# VISVESVARAYA TECHNOLOGICAL UNIVERSITY BELGAUM-590014



A Computer Graphics and Visualization Mini-Project Report

On

“Walking Robot”

A Mini-project report submitted in partial fulfilment of the requirements for the award of the degree of Bachelor of Engineering in Computer Science and Engineering of Visvesvaraya Technological University, Belgaum.

Submitted by:

Sai Nagesh C H (1DT19CS138)

AND

Sandeep (1DT19CS141)

Under the Guidance of:

Mrs. Apoorva Busad

Asst. Prof. Dept. of CSE

Mrs. Mangala H S

Asst. Prof. Dept. of CSE



Department of Computer Science and Engineering

DAYANANDA SAGAR ACADEMY OF TECHNOLOGY AND MANAGEMENT

Kanakapura Road, Udayapura, Bangalore 2021-2022



DAYANANDA SAGAR ACADEMY OF TECHNOLOGY AND MANAGEMENT

Kanakapura Road, Udayapura, Bangalore DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

(Accredited by NBA, NAAC A+, New Delhi)

CERTIFICATE

This is to certify that the Mini-Project on Computer Graphics (CG) entitled “Walking Robot” has been successfully carried out by Sai Nagesh C H (1DT19CS138) and Sandeep (1DT19CS141) a bonafide students of Dayananda sagar academy of technology and management in partial fulfillment of the requirements for the award of degree in Bachelor of Engineering in Computer Science and Engineering of Visvesvaraya Technological University, Belgaum during academic year 2019-23. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in the departmental library. The mini project report has been approved as it satisfies the academic requirements in respect of project work for the said degree

Mrs.Apoorva Busad Mrs Mangala H S

Asst. Prof. Dept of CSE Asst. Prof. Dept of CSE

#### Examiners: 1.

**2.**

#### Dr. C. NANDINI

Vice Principal & HOD, Dept. of CSE

**Signature with date**

# ABSTRACT

Walking robot is a robot movement simulator that potreys movement in multiple axis.The application features simulation of various movements like walk,run in multiple axis.The user can select between walk and run in order to make translation movement by using the mouse .Various other options such as "reset” button that resets the robot from its translated position to initial position and “exit” button which closes the application is also available.User can even adjust the viewing axis by making use of the arrow keys according to the users convenience.The walking robot in a whole is a good option for any user who aims at stress relief and to skip some time.

While presenting this Graphics Project on “Walking Robot”, I feel that it is our duty to acknowledge the help rendered to us by various persons.

Firstly I thank God for showering his blessings on me. I am grateful to my institution Dayananda Sagar Academy of Technology for providing me a congenial atmosphere to carry out the project successfully.

I would like to express my heartfelt gratitude to Dr. M. Ravishankar, Principal, Bangalore, for extending his support.

I would also like to express my heartfelt gratitude to Dr.C Nandini, Prof. Vice Principal & HOD, Computer Science and Engineering Department, whose guidance and support was truly invaluable.

I am very grateful to my guide, Mrs. Apoorva Busad, Asst. Prof. Department of Computer Science and Mrs. Mangala H S, Asst. Prof. Department of Computer Science, for her valuable guidance and advice at every stage of my project which helped me in the successful completion of my project.

I would also indebted to my Parents and Friends for their continued moral and material support throughout the course of project and helping me in finalize the presentation.

My heartful thanks to all those have contributed bits, bytes and words to accomplish this Project.

Thanking you all,

Sai Nagesh C H (1DT19CS138)

Sandeep (1DT19CS141)

TABLE OF CONTENT

CHAPTER NO. CHAPTER NAME PAGE NO.

1. INTRODUCTION ………………… 1

1.1 About computer graphics…………….. 1

1.2 About Open GL………………………. 2

1.3 About the Project……………………… 3

1. REQUIREMENT SPECIFICATION 4

2.1 Software Requirement………………… 4

2.2 Hardware Requirement……………….. 4

1. DESIGN 5
2. IMPLEMENTATION……………... 6

4.1 Psuedo Code………………………….. 6

4.2 Flow chart…………………………….. 7

4.3 Functions used………………………... 8

4.3.1 Header define………………………. 8

4.3.2 User defined function used………… 9

4.3.3 Standard library functions…………. 9

4.4 Source code………………………….. 11

1. SNAPSHOTS……………………… 13

5.1 Initial Postion………………………… 13

5.2 Menu Function……………………….. 13

5.3 Robot walking………………………... 14

5.4 Robot running………………………... 14

1. CONCLUTION AND FUTURE

ENHANCEMENT………………… 15

6.1 Conclusion…………………………… 15

6.2 Future Enhancement………………… 15

1. BIBILOGRAPHY………………… 16

7.1 Books………………………………… 16

7.2 Websites……………………………... 16

TABLE OF FIGURES

FIGURE NO. FIGURE NAME PAGE NO.

