, MANGALORE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
COMPUTER GRAPHICS AND VISUALIZATION LABORATORY

MINI PROJECT
ON

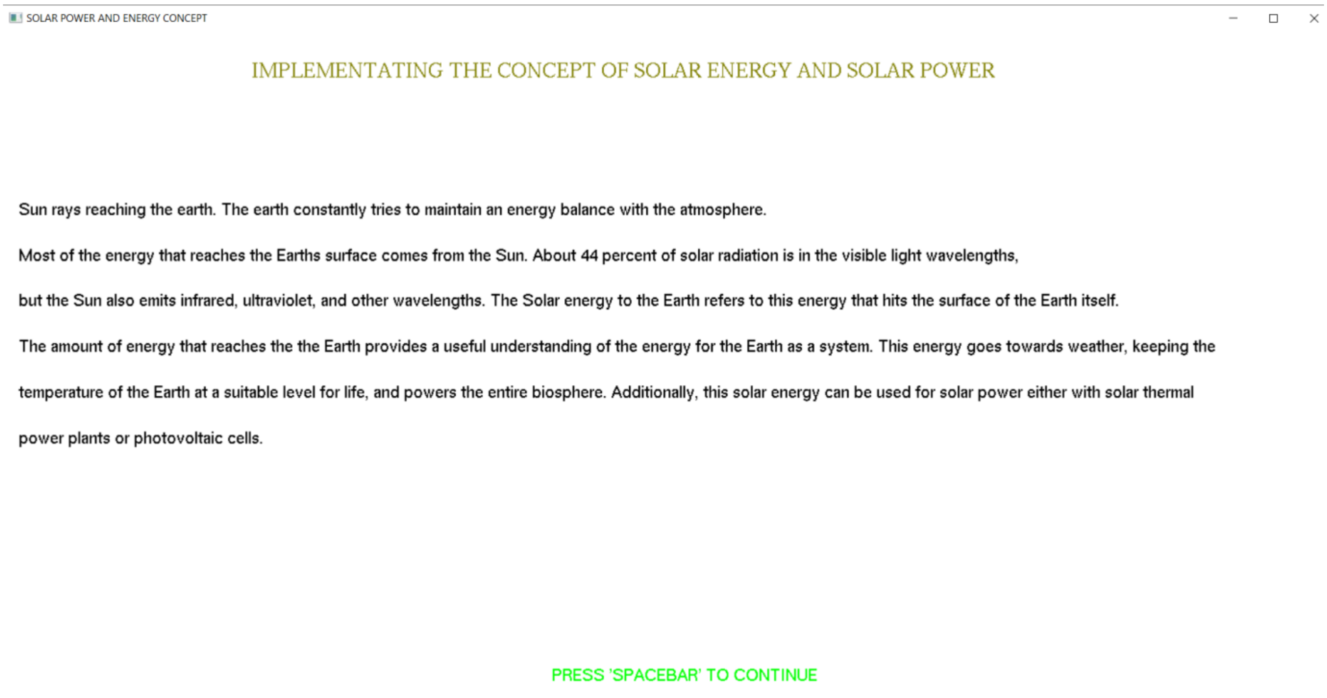
IMPLEMENTATING THE CONCEPT OF SOLAR ENERGY AND SOLAR POWER

TEAM MEMBERS:
ALLAN JOSTON FERNANDES – 4SF18CS009
YASH KORADIYA – 4SF18CS067
DEPT OF CSE

PROJECT GUIDED BY:
MS. MEGHA RANI R
ASSISTANT PROFESSOR
DEPT OF CSE

PRESS 'SPACEBAR' TO CONTINUE

Fig 5.1 Introduction Page



IMPLEMENTATING THE CONCEPT OF SOLAR ENERGY AND SOLAR POWER

Sun rays reaching the earth. The earth constantly tries to maintain an energy balance with the atmosphere.

Most of the energy that reaches the Earths surface comes from the Sun. About 44 percent of solar radiation is in the visible light wavelengths,

but the Sun also emits infrared, ultraviolet, and other wavelengths. The Solar energy to the Earth refers to this energy that hits the surface of the Earth itself.

