/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Beginning of the header file "bike.h"

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include<GL/glut.h>

#include<stdio.h>

#include<string.h>

#include<stdlib.h>

#include<math.h>

#define PI 3.14159

#define WIN\_WIDTH 1920

#define WIN\_HEIGHT 1080

#define BIKE\_LENGTH 3.3f

#define ROD\_RADIUS 0.05f

#define GAS\_TANK 0.3f

#define NUM\_SPOKES 5

#define SPOKE\_ANGLE 72

#define RADIUS\_WHEEL 1.0f

#define TUBE\_WIDTH 0.20f

#define RIGHT\_ROD 2.35f

#define RIGHT\_ANGLE 33.0f

#define MIDDLE\_ROD 1.8f

#define MIDDLE\_ANGLE 125.0f

#define BACK\_CONNECTOR 0.5f

#define LEFT\_ANGLE 50.0f

#define WHEEL\_OFFSET 0.11f

#define WHEEL\_LEN 0.8f

#define TOP\_LEN 1.0f

#define FRONT\_ROD 2.5f

#define CRANK\_ROD 1.9f

#define CRANK\_RODS 2.5f

#define CRANK\_ANGLE 0.0f

#define HANDLE\_ROD 1.2f

#define FRONT\_INCLINE 38.0f

#define HANDLE\_LIMIT 30.0f

#define INC\_STEERING 2.0f

#define INC\_SPEED 0.05f

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* All the Global Variables are Here

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Bike - related variables

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

GLfloat pedalAngle, speed, steering;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* User view related variables

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

GLfloat camx,camy,camz;

GLfloat anglex,angley,anglez;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Mouse related variables

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int prevx,prevy;

GLenum Mouse;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Bike position related

\* variables

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

GLfloat xpos,zpos,direction;

void welcome\_window();

void operations\_window();

void ZCylinder(GLfloat radius,GLfloat length);

void XCylinder(GLfloat radius,GLfloat length);

void drawFrame(void);

void gear( GLfloat inner\_radius, GLfloat outer\_radius,

GLfloat width,GLint teeth, GLfloat tooth\_depth );

void drawChain(void);

void drawPedals(void);

void drawTyre(void);

void drawSeat(void);

void init(void);

void reset(void);

void display\_bike(void);

void idle(void);

void updateScene(void);

void landmarks(void);

void special(int key,int x,int y);

void keyboard(unsigned char key,int x,int y);

void mouse(int button,int state,int x,int y);

void motion(int x,int y);

GLfloat Abs(GLfloat);

GLfloat degrees(GLfloat);

GLfloat radians(GLfloat);

GLfloat angleSum(GLfloat, GLfloat);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* End of the header file

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Start of the source file "bike.c"

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//#include "bike.h"

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* A

\* 1 ========== 2

\* /\ | B

\* / \ / 5

\* E / \D /

\* / \ / C

\* / \ /

\* 3 ==========/ 4

\* F

\* 1 = 212,82 \* 2 = 368,82

\* 5 = 369,94

\* 3 = 112,220

\* 4 = 249,232

\*

\* 214 = 73 //angle between 2,1 & 4

\* 124 = 55

\* 142 = 52

\* 143 = 73

\* 134 = 50

\* 431 = 57

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void ZCylinder(GLfloat radius,GLfloat length)

{

GLUquadricObj \*cylinder;

cylinder=gluNewQuadric();

glPushMatrix();

glTranslatef(0.0f,0.0f,0.0f);

gluCylinder(cylinder,radius,radius,length,15,5);

glPopMatrix();

}

void XCylinder(GLfloat radius,GLfloat length)

{

glPushMatrix();

glRotatef(90.0f,0.0f,1.0f,0.0f);

ZCylinder(radius,length);

glPopMatrix();

}

// called by idle()

void updateScene()

{

GLfloat xDelta, zDelta; //Distance

GLfloat rotation;

GLfloat sin\_steering, cos\_steering;

// if the tricycle is not moving then do nothing

if (-INC\_SPEED < speed && speed < INC\_SPEED) return; //"speed<INC\_SPEED" because if again accelerated, the bike stops moving if we put speed>INC\_SPEED since speed value crosses INC\_SPEED

if(speed < 0.0f)

pedalAngle = speed = 0.0f;

// otherwise, calculate the new position of the tricycle

// and the amount that each wheel has rotated.

// The tricycle has moved "speed\*(time elapsed)".

// We assume that "(time elapsed)=1".

xDelta = speed\*cos(radians(direction + steering)); //To find out the new position of the bike once it starts moving

zDelta = speed\*sin(radians(direction + steering));

xpos += xDelta; //Initialise this calculated new position to the variable xpos & zpos

zpos -= zDelta;

pedalAngle = degrees(angleSum(radians(pedalAngle), speed/RADIUS\_WHEEL)); //Pedal rotates as bike moves but this effect has been neutralised in the program // we'll be using sin(steering) and cos(steering) more than once

// so calculate the values one time for efficiency

sin\_steering = sin(radians(steering)); //Angle calculated for rotation of our bike towards the direction of handle rotation

cos\_steering = cos(radians(steering));

// see the assignment 3 "Hint"

rotation = atan2(speed \* sin\_steering, BIKE\_LENGTH + speed \* cos\_steering); //atan2(); used mostly to convert from rectangular (x,y) to polar (r,0)

direction = degrees(angleSum(radians(direction),rotation));

}

// angleSum(a,b) = (a+b) MOD 2\*PI

// a and b are two angles (radians)

// both between 0 and 2\*PI

GLfloat angleSum(GLfloat a, GLfloat b)

{

a += b;

if (a < 0) return a+2\*PI; //If returned angle is towards left and lesser than its threshold value, increment its value to zero & do nothing

else if (a > 2\*PI) return a-2\*PI; //If returned angle is towards right and greater than its threshold value, decrement its value to zero & do nothing

else return a; //If nothing, return the radians equivalent of angle 'a'

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Draw the metal frame of the bike and also

\* draw the seat and the back wheel with

\* this.