* 1. Graphic System 1

* 1. OpenGL Architecutre 3

3.1 Design of project 5

4.2 Control flow of project 8

5.1 Initial position of Robot 13

5.2 Menu function 13

5.3 Walking Robot 14

5.4 Robot exiting window 14

### 

### CHAPTER 1

* 1. **About Computer Graphics**

## INTRODUCTION

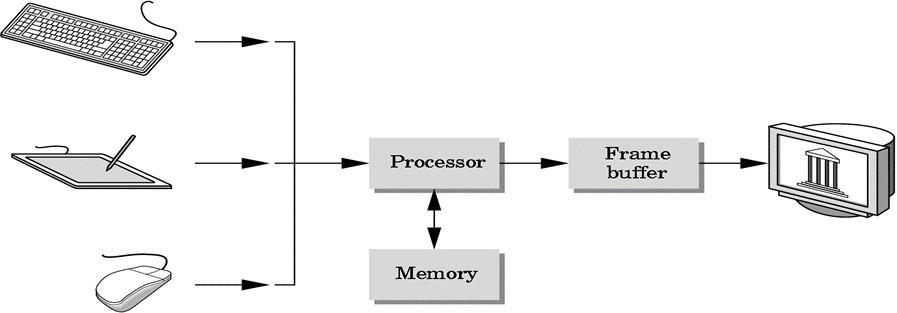
The Computer Graphics is one of the most effective and commonly used methods to communicate the processed information to the user. It displays the information in the form of graphics objects such as pictures, charts, graphs and diagram instead of simple text.

In computer graphics, pictures or graphics objects are presented as a collection of discrete picture elements called pixels. The pixel is the smallest addressable screen element

Computer graphics today is largely interactive: The user controls the contents structure, and appearance of objects and their displayed images by using input devices, such as a keyboard, mouse, or touch-sensitive panel on the screen.

Computer Graphics relies on an internal model of the scene, that is, mathematical representation suitable for graphical computations. The model describes the 3D shapes, layout and materials of the scene. This 3D representation then has to be projected to compute a 2D image from a given viewpoint, this is rendering step. Rendering involves projecting the objects, handling visibility (which parts of objects are hidden) and computing their appearance and lighting interactions. Finally, for animated sequence, the motion of objects has to be specified.

Computer graphics today is largely interactive: The user controls the contents structure, and appearance of objects and their displayed images by using input devices, such as a keyboard, mouse, or touch-sensitive panel on the screen.



**Fig. 1.1: Graphic System**

The image processing can be classified as

* + 1. Image enhancement.
    2. Pattern detection and recognition
    3. Scene analysis and computer vision.

The image enhancement deals with the improvement in the image quality by eliminating noise or by increasing image contrast. Pattern detection and recognition deals with the detection and clarification of standard patterns

And finding deviations from these patterns .The optical character recognition (OCR) technology is a practical example for pattern detection & recognition. Scene analysis deals with the recognition and reconstruction of 3D model of scene from several 2D images.

