

```

// Global scene parameters

SET SCENE_FOV = 60;
SET SCENE_NEAR = 0.5;
SET SCENE_FAR = 1000;

// Need includes

#include <Script3d.h>
#include <GLConst.h>
#include <Impulse.h>
#include "xvrgolite.h.s3d" //haptics
//#include "novodexvr.h.s3d" //physics
#include "hapticweb.h.s3d" //haptics

// Global variables
// Camera globals
var CamPos = [0.0,320.0,270.0]; //camera position
var CamDir = [0,-0.6,-0.8]; //camera direction (it is a vector)

var test10 = 0;

var real_rand = 0;
var peg_touch0 = 1;
var peg_touch1 = 1;
var peg_touch2 = 1;
var peg_touch3 = 1;
var peg_touch4 = 1;
var peg_touch5 = 1;
var peg_touch6 = 1;
var peg_touch7 = 1;
var peg_touch8 = 1;

var originX = -300;
var originY = -100;
var originZ = -300;

//fitts equations constants
var A_1_healthy = 0.0177;
var A_2_healthy = 0.3486;
var A_1_MS = -0.5446;
var A_2_MS = 0.5494;

var RMS0 = 0;
var RMS1 = 0;
var RMS2 = 0;
var RMS3 = 0;
var RMS4 = 0;
var RMS5 = 0;
var RMS6 = 0;
var RMS7 = 0;
var RMS8 = 0;

```

```
var Ratio0 = 0;
var Ratio1 = 0;
var Ratio2 = 0;
var Ratio3 = 0;
var Ratio4 = 0;
var Ratio5 = 0;
var Ratio6 = 0;
var Ratio7 = 0;
var Ratio8 = 0;

var RMS0_t = 0;
var RMS1_t = 0;
var RMS2_t = 0;
var RMS3_t = 0;
var RMS4_t = 0;
var RMS5_t = 0;
var RMS6_t = 0;
var RMS7_t = 0;
var RMS8_t = 0;

var Sm_t = Array(1000000);
var Sm = Array(1000000);
var count0 = 0;
var counter = 0;

var count1 = 0;
var h_0_count = 0;
var h_1_count = 0;
var h_2_count = 0;
var h_3_count = 0;
var h_4_count = 0;
var h_5_count = 0;
var h_6_count = 0;
var h_7_count = 0;

var y_int_loc = 0;
var x_int_loc = 0;

var i_p = 0;

var x_mjm_loc = 0;
var y_mjm_loc = 0;
var z_mjm_loc = 0;
var x_mjm_loc_test = 0;

var x_mjm = 0;
var y_mjm = 0;
var z_mjm = 0;

var x_mjm_loc_MS = 0;
```

```
var y_mjm_loc_MS = 0;
var z_mjm_loc_MS = 0;
var x_mjm_loc_test_MS = 0;

var x_mjm_MS = 0;
var y_mjm_MS = 0;
var z_mjm_MS = 0;

var t_fitts = 1000;
var t_fitts_MS = 1000;
var tau_start = 0.000;
var tau_start_stop = 0;

var epsilon0 = 100;

var tau0 = 0.0000;
var tau0_MS = 0.0000;
var tau0_int = 0.000;
var tau1_int = 0.000;
var S0_I_mjm = 0;
var nS = 5;

var x1_int = 0.00;
var y1_int = 0.00;

var x2_int = 0.00;
var y2_int = 0.00;

var iS = 1;

var tau1 = 0;
var tau2 = 0;
var tau3 = 0;
var tau4 = 0;
var tau5 = 0;
var tau6 = 0;
var tau7 = 0;
var tau8 = 0;

var delay_calc = 500;
var delay_force = 50;

var rand_angle = 0;

var rand_angle0 = 0;
var rand_angle1 = 0;
var rand_angle2 = 0;
var rand_angle3 = 0;
var rand_angle4 = 0;
```

```

var rand_angle5 = 0;
var rand_angle6 = 0;
var rand_angle7 = 0;
var rand_angle8 = 0;
var rand_angle9 = 0;

var check10 = 150;
var i_m = 0;
var drawline = 0;
// interaction parameters
var tolerance =19; // acceptable minimum difference between reference shape and peg for illumination
var plane_tolerance = 3.2;
var hole_diameter2 =7; //haptic diameter of hole
var hole_tolerance = 1;
var hole_diameter = hole_diameter2; // just a rename.. not sure why i did this
//object related parameters
var h_world; //the mother of all haptic objects!
var board; //board object
var board2; //the pegs appear on this board...