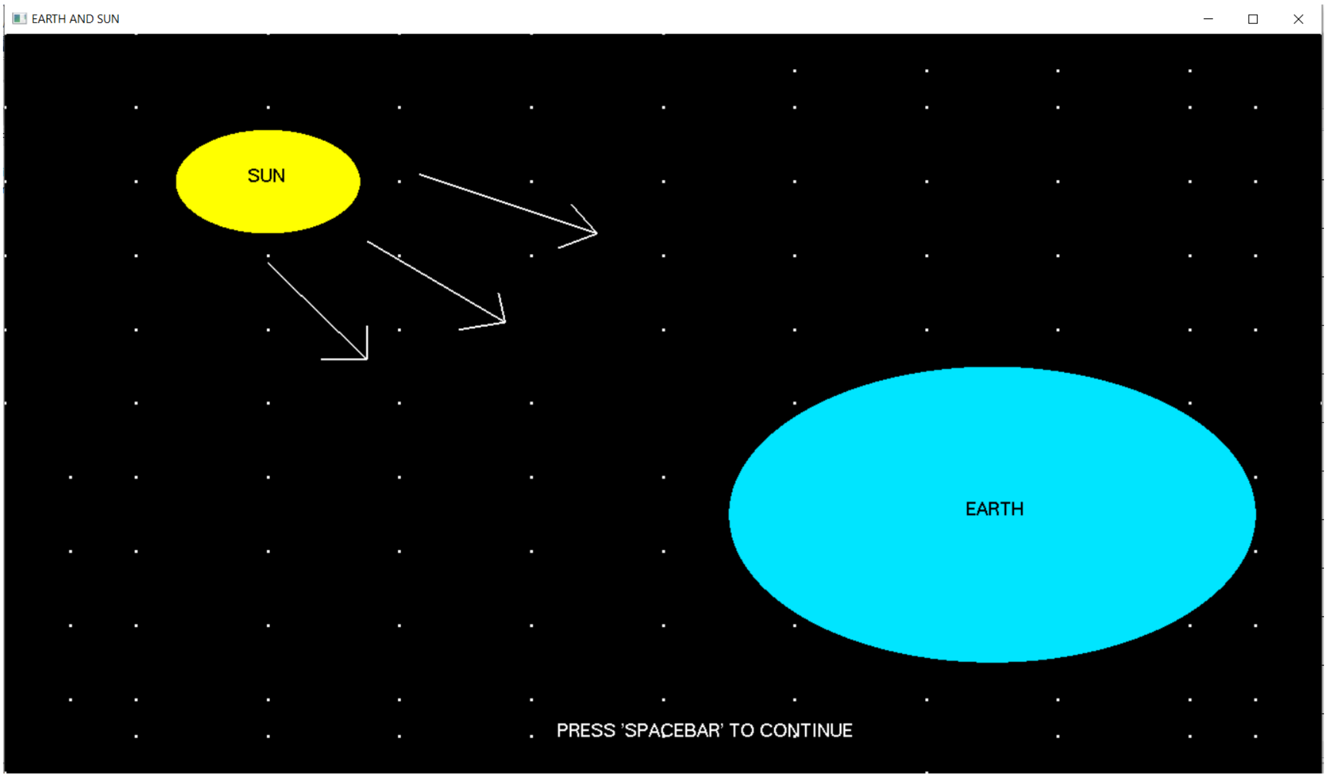
The amount of energy that reaches the the Earth provides a useful understanding of the energy for the Earth as a system. This energy goes towards weather, keeping the

temperature of the Earth at a suitable level for life, and powers the entire biosphere. Additionally, this solar energy can be used for solar power either with solar thermal

power plants or photovoltaic cells.