\* All these parts are always together in the

\* same plane.They never move out ot the

\* PLANE!

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void drawFrame()

{

glColor3f(1.0f,0.0f,0.0f);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* First draw the all the items

\* at the center of the frame.

\* Draw the bigger gear,the small

\* cylinder acting as the socket

\* for the foot rest.Also DON'T

\* forget to draw the two

\* connecting cemtral rods

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

glPushMatrix();

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Allow me to draw the BIGGER

\* gear and the socket cylinder

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

glPushMatrix();

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Let the gear have the

\* green color

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

glColor3f(0.0f,1.0f,0.0f);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* The gear should be

\* outside the frame !!!

\* This is the bigger

\* GEAR

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

glPushMatrix();

glTranslatef(0.0f,0.0f,0.06f);

glRotatef(-2\*pedalAngle,0.0f,0.0f,1.0f);

gear(0.08f,0.3f,0.03f,30,0.03f);

glPopMatrix();

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Restore the color of the

\* frame

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

glColor3f(1.0f,0.0f,0.0f);

glTranslatef(0.0f,0.0f,-0.2f);

ZCylinder(0.08f,0.32f);

glPopMatrix();

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Lets first draw the

\* rightmost rod of the frame

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

glRotatef(RIGHT\_ANGLE,0.0f,0.0f,1.0f);

XCylinder(ROD\_RADIUS,RIGHT\_ROD);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Now draw the centre rod of

\* the frame which also supports

\* the seat

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

glRotatef(MIDDLE\_ANGLE-RIGHT\_ANGLE,0.0f,0.0f,1.0f);

XCylinder(ROD\_RADIUS,MIDDLE\_ROD);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* We have drawn the support.So

\* let's draw the seat with a

\* new color

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

glColor3f(1.0f,1.0f,1.0f);

glTranslatef(MIDDLE\_ROD,0.0f,0.0f);

glRotatef(-MIDDLE\_ANGLE,0.0f,0.0f,1.0f);

glScalef(0.3f,ROD\_RADIUS,0.25f);

drawSeat();

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Restore the color !

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

glColor3f(1.0f,0.0f,0.0f);

glPopMatrix();

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Draw the horizontal part of

\* the frame.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Draw the main single rod

\* connecting the center of the

\* frame to the back wheel of the

\* bike

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

glPushMatrix();

glRotatef(-180.0f,0.0f,1.0f,0.0f);

XCylinder(ROD\_RADIUS,BACK\_CONNECTOR);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Draw the two rods on the either

\* side of the front wheel

\* These rods are part of the

\* horizontal part of the bike

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

glPushMatrix();

glTranslatef(0.5f,0.0f,WHEEL\_OFFSET);

XCylinder(ROD\_RADIUS,RADIUS\_WHEEL+TUBE\_WIDTH);

glPopMatrix();

glPushMatrix();

glTranslatef(0.5f,0.0f,-WHEEL\_OFFSET);

XCylinder(ROD\_RADIUS,RADIUS\_WHEEL+TUBE\_WIDTH);

glPopMatrix();

glPopMatrix();

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Draw the leftmost rods(back part) of the

\* frame of the bike

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

glPushMatrix();

glTranslatef(-(BACK\_CONNECTOR+RADIUS\_WHEEL+TUBE\_WIDTH),0.0f,0.0f);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Transalted to the back wheel

\* position.Why not draw the back

\* wheel and also the gear ?

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

glPushMatrix();

glRotatef(-2\*pedalAngle,0.0f,0.0f,1.0f);

drawTyre();

glColor3f(0.0f,1.0f,0.0f);

gear(0.03f,0.15f,0.03f,20,0.03f);

glColor3f(1.0f,0.0f,0.0f);

glPopMatrix();

glRotatef(LEFT\_ANGLE,0.0f,0.0f,1.0f);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Draw the two rods on the either

\* side of the wheel connecting the

\* backwheel and topmost horizontal

\* part of the wheel

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

glPushMatrix();

glTranslatef(0.0f,0.0f,-WHEEL\_OFFSET);

XCylinder(ROD\_RADIUS,WHEEL\_LEN);

glPopMatrix();

glPushMatrix();

glTranslatef(0.0f,0.0f,WHEEL\_OFFSET); XCylinder(ROD\_RADIUS,WHEEL\_LEN);

glPopMatrix();

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Draw the single rod of the

\* same setup

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

glTranslatef(WHEEL\_LEN,0.0f,0.0f);

XCylinder(ROD\_RADIUS,CRANK\_ROD-1.0); //It is the rod which connects the handle bar and the connector rods of the front wheel

//Draw Main Supporting Rod

glTranslatef(CRANK\_ROD,0.0f,0.0f);

glRotatef(-LEFT\_ANGLE,0.0f,0.0f,1.0f);

XCylinder(0.0000000001,TOP\_LEN); //Its thickness has been reduced since it acts as a supporting rod, hence it's not necessary to display it

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Now instead of again traversing

\* all the way back and again

\* forward.WHY NOT DRAW THE

\* HANDLE FROM HERE ITSELF?