var peg; //peg object
var peg_0,peg_1,peg_2,peg_3,peg_4,peg_5,peg_6,peg_7,peg_8; //nine pegs that are dr
var hand_char;
var hand_mesh;
var hand_avatar;
var peg_mesh4;
var peg0,peg1,peg2,peg3,peg4,peg5,peg6,peg7,peg8;
var peg0_t,peg1_t,peg2_t,peg3_t,peg4_t,peg5_t,peg6_t,peg7_t,peg8_t;
var peg_t;
var h_0,h_1,h_2,h_3,h_4,h_5,h_6,h_7,h_8, h_all;
var th_0,th_1,th_2,th_3,th_4,th_5,th_6,th_7,th_8;
var h_0out = 0;
var h_1out = 0;
var h_2out = 0;
var h_3out = 0;
var h_4out = 0;
var h_5out = 0;
var h_6out = 0;
var h_7out = 0;
var h_8out = 0;

var xn = 0;
var zn = 0;
var yn = 0;

var ind = 0;

var Alpha = 0;
var Beta = 0;
var Alpha2 = 0;
var Beta2 = 0;
var angle = 0;
var angle2 = 0;

```

```
var intersect_x = 0;
var intersect_z = 0;

var rdz = 0;
var rdx = 0;
var rdhx = 0;
var rdhz = 0;

var x_1 = 0;
var y_1 = 0;
var z_1 = 0;

var x_2 = 0;
var y_2 = 0;
var z_2 = 0;

var x_3 = 0;
var y_3 = 0;
var z_3 = 0;

var x_I = 0;
var y_I = 0;
var z_I = 0;

var x_H = 0;
var y_H = 0;
var z_H = 0;

var zH_loc = 0;
var xH_loc = 0;
var yH_loc = 0;
var yH_loc_prev = 0;

var k_plane0 = 0;
var k_plane1 = 0;
var k_plane2 = 0;
var k_plane33 = 0;
var k_plane4 = 0;
var k_plane5 = 0;
var k_plane6 = 0;
var k_plane7 = 0;
var k_plane8 = 0;

var RMSmax = 5;
var k_plane_max = 0;
var F_max = 30;
var plane_dist_av_max = 0;

var RMSmax_time = 5;
var k_time_max = 0;
var S0_I_mjm_av_max = 0;
```

```
var k_norm = 2000*10^-3;//stif_peg;  
var k_plane = 2000*10^-3;//2000*10^-3;//2000*10^-3;//stif_peg;  
var k_plane3 = 0;//0.001;  
var k_time = 2000*10^-3;//stif_peg;
```

```
var b_norm = 0;
```

```
var b_mjm = 0;  
var D_dist_mjm = 0;
```

```
var S0_I_mjm_prev = 0;
```

```
var b_plane = 0;
```

```
var check_dot_12_23 = 1;
```

```
var x_Hloc;
```

```
var x_1d = 0;  
var y_1d = 0;  
var z_1d = 0;
```

```
var phid = 320;  
var touch0 = 0;  
var touch1 = 0;  
var touch2 = 0;  
var touch3 = 0;  
var touch4 = 0;  
var touch5 = 0;  
var touch6 = 0;  
var touch7 = 0;  
var touch8 = 0;  
var touch9 = 0;  
var phi = 0;
```

```
var theta2 = 300;  
var thetad = 0;  
var theta2d = 0;  
var L_1 = 0;  
var L_2 = 0;  
var L_3 = 0;  
var L_4 = 0;  
var Alpha3 = 0;
```

```
var AA = 0;  
var BB = 0;  
var CC = 0;
```

```
var DD = 0;

var target_peg = 0;

var theta = 0;
var hold = 0;
var hold2 = 0;

var h_all_int = 0;
var h_all_p = 0;

var y_cor = 0;

var time = 1;
var time_step = 1;
var prev_time = 0;
var delta_time = 0.000;

var next_hole = 0;

var col_0 = 0;
var col_1 = 0;
var col_2 = 0;
var col_3 = 0;
var col_4 = 0;
var col_5 = 0;
var col_6 = 0;
var col_7 = 0;
var col_8 = 0;

var h_0in = 0;
var h_1in = 0;
var h_2in = 0;
var h_3in = 0;
var h_4in = 0;