### About OpenGL

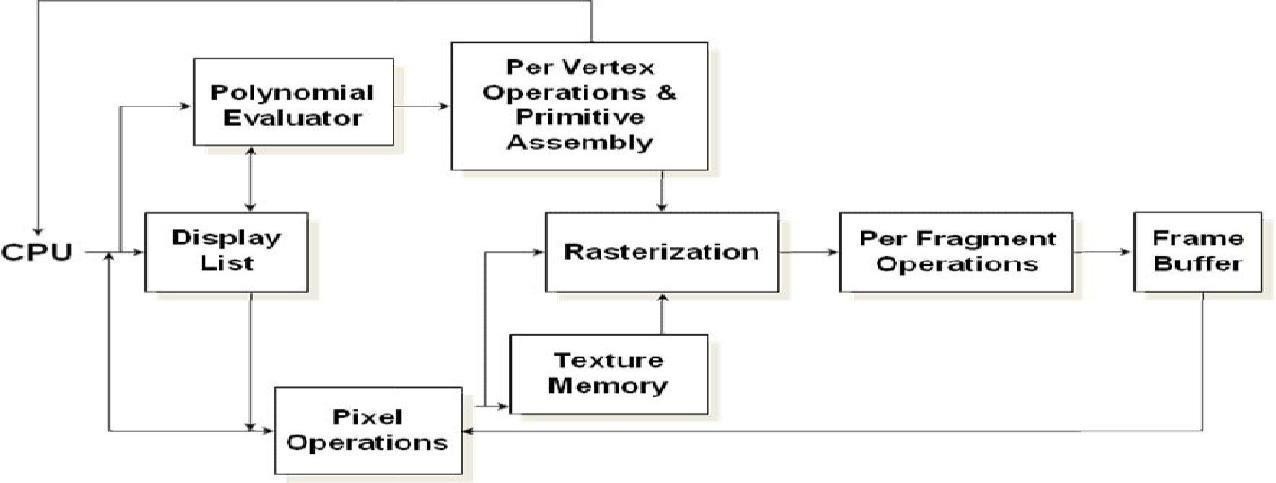
OpenGL (Open Graphics Library) is a standard specification defining a cross-language, cross-platform API for writing applications that produce 2D and 3D computer graphics. The interface consists of over 250 different function calls which can be used to draw complex three- dimensional scenes from simple primitives. OpenGL was developed by Silicon Graphics Inc.

OpenGL's basic operation is to accept primitives such as points, lines and polygons, and convert them into pixels. This is done by a graphics pipeline known as the OpenGL state machine.

#### Features of OpenGL:

* Geometric Primitives allow you to construct mathematical descriptions of objects.
* Viewing and Modeling permits arranging objects in a 3-dimensional scene, move our camera around space and select the desired vantage point for viewing the scene to be rendered.
* Materials lighting OpenGL provides commands to compute the color of any point given the properties of the material and the sources of light in the room.
* Transformations: rotation, scaling, translations, perspectives in 3D, etc.

### OpenGL Architecture



**Fig. 1.2:OpenGL Architecture**

This is the most important diagram you will see today, representing the flow of graphical information, as it is processed d from CPU to the frame buffer.

There are two pipelines of data flow. The upper pipeline is for geometric, vertex-based primitives. The lower pipeline is for pixel-based, image primitives. Texturing combines the two types of primitives together.

### About the Project

Computer graphics involves the designing of objects in different forms which are regular and irregular in shape. The mini project named “Transmission of messages” shows the way how the messages are transmitted from one place to another. This project is designed and implemented using OpenGL interactive application that basically deals with providing the graphical interface between the user and the system. Through the use of keyboard events, we can input to start the transmission of messages. A drop down menu has been created which provides the options of choosing the conversation.

**CHAPTER 2**

## REQUIREMENT SPECIFICATION

The requirement specification is a comprehensive description of the software and the hardware requirements required to run the project successfully.