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//Draw The Gas Tank

glPushMatrix();

glColor3f(0.5f,0.5f,0.5f);

glTranslatef(-0.6,-0.6f,0.0f);

XCylinder(GAS\_TANK,1.0f);

glPopMatrix();

//Draw the head light

glPushMatrix();

glColor3f(0.5f,0.5f,0.5f);

glTranslatef(0.6,0.5f,0.0f);

XCylinder(GAS\_TANK,0.5f);

glPopMatrix();

//Draw The Exhaust

glPushMatrix();

glColor3f(0.5f,0.5f,0.5f);

glTranslatef(-3.2,-2.1f,0.3f);

XCylinder(ROD\_RADIUS+0.1f,1.5f); //Right lower

glPopMatrix();

glPushMatrix();

glColor3f(0.5f,0.5f,0.5f);

glTranslatef(-3.2,-2.1f,0.3f);

glRotatef(120.0,0.0f,0.0f,1.0f);

XCylinder(ROD\_RADIUS+0.1f,0.5f); //Right upper

glPopMatrix();

glPushMatrix();

glColor3f(0.5f,0.5f,0.5f);

glTranslatef(-3.2,-2.1f,-0.3f);

XCylinder(ROD\_RADIUS+0.1f,1.5f); //Left lower

glPopMatrix();

glPushMatrix();

glColor3f(0.5f,0.5f,0.5f);

glTranslatef(-3.2,-2.1f,-0.3f);

glRotatef(120.0,0.0f,0.0f,1.0f);

XCylinder(ROD\_RADIUS+0.1f,0.5f); //Left upper

glPopMatrix();

//Draw The Right Doom

glPushMatrix();

glColor3f(0.5f,0.5f,0.5f);

glTranslatef(0.4,-0.8f,0.2f);

glRotatef(LEFT\_ANGLE,0.0f,0.0f,1.0f);

XCylinder(GAS\_TANK,1.0f); //Right upper

glTranslatef(0.2,-0.8f,0.2f);

glRotatef(LEFT\_ANGLE-50.0,0.0f,0.0f,1.0f);

XCylinder(GAS\_TANK,1.0f); //Right lower

glPopMatrix();

//Draw The Left Doom

glPushMatrix();

glColor3f(0.5f,0.5f,0.5f);

glTranslatef(0.4,-0.8f,-0.2f);

glRotatef(LEFT\_ANGLE,0.0f,0.0f,1.0f);

XCylinder(GAS\_TANK,1.0f); //Left upper

glTranslatef(0.2,-0.8f,-0.2f);

glRotatef(LEFT\_ANGLE-50.0,0.0f,0.0f,1.0f);

XCylinder(GAS\_TANK,1.0f); //Left lower

glPopMatrix();

//Draw The Engine

glPushMatrix();

glColor3f(0.1,0.1,0.1);

glTranslatef(-0.4f,-1.2f,0.0f);

XCylinder(GAS\_TANK,1.3f);

glPopMatrix();

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Now draw the handle and

\* small support rod which

\* is incorporated in the

\* frame itself.

\* Set y-axis at the required

\* incline.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

glTranslatef(TOP\_LEN,0.0f,0.0f); //Supporting rod position

glRotatef(-FRONT\_INCLINE,0.0f,0.0f,1.0f);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* I Hope the handle can rotate

\* about its mean position

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

glPushMatrix();

glRotatef(-steering/2.0,1.0f,0.0f,0.0f);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Roll back to the height of

\* the handle to draw it

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

glTranslatef(-0.3f,0.0f,0.0f);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* We cannot use the incline

\* the incline to draw the

\* horizontal part of the rod

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

glPushMatrix();

glRotatef(FRONT\_INCLINE,0.0f,0.0f,1.0f);

glPushMatrix();

glTranslatef(-0.6f,0.5f,-HANDLE\_ROD/2);

ZCylinder(ROD\_RADIUS,HANDLE\_ROD);

glPopMatrix();

glPushMatrix();

glColor3f(1.0f,1.0f,0.0f);

glTranslatef(-0.6f,0.5f,-HANDLE\_ROD/2); //Handle

ZCylinder(0.07f,HANDLE\_ROD/4); //Rods

glTranslatef(0.0f,0.0f,HANDLE\_ROD\*3/4); //Are

ZCylinder(0.07f,HANDLE\_ROD/4); //Drawn

glColor3f(1.0f,0.0f,0.0f); //Here

glPopMatrix();

glPopMatrix();

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Using this incline now draw

\* the handle.Maybe use this again

\* to draw the wheel.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

glPushMatrix();

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Draw the main big rod

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

glTranslatef(-0.75,0.0,0.0);

XCylinder(ROD\_RADIUS,FRONT\_ROD);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Why not draw the two rods and

\* the WHEEL?

\* Yes!So,first go to the

\* end of the main rod.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

glTranslatef(CRANK\_ROD,0.0f,0.0f); //Position set to the end of the main big connector rod in the front to draw the front connector rods at that position

glRotatef(CRANK\_ANGLE,0.0f,0.0f,1.0f);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Draw the two rods connecting

\* the handle and the front

\* wheel.

\* The two rods are at a incline

\* to the connector.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

glPushMatrix();

glTranslatef(0.0f,0.0f,WHEEL\_OFFSET-0.35);

XCylinder(ROD\_RADIUS,CRANK\_RODS);

glPopMatrix();

glPushMatrix();

glTranslatef(0.0f,0.0f,-WHEEL\_OFFSET+0.35);

XCylinder(ROD\_RADIUS,CRANK\_RODS);

glPopMatrix();

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Why not draw the wheel.

\* The FRONT wheel to be precise

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

glTranslatef(CRANK\_RODS,0.0f,0.0f);

glRotatef(-2\*pedalAngle,0.0f,0.0f,1.0f);

drawTyre();

glPopMatrix();

glPopMatrix(); /\* End of the rotation of the handle effect \*/

glPopMatrix();

//Draw the bulb for the headlight

glPushMatrix();

glColor3f(1.0,1.0,0.0);

glRotatef(360.0,1.0,0.0,0.0);

glTranslatef(1.0,2.6,0.0);

glutSolidSphere(0.2,160.0,180.0);

glPopMatrix();

//Draw the human on the bike

//Draw the head

glPushMatrix();

glRotatef(-steering/2.0,1.0f,0.0f,0.0f);

glColor3f(0.5f,0.3f,0.0f);

glRotatef(180.0,1.0,0.0,0.0);

glTranslatef(-1.0,-3.5f,0.0f);

glutSolidSphere(0.4,160.0,180.0);

glPopMatrix();