PRESS 'SPACEBAR' TO CONTINUE

Fig 5.2 Concept of Solar Power and Energy Page



PRESS 'SPACEBAR' TO CONTINUE

Fig 5.3 Sun and the Earth Screen

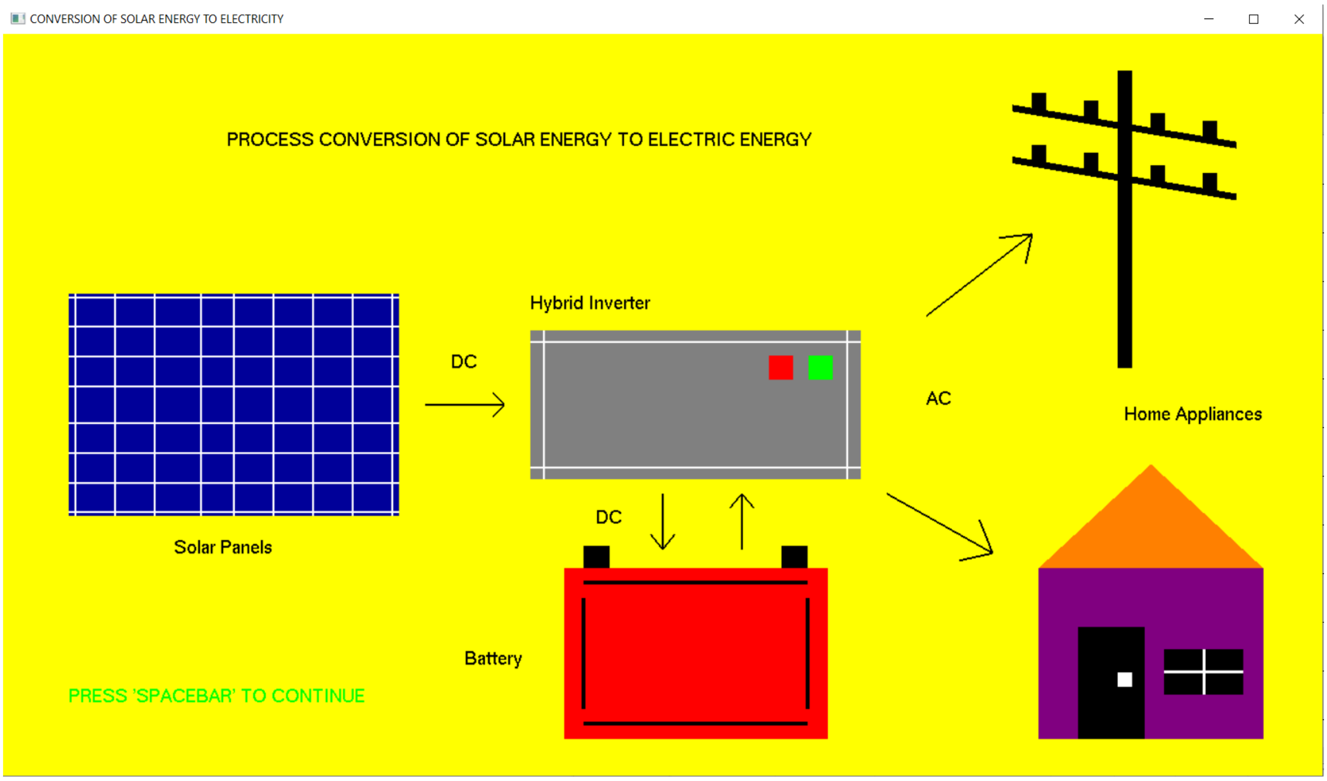

CONVERSION OF SOLAR ENERGY TO ELECTRICITY
PROCESS CONVERSION OF SOLAR ENERGY TO ELECTRIC ENERGY
Hybrid Inverter
DC
AC
Home Appliances
Solar Panels
DC
Battery
PRESS 'SPACEBAR' TO CONTINUE