### Software Requirements

Operating System: Windows 7/8/10/11 Languages : OpenGL

Compiler used : GNU GCC compiler IDE: Codeblocks

### Hardware Requirements

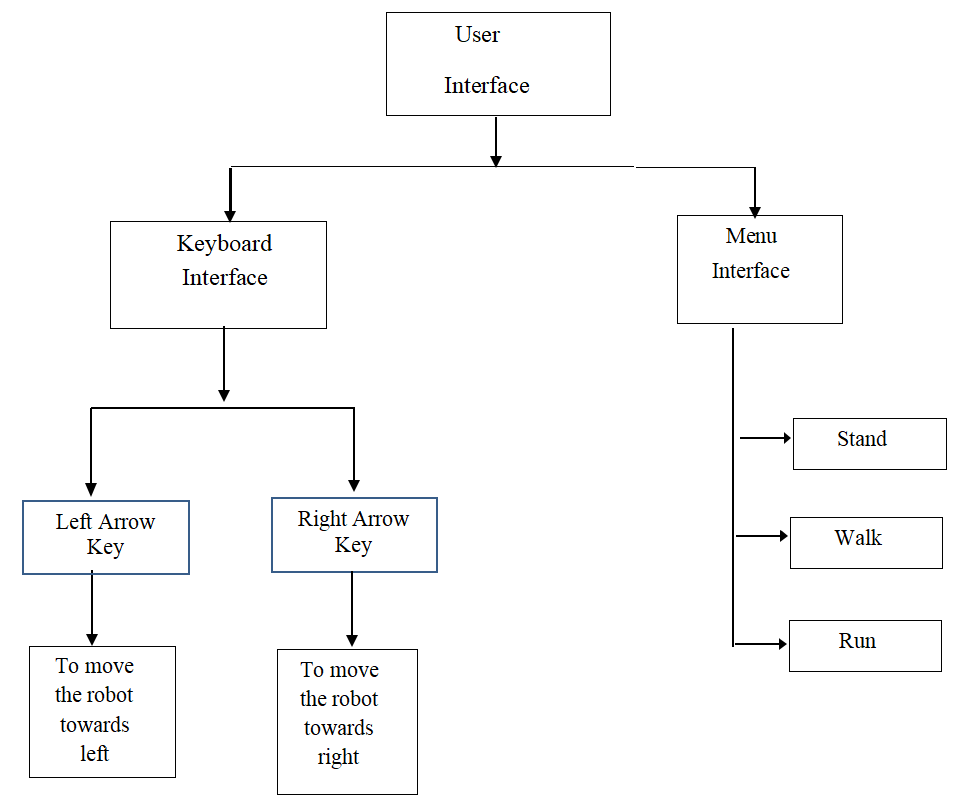
Processor : Pentium 3 and above RAM : 1GB 256Mhz

Harddisk : 2Mb

Keyboard: Standard 101 key Keyboard

## DESIGN

Design is the planning that lays the basics for the making of every objects or systems. This chapter involves designing of various aspects and different stages of project. When program is made to execute, the output window is displayed first. The flow of operation from output window is shown in figure



**Fig. 3.1:Design of the project**

**Fig.3.1 Design of the project**

## IMPLEMENTATION

Implementation is the stage where all planned activities are put into action. Before the implementation of a project, the implementer’s (spearheaded by the project committee or executive) should identify their strength and weaknesses (internal forces), opportunities and threats (external forces).

Implementation also includes a pseudo code.

### Pseudo code

Void init ()

{

This function is used to initialize the viewing of the output.

}

Void reshape()

{

The reshape function defines what to do when the window is resized.

}

Void arrowkeys ()

{

This function sets the keyboard input keys which can be used to interact with the program

}

Void display()

{

The color buffer is cleared using the glClear(). Text in the instruction page is set here. Light properties are set in here.

}

Void timer() {

This function is used to control the speed of movements.

}

Void stand() {

This function is used to stand the robot in correct position.

}

Void MenuFunc() {

This function is used to display the menu.