//Draw the eyes

glPushMatrix();

glRotatef(-steering/2.0,1.0f,0.0f,0.0f);

glColor3f(1.0f,1.0f,1.0f);

glRotatef(180.0,1.0,0.0,0.0);

glTranslatef(-0.6,-3.6f,-0.15f);

glutSolidSphere(0.05,160.0,180.0); //Right white eye

glPopMatrix();

glPushMatrix();

glRotatef(-steering/2.0,1.0f,0.0f,0.0f);

glColor3f(1.0f,1.0f,1.0f);

glRotatef(180.0,1.0,0.0,0.0);

glTranslatef(-0.6,-3.6f,0.15f);

glutSolidSphere(0.05,160.0,180.0); //Left white eye

glPopMatrix();

glPushMatrix();

glRotatef(-steering/2.0,1.0f,0.0f,0.0f);

glColor3f(0.0f,0.0f,0.0f);

glRotatef(180.0,1.0,0.0,0.0);

glTranslatef(-0.6,-3.6f,-0.15f);

glutSolidSphere(0.025,160.0,180.0); //Right black eye lid

glPopMatrix();

glPushMatrix();

glRotatef(-steering/2.0,1.0f,0.0f,0.0f);

glColor3f(0.0f,0.0f,0.0f);

glRotatef(180.0,1.0,0.0,0.0);

glTranslatef(-0.6,-3.6f,0.15f);

glutSolidSphere(0.025,160.0,180.0); //Left black eye lid

glPopMatrix();

//Draw the nose

glPushMatrix();

glRotatef(-steering/2.0,1.0f,0.0f,0.0f);

glColor3f(0.5f,0.3f,0.0f);

glRotatef(-270.0,-1.0,0.0,0.0);

glTranslatef(-0.6,0.0f,3.4f);

glutSolidCone(0.1,0.2,50.0,50.0);

glPopMatrix();

//Draw the mouth

glPushMatrix();

glRotatef(-steering/2.0,1.0f,0.0f,0.0f);

glColor3f(1.0f,0.0f,1.0f); //It is made pink to indicate the rider's lips

glRotatef(90.0,0.0,1.0,0.0);

glTranslatef(-0.1,3.3f,-0.6f);

XCylinder(ROD\_RADIUS,0.25f);

glPopMatrix();

//Draw the upper part of the body

glPushMatrix();

glRotatef(-steering/4.0,1.0f,0.0f,0.0f);

glColor3f(0.0f,0.0f,0.0f);

glRotatef(80.0,0.0,0.0,1.0);

glTranslatef(1.5,1.6,0.0);

XCylinder(GAS\_TANK+0.15,1.5);

glPopMatrix();

//Draw the hands

glPushMatrix();

glRotatef(-steering/4.0,1.0f,0.0f,0.0f);

glColor3f(1.0f,1.0f,1.0f);

glRotatef(150.0,0.0,0.0,1.0);

glTranslatef(1.8,-2.0,-0.6);

XCylinder(GAS\_TANK-0.1,1.0); //Left upper arm

glPopMatrix();

glPushMatrix();

glRotatef(-steering/4.0,1.0f,0.0f,0.0f);

glColor3f(1.0f,1.0f,1.0f);

glRotatef(150.0,0.0,0.0,1.0);

glTranslatef(1.8,-2.0,0.6);

XCylinder(GAS\_TANK-0.1,1.0); //Right upper arm

glPopMatrix();

glPushMatrix();

glRotatef(-steering/4.0,1.0f,0.0f,0.0f);

glColor3f(0.5f,0.3f,0.0f);

glRotatef(180.0,0.0,0.0,1.0);

glTranslatef(-0.3,-2.6,-0.6);

XCylinder(GAS\_TANK-0.1,1.0); //Left lower arm

glPopMatrix();

glPushMatrix();

glRotatef(-steering/4.0,1.0f,0.0f,0.0f);

glColor3f(0.5f,0.3f,0.0f);

glRotatef(180.0,0.0,0.0,1.0);

glTranslatef(-0.3,-2.6,0.6);

XCylinder(GAS\_TANK-0.1,1.3); //Right lower arm

glPopMatrix();

//Draw the legs

glPushMatrix();

glColor3f(1.0f,1.0f,1.0f);

glRotatef(180.0,0.0,0.0,1.0);

glTranslatef(0.3,-1.6,0.6);

XCylinder(GAS\_TANK-0.1,1.5); //Right upper leg

glPopMatrix();

glPushMatrix();

glColor3f(1.0f,1.0f,1.0f);

glRotatef(180.0,0.0,0.0,1.0);

glTranslatef(0.3,-1.6,-0.6);

XCylinder(GAS\_TANK-0.1,1.5); //Left upper leg

glPopMatrix();

glPushMatrix();

glColor3f(0.5f,0.3f,0.0f);

glRotatef(100.0,0.0,0.0,1.0);

glTranslatef(0.0,0.0,-0.6);

XCylinder(GAS\_TANK-0.1,1.7); //Left lower leg

glPopMatrix();

glPushMatrix();

glColor3f(0.5f,0.3f,0.0f);

glRotatef(100.0,0.0,0.0,1.0);

glTranslatef(0.0,0.0,0.6);

XCylinder(GAS\_TANK-0.1,1.7); //Right lower leg

glPopMatrix();

}

/\*

\* Draw a gear wheel. You'll probably want to call this function when

\* building a display list since we do a lot of trig here.