var h_5in = 0;
var h_6in = 0;
var h_7in = 0;
var h_8in = 0;

var h_0drop = 0;
var h_1drop = 0;
var h_2drop = 0;
var h_3drop = 0;
var h_4drop = 0;
var h_5drop = 0;
var h_6drop = 0;
var h_7drop = 0;
var h_8drop = 0;

var test = 0;

var stif_ratio = 2;
```

```

var stif_ratio2 = 3;
var stif_ratio3 = 3.5;

var stif_ratio4 = 2.5;

var stif_limit = 3;//0.5*tolerance / 3;
var stif_limit2 =3;// 2*tolerance / 3;
var stif_limit3 =3;// 2*tolerance / 3;
var stif_limit4 = 4*tolerance / 3;

var in_hole = -1;

var angle_3 = 0;
var angle_4 = 0;
var angle_6 = 0;

var angle_t1 = 0;
var angle_t2 = 0;
var angle_t3 = 0;

var angle_i1 = 0;
var angle_i2 = 0;
var angle_i3 = 0;

var tth_0, tth_1, tth_2, tth_3, tth_4, tth_5, tth_6, tth_7, tth_8;
var th_all = 0;
var sphere;
var dong, feelgood;
var h_t;
var n = 9; //number of pegs
var nx = 3; //number of holes along the x direction
var nz = 3; //number of holes along the z direction
var v;
var comp;
var h_0outout = 0;
var h_1outout = 0;
var h_2outout = 0;
var h_3outout = 0;
var h_4outout = 0;
var h_5outout = 0;
var h_6outout = 0;
var h_7outout = 0;
var h_8outout = 0;
var r_0 = 11; //rounding up the peg
var ri;

var peak_tol = 4;

var ro_0 = 0;
var ro_1 = 0;
var ro_2 = 0;
var ro_3 = 0;

```



```

var ro_4 = 0;
var ro_5 = 0;
var ro_6 = 0;
var ro_7 = 0;
var ro_8 = 0;

var F_help_x = 0.5;
var F_help_y = 0.5;
var F_help_z = 0.5;

var F_help_mag = 0;

var F_help_x_amount = 0.5;//0.003;
var F_help_y_amount = 0.5;
var F_help_z_amount = 0.5;//0.003;

//dimensional parameters
var Lx,Ly,Lz; //size of board
var board_pos_in=[-40,20,-5];// [-40,20,-20]; //Initial board position (3x1 vector)
var peg_pos_in = [80,38,30]; //initial peg position (3x1 vector)
var hole_dep = 19; //depth of holes
var hole_diam = 6.4; //diameter of hole
var T; //period of refresh rate for timer loop
var peg_height = 38;
var peg_diam = 6.4;
var zeta = 0.4;
var m_peg = 0.01*10^-3;//0.01*10^-3;//mega_grams
var stif_peg=(2000*10^-3);//(4000*10^-3);
var dam_peg=zeta*2*sqrt(stif_peg*m_peg);//*10^-3;

var stif_board = 400*10^-3;
var dam_board =16*10^-3;

var delta_x,delta_y,delta_z;
var board_x, board_y, board_z; //position of board centre
var L_1x, L_2x, L_3z, L_4z, L_5y, L_6y; //levels for collision detection
var L_1x_in, L_2x_in, L_3z_in, L_4z_in, L_5y_in, L_6y_in;
var t_1,t_2;
// Light globals
var Luce1, Luce2;
var PosL1 = [board_pos_in[0]+760,board_pos_in[1]+42,board_pos_in[2]+50]; //default light1 position
var PosL2 = [-90,17,-90]; //default light2 position
//force vectors
var F_x;
var F_y;
var F_z;
//force magnification factors
var F_m_x =1;
var F_m_y =1;
var F_m_z =1;
var rounding = 8;
// peg/hip relative mov.