Fig 5.4 Conversion of Solar Energy to Electrical Energy Screen

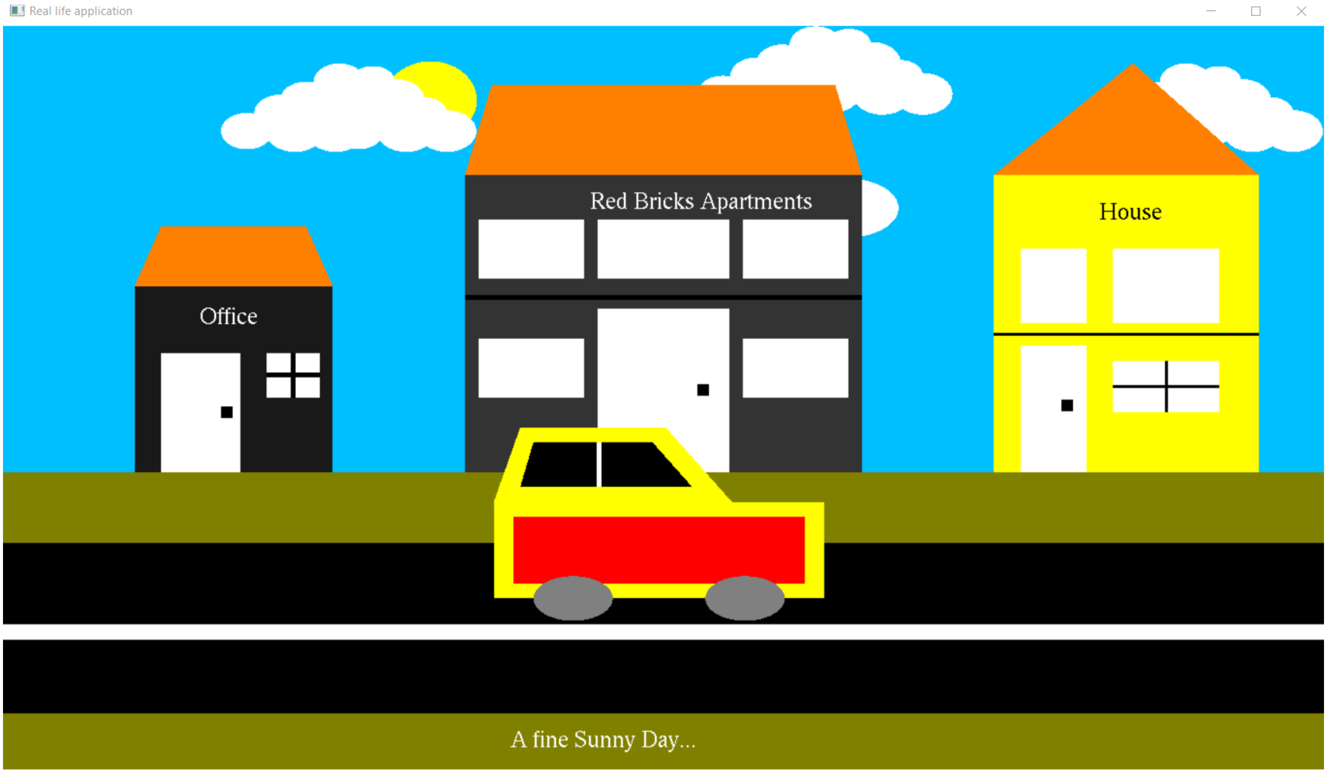