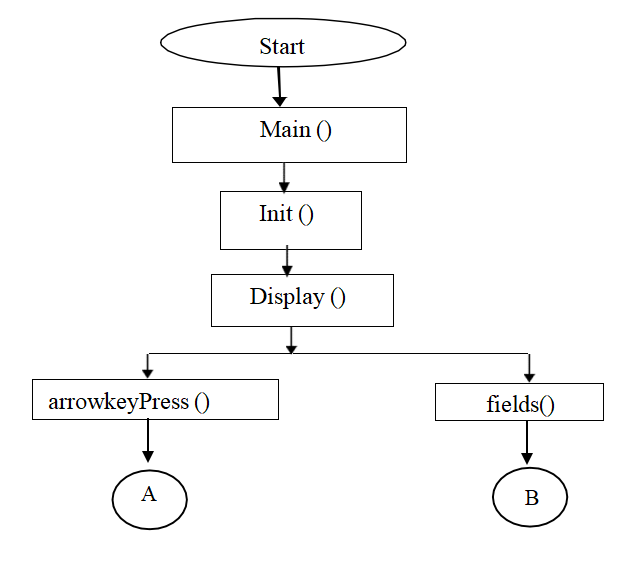
}

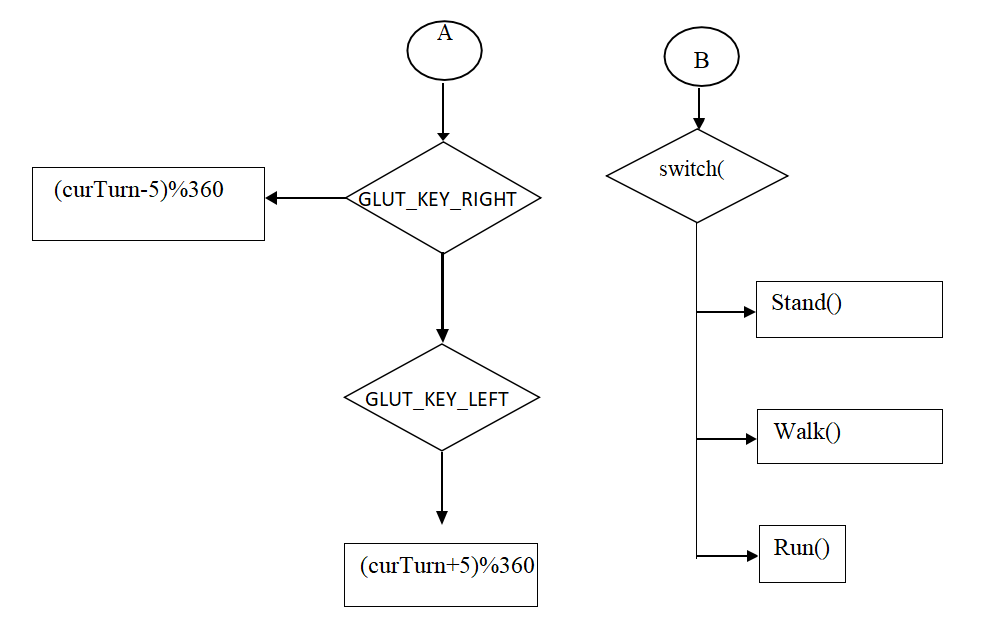
Int Main() {

This is the main function of the program.

}

**4.2 Flowchart**

****

****

**Fig. 4.2:Control flow of the project**

### 4.3 Functions used:

**4.3.1 Headers defined**

The inbuilt functions are defined in the OpenGL library. Some of the headers that are used are as follows.

**#include<stdio.h>:** To take input from standard input and write it to standard output.

**#include<GL/glut>:** To include glut library functions. **#include<math.h>:** To define common mathematical functions. **#include <string.h>:** To define string functions

### 4.3.2 User defined functions used in the project

* + - * **stand():** This function is used to stand the robot at particular place
      * **timer():** This function is used for movements of robot.
      * **reshape():** This function defines what to do when window is resized.
      * **arrowKeyPress():** This function is used to move the robot.
      * **MenuFunc():** This function is used to access the menu functions.

### 4.3.3 Standard library functions used in the project:

* **glutInit (int \*argc, char \*\*argv):** glutInit is used to initialize the GLUT library.
* **glutInitDisplayMode(unsigned int mode):**glutInitDisplayMode sets the initial display mode.
* **glutInitWindowPosition(intx,int y):** Specifies the initial position of the top-left corner of the window in pixels.
* **glutInitWindowSize(intwidth,int height):** Specifies the initial height and width of the window in pixels.
* **glutKeyboardFunc (void \*f(char key,int width,int height):**KeyboardFunc sets the keyboard callback for the current window.
* **glClear():**The clear function clears buffers to preset values.
* **glClearColor(GLclampf r, GLclampf g, GLclampf b, GLclampf a):**The glClearColor function specifies clear values for the color buffers.
* **glMatrixMode(GLenum mode):**This function specifies which matrix is the current matrix.
* **glLoadIdentity():** Set the current transformation matrix to an identity matrix
* **glPushMatrix(),glPopMatrix():** This pushes to and pops from the matrix stack corresponding to the current matrix mode.
* **glPointSize():**The glPointSize function specifies the diameter of rasterized points.
* **glTranslate[fd](TYPE x,TYPEy,TYPE z):** This function multiplies the current matrix by a translation matrix.
* **gluOrtho2D(GLdouble left, GLdouble right, GLdouble bottom, GLdouble top):** The gluOrtho2D function defines a 2-D orthographic projection matrix.
* **glBegin (glEnum mode):** It initiates a new primitive of type mode and starts the collection of vertices. Values of mode include GL\_POINTS, GL\_LINES, GL\_LINE\_STRIP, and GL\_POLYGON.
* **glEnd():** It terminates a list of vertices.
* **glFlush ():** It forces any buffered openGL commands to execute.
* **glutMainLoop() :** Itcauses the program to enter an event processing loop.
* **glutDisplayFunc(void (\*func)(void)):** It registers the display function ‘func’ that is executed when the window needs to be redrawn.
* **glutCreateMenu(void (\*f)(void)):** It returns an identifier for a top-level menu and registers the callback function f that returns an interger value corresponding to the menu entry selected.