var dx;
var dy;

```

```
var dz;

var plane_dist_vec = Array(1000000);
var S0_I_mjm_vec = Array(1000000);

var h_h0 = 0;
var h_h1 = 0;
var h_h2 = 0;
var h_h3 = 0;
var h_h4 = 0;
var h_h5 = 0;
var h_h6 = 0;
var h_h7 = 0;
var h_h8 = 0;

var plane_dist_sum0 = 0;
var plane_dist_sum1 = 0;
var plane_dist_sum2 = 0;
var plane_dist_sum3 = 0;
var plane_dist_sum4 = 0;
var plane_dist_sum5 = 0;
var plane_dist_sum6 = 0;
var plane_dist_sum7 = 0;
var plane_dist_sum8 = 0;
var S0_I_mjm_sum = 0;

var plane_dist_av0 = 0;
var plane_dist_av1 = 0;
var plane_dist_av2 = 0;
var plane_dist_av3 = 0;
var plane_dist_av4 = 0;
var plane_dist_av5 = 0;
var plane_dist_av6 = 0;
var plane_dist_av7 = 0;
var plane_dist_av8 = 0;

var plane_dist_sum_temp = 0;

var hole_dis = 30; //distance between holes
var hole_x = Array(n);
var hole_y = Array(n);
var hole_z = Array(n);

var hole_x_in = Array(n);
var hole_y_in = Array(n);
var hole_z_in = Array(n);

var CurrentIndex = 0;
```

```

var hand_bone = Array(26);

var x_mjm_draw = array(10);
var y_mjm_draw = array(10);
var z_mjm_draw = array(10);
var tau_draw = 0.00;
var i_mjm_draw = 0;
var draw_points = 10;

var u,y;

var wood;
var peg_mesh1,peg_mesh2,peg_mesh3,board_mesh, board_mesh2, sphere_mesh;
var h_board_mesh, h_board_mesh2;

var b1x,b2x,b3x,b1y,b2y,b3y,b1z,b2z,b3z;
var b_1,b_2,b_3;
var x_1loc;
var x_2loc;
var x_3loc;
var x_1loc;
var dxi_loc;

var R_safe = 3;
var R_H = 0;

var inst_vel_x = 0;
var inst_vel_y = 0;
var inst_vel_z = 0;

var catch0 = 0;

var cross_13_23;
var r_13;
var r_23;

var norm_dist = 0;
var D_norm_dist = 0;
var vxH, vyH, vzH = 0;
var vxHI_loc, vyHI_loc, vzHI_loc = 0;
var vH_mag = 0;
var plane_dist = 0;
var D_plane_dist = 0;
var delta_tau = 0;

var itavec;
var lamvec;
var Fvec_loc;
var Fvec;

var F_time_x_d, F_time_y_d, F_time_z_d = 0;
var F_plane_x_d, F_plane_y_d, F_plane_z_d = 0;
var F_norm_x_d, F_norm_y_d, F_norm_z_d = 0;

```

```

var F_time_x_k, F_time_y_k, F_time_z_k = 0;
var F_plane_x_k, F_plane_y_k, F_plane_z_k = 0;
var F_norm_x_k, F_norm_y_k, F_norm_z_k = 0;

var fitts_dist = 0;

var xi,xf,yi,yf,mjm_a0,mjm_a1,mjm_a2,mjm_a3,mjm_a4,mjm_a5,mjm_b0,mjm_b1,mjm_b2,mjm_b3,mjm_b4,mjm_b5;
var xHI_loc,yHI_loc,zHI_loc;
var evagoras1;
var theta2_low = 1000;
var FxP = 0;
var FyP = 0;
var FzP = 0;

var next_hole_mem_0 = 0;

var xP = 0;
var yP = 0;
var zP = 0;

// normalized time
var tau;
// fake normalized time for the sake of drawing the poly

var root0 = 0.01;
var root1 = 0;
var root2 = 0;
var root3 = 0;
var root4 = 0;
var root5 = 0;
var root6 = 0;
var root7 = 0;
var root8 = 0;

var hn = 0;
var h_num = 0;
var fn = 0.0;
var dfn = 0.0;

var epsilon = 1;

var x_IHloc, y_IHloc, z_IHloc;
var xP_loc = 0;
var yP_loc = 0;
var zP_loc = 0;
var test117;

var FxP_loc = 0;
var FyP_loc = 0;
var FzP_loc = 0;

```

```

var dx_f = -10;//-50;
var dy_f = 83.33;//10;

var dx_i = 10;//-10;
var dy_i = 66.66;//30;

//maximum array for motion vectors. after that a reset of the vectors takes place
var v_size=300000;
//vectors for position, velocity and acceleration of peg and HIP
var position_x, position_y, position_z;
var velocity_x, velocity_y, velocity_z;
//var acceleration_x, acceleration_y, acceleration_z;
var t_position_x, t_position_y, t_position_z;
var t_velocity_x, t_velocity_y, t_velocity_z;
//var t_acceleration_x, t_acceleration_y, t_acceleration_z;
var temp_peg_pos;
var temp_hip_pos;

//indexes