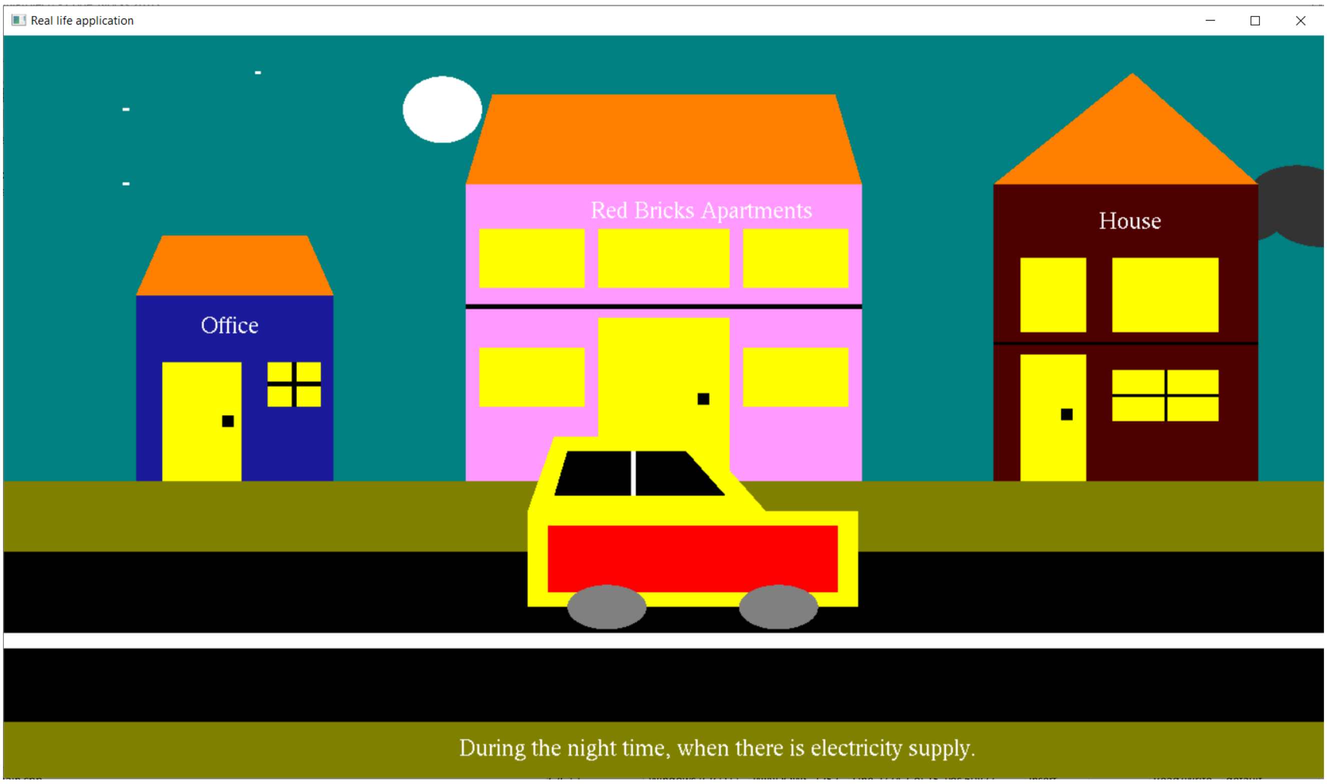
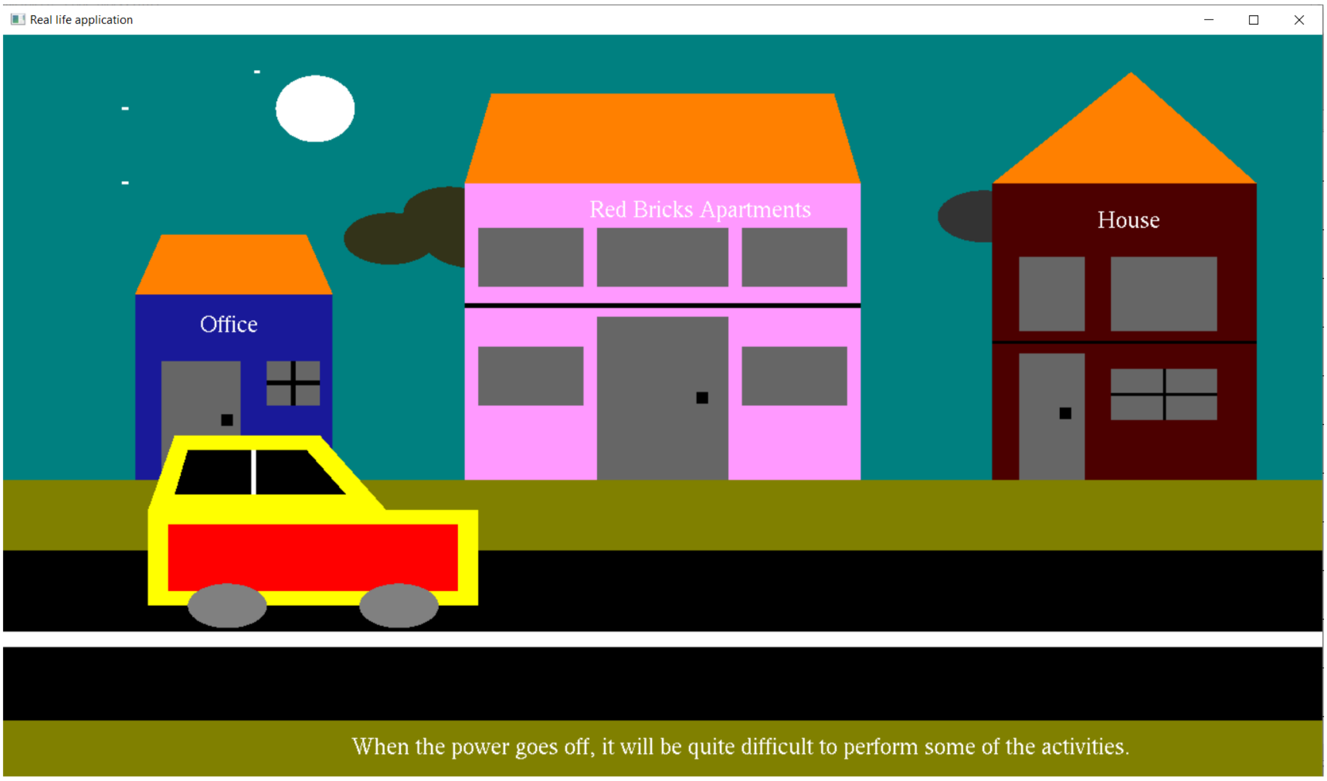