### Source Code

#include <stdio.h>

#include <string.h>

#include <stdlib.h>

#include <math.h>

#define GLUT\_DISABLE\_ATEXIT\_HACK

#include <gl/glut.h>

//The width and height of the current window

int curw;

int curh;

int swingLeft = 0;

int swingRight = 0;

bool isSwingForward = false;

bool isLighting = false;

int curTurn=180 ;//current facing angle

int maxAngel = 0;//maximum swing angle

int stepDis = 0;//thigh swing speed

float speed = 0;//forward speed

bool isStand = true;

float curDistanceX=0 , curDistanceZ=0;//current distance

float legDis = 0 ;

void display(void) {

glClear (GL\_COLOR\_BUFFER\_BIT); glShadeModel(GL\_SMOOTH);

//painted background

glBegin(GL\_QUADS);

glColor3ub(255,255,255);

glVertex3f(-8.0f, -3.0f, 4.0f);

glVertex3f(8.0f, -3.0f, 4.0f);

glColor3ub(33,33,33);

glVertex3f(8.0f, -3.0f, -4.0f);

glVertex3f(-8.0f, -3.0f, -4.0f);

glEnd();

glVertex3f(8.0f, 8.0f, -4.0f);

glVertex3f(-8.0f, 8.0f, -4.0f); glEnd();

//Push the stack, draw the lower right half of the body

glPushMatrix();

glTranslatef (curDistanceX,0.0,curDistanceZ); //go ahead

glRotatef ((GLfloat) curTurn, 0.0, 1.0, 0.0);//turn

//Adjust the coordinate system to the right to locate the center of rotation

glTranslatef (0.375,0.0, 0.0);

//whether to lift the right thigh

glRotatef ((GLfloat) swingRight, 1.0, 0.0, 0.0);

//Lower the coordinate system and prepare to draw the right thigh

glTranslatef (0.0, -0.5, 0.0);

glColor3f(0.5,0.8,0.3);

//draw right thigh glPushMatrix();

glScalef (0.4, 1.0, 0.4);

glutSolidCube(1.0);

glPopMatrix();

//Lower the coordinate system and locate the center of rotation

glTranslatef (0.0, -0.5, 0.0);

//Whether to lift the right calf

glRotatef ((GLfloat) legDis, 1.0, 0.0, 0.0);

//Lower the coordinate system and prepare to draw the right calf

glTranslatef (0.0, -0.5, 0.0);

glColor3f(0.1,0.5,0.5);

//draw right calf

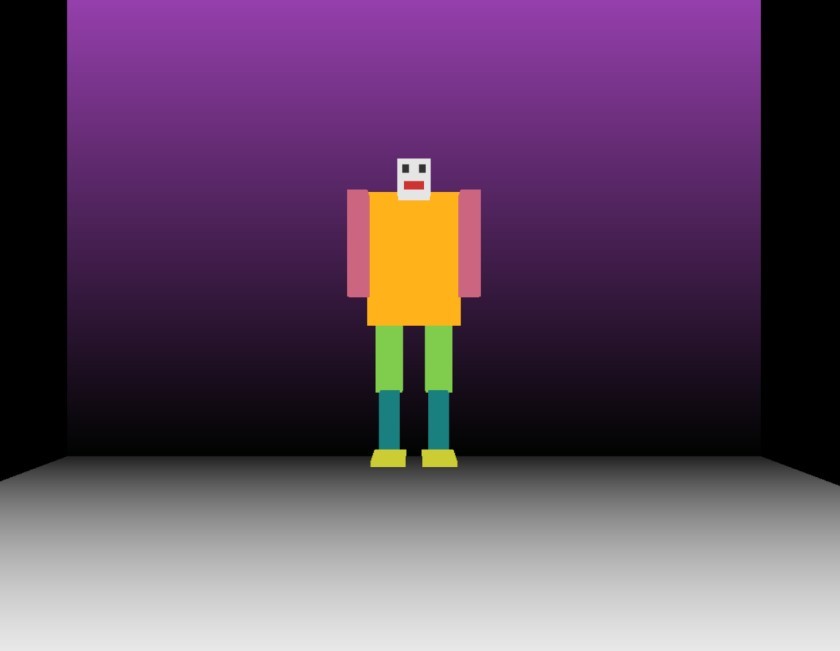
glPushMatrix();