\*

\* Input: inner\_radius - radius of hole at center

\* outer\_radius - radius at center of teeth

\* width - width of gear

\* teeth - number of teeth

\* tooth\_depth - depth of tooth

\*/

void gear( GLfloat inner\_radius, GLfloat outer\_radius, GLfloat width,

GLint teeth, GLfloat tooth\_depth )

{

GLint i;

GLfloat r0, r1, r2;

GLfloat angle, da;

GLfloat u, v, len;

const double pi = 3.14159264;

r0 = inner\_radius;

r1 = outer\_radius - tooth\_depth/2.0;

r2 = outer\_radius + tooth\_depth/2.0;

da = 2.0\*pi / teeth / 4.0;

glShadeModel( GL\_FLAT ); //Select flat(matt finished) or smooth shading

glNormal3f( 0.0, 0.0, 1.0 ); //For flat shading,glNormal3f() has to be called for each face

/\* draw front face \*/

glBegin( GL\_QUAD\_STRIP ); //For indicating the sharpness of gear teeth

for (i=0;i<=teeth;i++) {

angle = i \* 2.0\*pi / teeth;

glVertex3f( r0\*cos(angle), r0\*sin(angle), width\*0.5 );

glVertex3f( r1\*cos(angle), r1\*sin(angle), width\*0.5 );

glVertex3f( r0\*cos(angle), r0\*sin(angle), width\*0.5 );

glVertex3f( r1\*cos(angle+3\*da), r1\*sin(angle+3\*da), width\*0.5 );

}

glEnd();

/\* draw front sides of teeth \*/

glBegin( GL\_QUADS );

da = 2.0\*pi / teeth / 4.0;

for (i=0;i<teeth;i++) {

angle = i \* 2.0\*pi / teeth;

glVertex3f( r1\*cos(angle), r1\*sin(angle), width\*0.5 );

glVertex3f( r2\*cos(angle+da), r2\*sin(angle+da), width\*0.5 );

glVertex3f( r2\*cos(angle+2\*da), r2\*sin(angle+2\*da), width\*0.5 );

glVertex3f( r1\*cos(angle+3\*da), r1\*sin(angle+3\*da), width\*0.5 );

}

glEnd();

glNormal3f( 0.0, 0.0, -1.0 );

/\* draw back face \*/

glBegin( GL\_QUAD\_STRIP );

for (i=0;i<=teeth;i++) {

angle = i \* 2.0\*pi / teeth;

glVertex3f( r1\*cos(angle), r1\*sin(angle), -width\*0.5 );

glVertex3f( r0\*cos(angle), r0\*sin(angle), -width\*0.5 );

glVertex3f( r1\*cos(angle+3\*da), r1\*sin(angle+3\*da), -width\*0.5 );

glVertex3f( r0\*cos(angle), r0\*sin(angle), -width\*0.5 );

}

glEnd();

/\* draw back sides of teeth \*/

glBegin( GL\_QUADS );

da = 2.0\*pi / teeth / 4.0;

for (i=0;i<teeth;i++) {

angle = i \* 2.0\*pi / teeth;

glVertex3f( r1\*cos(angle+3\*da), r1\*sin(angle+3\*da), -width\*0.5 );

glVertex3f( r2\*cos(angle+2\*da), r2\*sin(angle+2\*da), -width\*0.5 );

glVertex3f( r2\*cos(angle+da), r2\*sin(angle+da), -width\*0.5 );

glVertex3f( r1\*cos(angle), r1\*sin(angle), -width\*0.5 );

}

glEnd();

/\* draw outward faces of teeth \*/

glBegin( GL\_QUAD\_STRIP );

for (i=0;i<teeth;i++) {

angle = i \* 2.0\*pi / teeth;

glVertex3f( r1\*cos(angle), r1\*sin(angle), width\*0.5 );

glVertex3f( r1\*cos(angle), r1\*sin(angle), -width\*0.5 );

u = r2\*cos(angle+da) - r1\*cos(angle);

v = r2\*sin(angle+da) - r1\*sin(angle);

len = sqrt( u\*u + v\*v );

u /= len;

v /= len;

glNormal3f( v, -u, 0.0 );

glVertex3f( r2\*cos(angle+da), r2\*sin(angle+da), width\*0.5 );

glVertex3f( r2\*cos(angle+da), r2\*sin(angle+da), -width\*0.5 );

glNormal3f( cos(angle), sin(angle), 0.0 );

glVertex3f( r2\*cos(angle+2\*da), r2\*sin(angle+2\*da), width\*0.5 );

glVertex3f( r2\*cos(angle+2\*da), r2\*sin(angle+2\*da), -width\*0.5 );

u = r1\*cos(angle+3\*da) - r2\*cos(angle+2\*da);

v = r1\*sin(angle+3\*da) - r2\*sin(angle+2\*da);

glNormal3f( v, -u, 0.0 );

glVertex3f( r1\*cos(angle+3\*da), r1\*sin(angle+3\*da), width\*0.5 );

glVertex3f( r1\*cos(angle+3\*da), r1\*sin(angle+3\*da), -width\*0.5 );

glNormal3f( cos(angle), sin(angle), 0.0 );

}

glVertex3f( r1\*cos(0.0), r1\*sin(0.0), width\*0.5 );

glVertex3f( r1\*cos(0.0), r1\*sin(0.0), -width\*0.5 );

glEnd();

glShadeModel( GL\_SMOOTH );

/\* draw inside radius cylinder \*/

glBegin( GL\_QUAD\_STRIP );

for (i=0;i<=teeth;i++) {

angle = i \* 2.0\*pi / teeth;

glNormal3f( -cos(angle), -sin(angle), 0.0 );

glVertex3f( r0\*cos(angle), r0\*sin(angle), -width\*0.5 );

glVertex3f( r0\*cos(angle), r0\*sin(angle), width\*0.5 );

}

glEnd();

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* The chain has been done

\* by using the trippled lines strip.