Fig 5.5 Frame 1 – During day



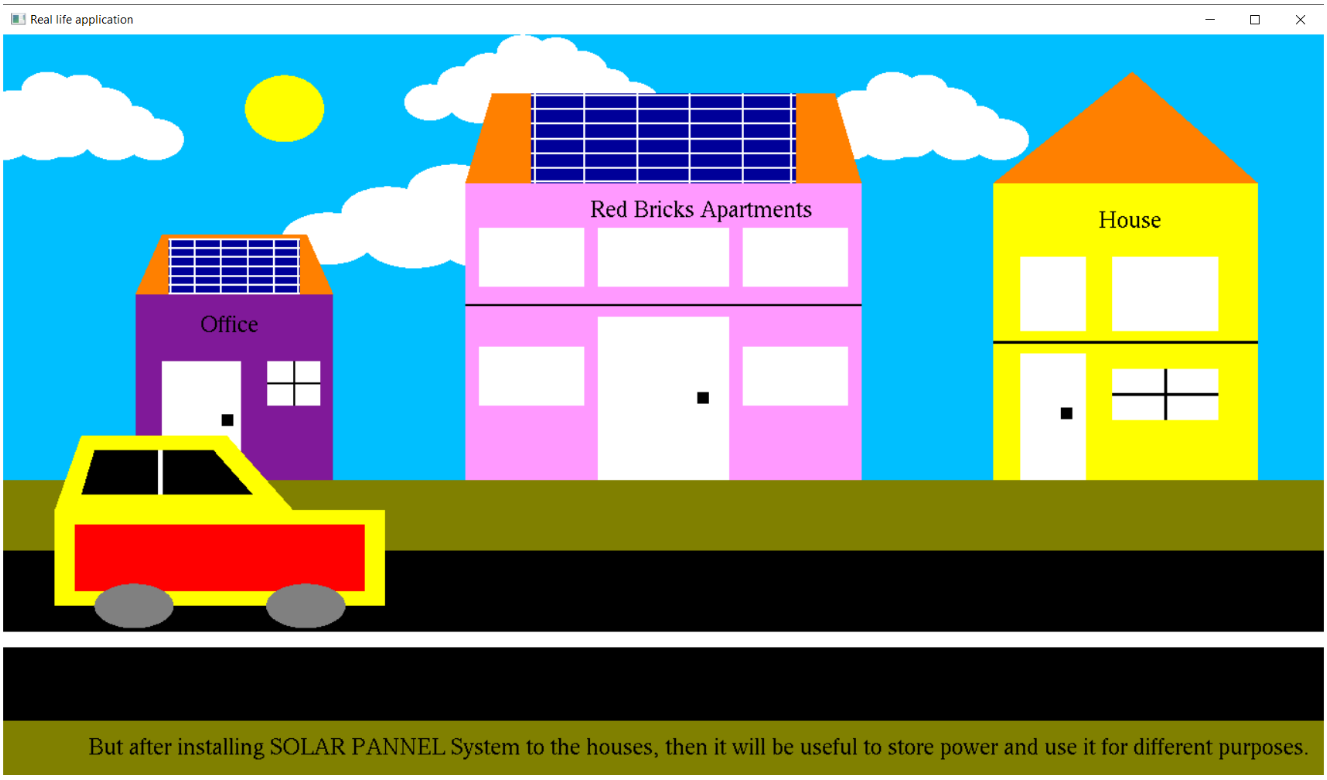
During the night time, when there is electricity supply.