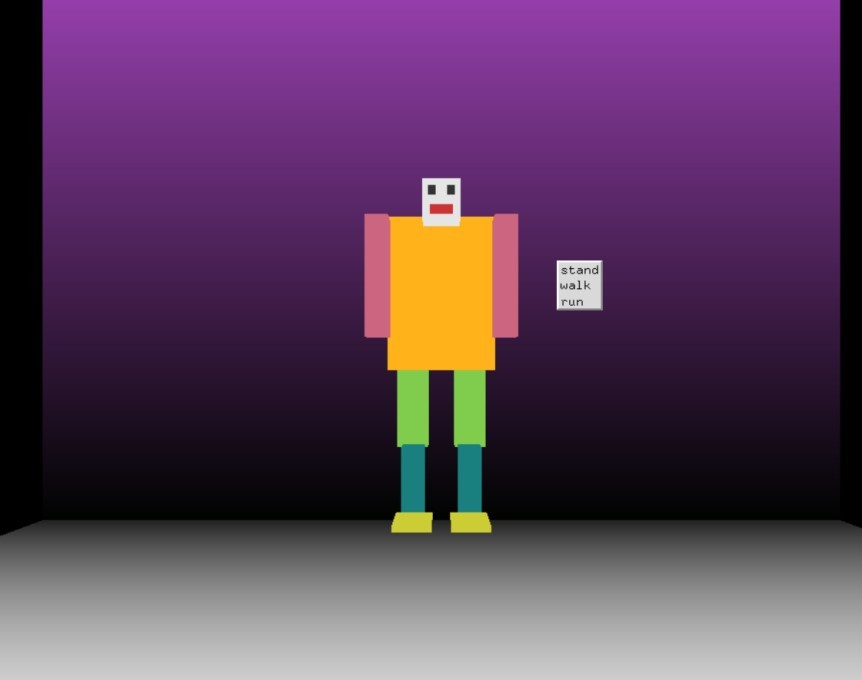
glScalef (0.3, 1.0, 0.3);

**CHAPTER 5**

## SNAPSHOTS

A snapshot is the state of the system at a particular point in time. It can refer to the actual copy of a state of a system or to a capability provide by systems.

**Fig. 5.1: Initial position of the Robot**



**Fig. 5.2.Menu Function**



**Fig. 5.3. Walking Robot**



**Fig. 5.4. Robot exiting the window**

**CHAPTER 6**

## CONCLUSION

The mini project named “Walking Robot” creates a 3D Robot which has working functions. Some of the functions are walking, standing, running in different directions from the keyboard input. This project is designed and implemented using OpenGL interactive application that basically deals with providing the graphical interface between the user and the system. Using the drop down menu various functions can applied on the robot.

### Future Enhancements

The project has graphical usage where in different functions which can be applicable in graphical packages are used. The applications developed so far is in its basic working stage. There can be more enhancements like

* Enhancing the quality of the animation.
* Adding the capabilities of camera view.
* By providing real time info about various stages of transmission.
* We can add other features like showing signal traffic

**BOOKS**

## BIBILOGRAPHY

1. Edward Angel: Interactive Computer Graphics A Top-Down Approach with OpenGL, 5th Edition, Pearson Education, 2008.
2. Donald Hearn and Pauline Baker: Computer Graphics- OpenGL Version, 3rd Edition, Pearson Education, 2008.
3. F.S Hill Jr.: Computer Graphics Using OpenGL, 3rd Edition, PHI, 2009.

### WEBSITES

1. <http://www.opengl.org/resources/libraries/glut/>
2. <http://www.opengl.org/>
3. <http://en.wikipedia.org/wiki/opengl>