\* But all those alternate line strips

\* are drawn such that those will sit

\* on the teeth only

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void drawChain()

{

GLfloat depth;

static int mode=0;

glColor3f(0.0f,1.0f,0.0f);

glEnable(GL\_LINE\_STIPPLE);

mode=(mode+1)%2;

if(mode==0 && speed>0)

glLineStipple(1,0x1c47); //Used ASCII value for situating on the teeth exactly while bike is at rest

else if(mode==1 && speed>0)

glLineStipple(1,0x00FF); //Used ASCII value for situating on the teeth exactly while bike is moving

glBegin(GL\_LINES);

for(depth=0.06f;depth<=0.12f;depth+=0.01f)

{

glVertex3f(-1.6f,0.15f,ROD\_RADIUS);

glVertex3f(0.0f,0.3f,depth);

glVertex3f(-1.6f,-0.15f,ROD\_RADIUS);

glVertex3f(0.0f,-0.3f,depth);

}

glEnd();

glDisable(GL\_LINE\_STIPPLE); //Enable or disable server-side GL capabilities

}

void drawSeat() //Drawn somewhat to similar seat for a single seater

{

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Draw the top of the seat

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

glBegin(GL\_POLYGON);

glVertex3f(-0.1f, 1.0f, -0.5f);

glVertex3f( 1.0f, 1.0f, -0.3f);

glVertex3f( 1.0f, 1.0f, 0.3f);

glVertex3f(-0.1f, 1.0f, 0.5f);

glVertex3f(-0.5f, 1.0f, 1.0f);

glVertex3f(-1.0f, 1.0f, 1.0f);

glVertex3f(-1.0f, 1.0f, -1.0f);

glVertex3f(-0.5f, 1.0f, -1.0f);

glEnd();

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Draw the bottom base part of the

\* seat

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

glBegin(GL\_POLYGON);

glVertex3f(-0.1f, -1.0f, -0.5f);

glVertex3f( 1.0f, -1.0f, -0.3f);

glVertex3f( 1.0f, -1.0f, 0.3f);

glVertex3f(-0.1f, -1.0f, 0.5f);

glVertex3f(-0.5f, -1.0f, 1.0f);

glVertex3f(-1.0f, -1.0f, 1.0f);

glVertex3f(-1.0f, -1.0f, -1.0f);

glVertex3f(-0.5f, -1.0f, -1.0f);

glEnd();

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Draw the sides!

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

glBegin(GL\_QUADS);

glVertex3f(1.0f,1.0f,-0.3f);

glVertex3f(1.0f,1.0f,0.3f);

glVertex3f(1.0f,-1.0f,0.3f);

glVertex3f(1.0f,-1.0f,-0.3f);

glVertex3f(1.0f,1.0f,0.3f);

glVertex3f(-0.1f,1.0f,0.5f);

glVertex3f(-0.1f,-1.0f,0.5f);

glVertex3f(1.0f,-1.0f,0.3f);

glVertex3f(1.0f,1.0f,-0.3f);

glVertex3f(-0.1f,1.0f,-0.5f);

glVertex3f(-0.1f,-1.0f,-0.5f);

glVertex3f(1.0f,-1.0f,-0.3f);

glVertex3f(-0.1f,1.0f,0.5f);

glVertex3f(-0.5f,1.0f,1.0f);

glVertex3f(-0.5f,-1.0f,1.0f);

glVertex3f(-0.1f,-1.0f,0.5f);

glVertex3f(-0.1f,1.0f,-0.5f);

glVertex3f(-0.5f,1.0f,-1.0f);

glVertex3f(-0.5f,-1.0f,-1.0f);

glVertex3f(-0.1f,-1.0f,-0.5f);

glVertex3f(-0.5f,1.0f,1.0f);

glVertex3f(-1.0f,1.0f,1.0f);

glVertex3f(-1.0f,-1.0f,1.0f);

glVertex3f(-0.5f,-1.0f,1.0f);

glVertex3f(-0.5f,1.0f,-1.0f);

glVertex3f(-1.0f,1.0f,-1.0f);

glVertex3f(-1.0f,-1.0f,-1.0f);

glVertex3f(-0.5f,-1.0f,-1.0f);

glVertex3f(-1.0f,1.0f,1.0f);

glVertex3f(-1.0f,1.0f,-1.0f);

glVertex3f(-1.0f,-1.0f,-1.0f);

glVertex3f(-1.0f,-1.0f,1.0f);

glEnd();

}

void drawPedals() //Drawn for foot rest

{

glColor3f(0.0f,0.0f,1.0f);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Lets draw the two pedals

\* offset from the center

\* of the frame.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* First draw the one visible

\* to the viewer

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

glPushMatrix();

glTranslatef(0.0f,0.0f,0.105f);

glTranslatef(0.25f,0.0f,0.0f);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Draw the pedal rod

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

glPushMatrix();

glScalef(0.5f,0.1f,0.1f);

glutSolidCube(1.0f);

glPopMatrix();

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Draw the resting pad

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

glPushMatrix();

glTranslatef(0.25f,0.0f,0.15f);

glScalef(0.2f,0.02f,0.3f);

glutSolidCube(1.0f);

glPopMatrix();

glPopMatrix();

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Draw the one on the other

\* side of the frame

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

glPushMatrix();

glTranslatef(0.0f,0.0f,-0.105f);

glTranslatef(0.25f,0.0f,0.0f);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Now again draw the pedal

\* rod

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

glPushMatrix();

glScalef(0.5f,0.1f,0.1f);

glutSolidCube(1.0f);

glPopMatrix();

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Draw the resting pad of

\* the pedal

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

glPushMatrix();

glTranslatef(0.25f,0.0f,-0.15f);

glScalef(0.2f,0.02f,0.3f);

glutSolidCube(1.0f);

glPopMatrix();

glPopMatrix();

glColor3f(1.0f,0.0f,0.0f);

}

void drawTyre(void)

{

int i;

// Draw The Disc

glColor3f(1.0f,1.0f,1.0f);

glutSolidTorus(-0.17,0.29,100,100);

// Draw The Rim

glColor3f(1.0f,1.0f,1.0f);

glutSolidTorus(-0.05f,-0.75f,100,100);

// Draw The Central Cylinder

// Length of cylinder 0.12f

glColor3f(1.0f,1.0f,0.5f);

glPushMatrix();

glTranslatef(0.0f,0.0f,-0.06f);

ZCylinder(0.02f,0.12f);

glPopMatrix();

glutSolidTorus(0.02f,0.02f,3,20);