Fig 5.6 Frame 2 – During night with electricity



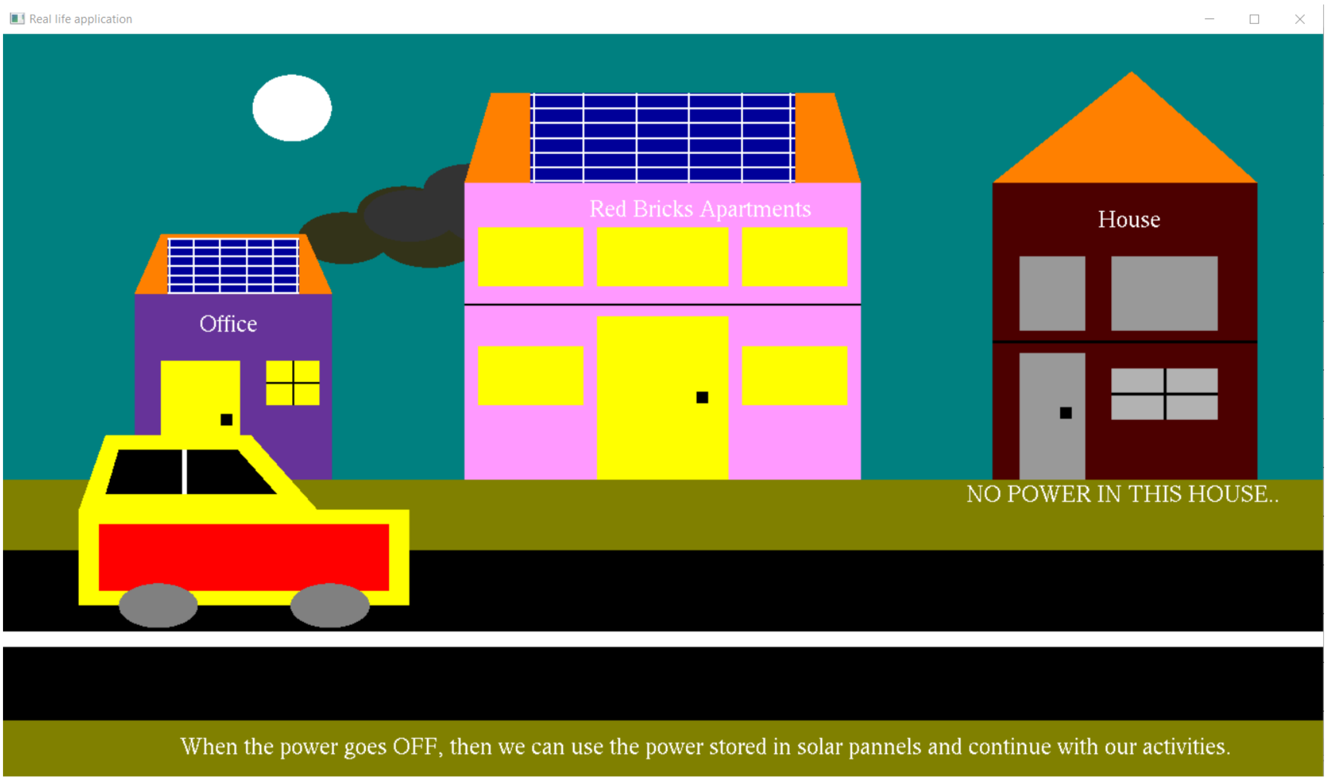
When the power goes off, it will be quite difficult to perform some of the activities.

Fig 5.7 Frame 3 – During night when no power



But after installing SOLAR PANNEL System to the houses, then it will be useful to store power and use it for different purposes.

Fig 5.8 Frame 4 – During day with solar panels



When the power goes OFF, then we can use the power stored in solar pannels and continue with our activities.

Fig 5.9 Frame 5 – During night with electricity stored in solar panels

CONCLUSION

Our project **“IMPLEMENTING THE CONCEPT OF SOLAR POWER AND ENERGY”** has helped us in understanding about computer graphics using OpenGL basic functions which can be used to manipulate the data and provide some animations and various concepts and methodologies used in computer graphics.

Sunlight, or solar energy, can be used directly for heating and lighting homes and businesses, for generating electricity, and for hot water heating, solar cooling, and a variety of other commercial and industrial uses etc. The expected outcome of this project is to make people understand how it works and therefore they can reduce the electricity consumption costs.

Developing this project helped us to learn about the computer graphics practically and to add realistic animations required for the project. The project is user friendly and has features which can be easily understood by the user. It demonstrates the OpenGL applications in 2D with animation.

REFERENCES

1. Donald Hearn & Pauline Baker: Computer Graphics with OpenGL Version, 3rd / 4th Edition, Pearson Education, 2011.
2. Edward Angel: Interactive Computer Graphics – A Top Down approach with OpenGL, 5th edition, Pearson Education, 2008.
3. Drawing and animating car using OpenGL libraries : https://youtu.be/FifbkLT4tb4 .
4. Drawing a circle using OpenGL libraries : https://youtu.be/Wq_VmYKqMzg , https://youtu.be/alL6FAH8pL4 .