// Draw The Spokes

// Note: Spokes are not made thicker

glColor3f(1.0f,1.0f,1.0f);

for(i=0;i<NUM\_SPOKES;++i)

{

glPushMatrix();

glRotatef(i\*SPOKE\_ANGLE,0.0f,0.0f,1.0f);

glBegin(GL\_LINES);

glVertex3f(0.0f,0.02f,0.0f);

glVertex3f(0.0f,0.86f,0.0f);

glEnd();

glPopMatrix();

}

// Draw The Tyre

glColor3f(0.0f,0.0f,0.0f);

glutSolidTorus(TUBE\_WIDTH,RADIUS\_WHEEL,10,30);

glColor3f(1.0f,0.0f,0.0f);

}

void init()

{

GLfloat mat\_specular[]={1.0,1.0,1.0,1.0};

GLfloat mat\_shininess[]={100.0};

GLfloat light\_directional[]={1.0,1.0,1.0,1.0};

GLfloat light\_positional[]={1.0,1.0,1.0,1.0};

GLfloat light\_diffuse[]={1.0,1.0,1.0};

reset();

glShadeModel(GL\_SMOOTH);

glLightfv(GL\_LIGHT0,GL\_POSITION,light\_directional);

glLightfv(GL\_LIGHT0,GL\_AMBIENT,light\_diffuse);

glLightfv(GL\_LIGHT0,GL\_DIFFUSE,light\_diffuse);

glMaterialfv(GL\_FRONT,GL\_SHININESS,mat\_shininess);

glMaterialfv(GL\_FRONT,GL\_SPECULAR,mat\_specular);

glColorMaterial(GL\_FRONT,GL\_DIFFUSE);

glEnable(GL\_LIGHTING);

glEnable(GL\_LIGHT0);

glEnable(GL\_COLOR\_MATERIAL);

glEnable(GL\_DEPTH\_TEST);

}

void landmarks(void) //Created using a rectangular mesh

{

GLfloat i;

glColor3f(0.8f,0.4f,0.3f);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Draw the ground for the bike

\* Looks incomplete with it!Don't

\* forget to define the normal

\* vectors for the vertices.

\* gotta fix this bug!

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

glBegin(GL\_LINES);

for(i=-100.0f ; i<100.0f ; i += 1.0f)

{

glVertex3f(-100.0f,-RADIUS\_WHEEL,i);

glVertex3f( 100.0f,-RADIUS\_WHEEL,i);

glVertex3f(i,-RADIUS\_WHEEL,-100.0f);

glVertex3f(i,-RADIUS\_WHEEL,100.0f);

}

glEnd();

}

void display\_bike(void)

{

glClear(GL\_COLOR\_BUFFER\_BIT | GL\_DEPTH\_BUFFER\_BIT);

glEnable(GL\_NORMALIZE);

glPushMatrix();

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Prepare the rotations

\* and start doing the

\* remaining scene

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

glRotatef(angley,1.0f,0.0f,0.0f);

glRotatef(anglex,0.0f,1.0f,0.0f);

glRotatef(anglez,0.0f,0.0f,1.0f);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Start rendering

\* the scene of

\* the bike

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

landmarks();

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Move the bike.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

glPushMatrix();

glTranslatef(xpos,0.0f,zpos);

glRotatef(direction,0.0f,1.0f,0.0f);

drawFrame();

drawChain();

glPopMatrix();

glPopMatrix();

glMatrixMode(GL\_MODELVIEW);

glLoadIdentity();

gluLookAt(camx,camy,camz, camx,1.0,0.0,0.0,1.0,1.0);

glutSwapBuffers();

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Returns the absolute

\* value of a given

\* float number

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

GLfloat Abs(GLfloat a)

{

if(a < 0.0f)

return -a;

else

return a;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Returns the value of

\* the given angle in

\* degrees

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

GLfloat degrees(GLfloat a)

{

return a\*180.0f/PI;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Returns the value of

\* the given angle in

\* radians

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

GLfloat radians(GLfloat a)

{

return a\*PI/180.0f;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* The idle function of

\* the program which makes

\* the contniuous loop

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void idle(void)

{

updateScene();

glutPostRedisplay();

}

void special(int key,int x,int y)

{

switch(key)

{

case GLUT\_KEY\_UP:

camz -= 0.1f;

break;

case GLUT\_KEY\_DOWN:

camz += 0.1f;

break;

case GLUT\_KEY\_LEFT:

camx -= 0.1f;

break;

case GLUT\_KEY\_RIGHT:

camx += 0.1f;

break;

}

glutPostRedisplay();

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Reset The scene

\* Bringing the bike

\* back to its initial position

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void reset()

{

glColor3f(1.0f,1.0f,1.0f);

anglex=angley=anglez=0.0f;

pedalAngle=steering=0.0f;

Mouse=GLUT\_UP;

pedalAngle=speed=steering=0.0f;

camx=camy=0.0f;

camz=5.0f;

xpos=zpos=0.0f;

direction=0.0f;

glPushMatrix();

glColor3f(0.5f,0.3f,0.0f);

glRotatef(100.0,0.0,0.0,1.0);

glTranslatef(-1.0,0.0,-0.6);

XCylinder(GAS\_TANK-0.1,2.7);

glPopMatrix();

}

void keyboard(unsigned char key,int x,int y)

{

GLfloat r=0.0f,g=0.0f;

switch(key)

{

case 'r':

case 'R':

reset();

break;

case 's':

case 'S':

glutDisplayFunc(operations\_window);

break;

case 'c':

case 'C':

glutDisplayFunc(display\_bike);

break;

case '1':

if(steering < HANDLE\_LIMIT)

steering += INC\_STEERING;

break;

case '2':

if(steering > -HANDLE\_LIMIT)

steering -= INC\_STEERING;

break;

case '+':

speed += INC\_SPEED;

break;

case '-':

speed -= INC\_SPEED;

break;

case 27:

exit(1);

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Where is my Bike?

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* When you rotate the handle the

\* bike as a whole does not rotate

\* at once immediately.

\* For each unit of time, the \* handle slowly begins to align

\* with the rest of the body of the

\* bike.

\* I Think that the rate at which the

\* handle aligns with the body is

\* dependant on the speed too!!

\* The rate is given by 'delta'

\* and the speed is given by 'speed'

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Check out the error

\* conditions ;>

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

pedalAngle += speed;

if(speed < 0.0f)

speed = 0.0f;

if(pedalAngle < 0.0f)

pedalAngle = 0.0f;

if(pedalAngle >= 360.0f)

pedalAngle -= 360.0f;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Go! Display

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

glutPostRedisplay();

}

void mouse(int button,int state,int x,int y)

{

switch(button)

{

case GLUT\_LEFT\_BUTTON:

if(state==GLUT\_DOWN)

{

Mouse=GLUT\_DOWN;

prevx=x;

prevy=y;

}

if(state==GLUT\_UP)

{

Mouse=GLUT\_UP;

}

break;

case GLUT\_RIGHT\_BUTTON:

/\* DO NOTHING \*/

break;

}

glutPostRedisplay();

}

void passive(int x,int y)

{

/\* DO NOTHING \*/

}

void motion(int x,int y) {

if(Mouse==GLUT\_DOWN)

{

int deltax,deltay;

deltax=prevx-x;

deltay=prevy-y;

anglex += 0.5\*deltax;

angley += 0.5\*deltay;

if(deltax!=0 && deltay!=0)

anglez += 0.5\*sqrt(deltax\*deltax + deltay\*deltay);

if(anglex < 0)

anglex+=360.0;

if(angley < 0)

angley+=360.0;

if(anglez < 0)

anglez += 360.0;

if(anglex > 360.0)

anglex-=360.0;

if(angley > 360.0)

angley-=360.0;

if(anglez > 360.0)

anglez-=360.0;

}

else

{

Mouse=GLUT\_UP;

}

prevx=x;

prevy=y;

glutPostRedisplay();

}

void reshape(int w,int h)

{

glViewport(0,0,(GLsizei)w,(GLsizei)h);

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

gluPerspective(60.0,(GLfloat)w/(GLfloat)h,0.1,100.0);

//Angle,Aspect Ratio,near plane,far plane

glMatrixMode(GL\_MODELVIEW);

glLoadIdentity();

gluLookAt(camx,camy,camz, 0.0,0.0,0.0, 0.0,1.0,0.0);

}

void bitmap\_output(float x, float y, float z, char \*string)

{

int len, i;

glRasterPos3f(x,y,z);

len = (int) strlen(string);

for (i = 0; i < len; i++) {

glutBitmapCharacter(GLUT\_BITMAP\_TIMES\_ROMAN\_24, string[i]);

}

}

void welcome\_window()

{

glClear(GL\_COLOR\_BUFFER\_BIT|GL\_DEPTH\_BUFFER\_BIT);

glClearColor(0,0,0,0);

glColor3f(1.0,1.0,1.0);

bitmap\_output(-1.25,1.8,0.50,"VISVESVARAYA TECHNOLOGICAL UNIVERSITY");

bitmap\_output(-0.6,1.6,0.50,"BELGAUM,KARNATAKA");

bitmap\_output(-0.3,0.70,0.50,"Project On");

bitmap\_output(-0.85,0.50,0.50,"'SIMULATION OF A 3D BIKE'");

bitmap\_output(-0.6,-1.5,0.50,"PLEASE PRESS S TO START");

glutSwapBuffers();

glFlush();

}

void operations\_window()

{

glClear(GL\_COLOR\_BUFFER\_BIT|GL\_DEPTH\_BUFFER\_BIT);

glClearColor(0,0,0,0);

glColor3f(1.0,1.0,1.0);

bitmap\_output(-1.25,1.7,0.50,"OPERATIONS THAT CAN BE PERFORMED BY THE BIKE");

bitmap\_output(-1.25,1.2,0.50,"1. RESET THE CAMERA - USE 'R' OR 'r'");

bitmap\_output(-1.25,1.0,0.50,"2. ACCELERATE THE BIKE - USE '+'");

bitmap\_output(-1.25,0.8,0.50,"3. DEACCELERATE THE BIKE - USE '-'");

bitmap\_output(-1.25,0.6,0.50,"4. TURN RIGHT - USE '2'");

bitmap\_output(-1.25,0.4,0.50,"5. TURN LEFT - USE '1'");

bitmap\_output(-1.25,0.2,0.50,"6. ZOOM IN - USE 'UPWARD ARROW'");

bitmap\_output(-1.25,0.0,0.50,"7. ZOOM OUT - USE 'DOWNWARD ARROW'");

bitmap\_output(-1.25,-0.2,0.50,"8. MOVE LEFT - USE 'LEFT ARROW'"); bitmap\_output(-1.25,-0.4,0.50,"9. MOVE RIGHT - USE 'RIGHT ARROW'");

bitmap\_output(-1.25,-0.6,0.50,"10. USE MOUSE TO CHANGE THE ANGLE OF VIEWING");

bitmap\_output(-1.25,-1.0,0.50,"PLEASE PRESS C TO CONTINUE");

glutSwapBuffers();

glFlush();

}

int main(int argc,char \*argv[])

{

glutInit(&argc,argv);

glutInitDisplayMode(GLUT\_DOUBLE | GLUT\_RGB);

glutInitWindowPosition(100,100);

glutInitWindowSize(WIN\_WIDTH,WIN\_HEIGHT);

glutCreateWindow("Bike");

init();

glutDisplayFunc(welcome\_window);

glutReshapeFunc(reshape);

glutIdleFunc(idle);

glutSpecialFunc(special);

glutKeyboardFunc(keyboard);

glutMouseFunc(mouse);

glutMotionFunc(motion);

glutPassiveMotionFunc(passive);

glutSetCursor(GLUT\_CURSOR\_CROSSHAIR);

glutMainLoop();

}