var i = 100; //variable for initializing hole dimensions vectors
var j;
var l=0;
var h=1;
var f=0;
var c=0; //to be used for space partitioning for vertical planes
var b=0; //for horizontal planes
var a= 0; //for holding or not holding the peg
var m=10;
var z=0;
var g=0;
var frame=0; //for the onframe loop
var angles_vector;

var fpx = 0.00;
var fpy = 0.00;
var fpz = 0.00;
var apx = 0;
var apy = 0;
var apz = 0;

var fbx = 0;
var fby = 0;
var fbz = 0;
var abx = 0;
var aby = 0;
var abz = 0;

var fr_k = 0;//*10^-3;
var mk=5*10^-3;
var mk_h=50*10^-3;

//haptics

```

```

var h_toolView;

var R_c = 5;
var ni_draw = 100;
var x_c = 0;
var z_c1 = 0;
var z_c2 = 0;
var dx_c = 0;
var xf_c = 0;
var xs_c = 0;

// Function declarations
function CameraMoveMouse();
function DrawGrid();
function DrawLines();
function ShowNextHole();

function OnDownload()
{
    EraseCache();
    Output("Downloading ...");
    FileDownload("hapticxvr.zip"); //the file containing the dll's for the haptic interfaces
    FileDownload("board.aam"); //the nine hole peg board
    FileDownload("peg.aam"); //the peg
    FileDownload("dong.wav");
// FileDownload("feelgood.wav");
    FileDownload("sphere.aam");
    FileDownload("hand.aam");
// FileDownload("hand_skin.aam");
    FileDownload("Hands.jpg");
// FileDownload("wood_texture.jpg");

}

function OnInit(params)
{
    EraseCache();

    // Camera
    SetCameraPosition(CamPos); //Set Camera Position
    SetCameraDirection(CamDir); //Set Camera Direction
    //SetTimeStep(1000);
    SetFrameRate(60);

    T=1*10^-3; //seconds

    h_0 = 0;
    h_1 = 0;
    h_2 = 0;
    h_3 = 0;
    h_4 = 0;

```

```

h_5 = 0;
h_6 = 0;
h_7 = 0;
h_8 = 0;
h_t = 0;

test117 = 0;
hn = 0;
// epsilon = 10^4;

angles_vector = vector(500);

angles_vector = [108, 120, 119, 103, 137, 130, 119, 143, 156, 104, 125, 104, 140, 135, 115, 114, 111, 144, 121, 122,
101, 132, 178, 131, 98, 130, 97, 128, 92, 139, 148, 156, 153,
111, 134, 143, 104, 129, 140, 100, 107, 141, 125, 136, 152, 135, 110, 140, 122, 109, 166, 127, 107, 138, 158, 137,
141, 127, 122, 142, 122, 127, 115, 112, 120, 102, 130, 127, 125,
81, 115, 130, 133, 129, 125, 94, 132, 123, 135, 137, 124, 132, 119, 119, 118, 107, 130, 150, 114, 116, 140, 126, 156,
92, 105, 110, 168, 134, 117, 126, 119, 145, 126, 141,
139, 121, 123, 128, 110, 121, 124, 105, 144, 103, 111, 102, 146, 112, 92, 141, 154, 139, 128, 109, 111, 150, 148,
113, 141, 130, 118, 109, 131, 90, 95, 165, 124, 177, 121, 129,
143, 122, 115, 153, 108, 141, 115, 140, 142, 109, 136, 146, 105, 132, 125, 101, 120, 122, 104, 87, 108, 125, 92,
150, 114, 85, 128, 155, 126, 129, 125, 129, 143, 129, 140, 135,
128, 131, 106, 107, 125, 132, 108, 141, 128, 179, 133, 100, 133, 135, 115, 130, 114, 126, 134, 149, 141, 74, 131,
142, 125, 142, 141, 151, 150, 144, 93, 106, 139, 106, 114, 139,
126, 146, 99, 135, 93, 97, 131, 123, 103, 120, 114, 108, 130, 96, 132, 86, 136, 132, 40, 63, 57, 34, 86, 60, 72,
44, 45, 67, 62, 101, 54, 109, 93, 17, 23, 47, 56, 60, 79, 44,
44, 56, 60, 66, 83, 74, 21, 75, 78, 74, 89, 67, 45, 57, 6, 71, 63, 47, 45, 36, 59, 68, 53, 49, 59, 56, 39, 89,
70, 88, 49, 35, 78, 61, 43, 58, 15, 84, 38, 45, 90, 40, 51, 49,
96, 62, 36, 35, 99, 103, 96, 32, 55, 56, 67, 47, 38, 50, 84, 113, 65, 54, 76, 49, 101, 85, 33, 45, 90, 69, 69,
22, 65, 75, 46, 38, 56, 37, 58, 66, 90, 64, 48, 96, 78, 65, 75,
49, 79, 66, 74, 37, 94, 70, 17, 70, 88, 74, 91, 89, 40, 9, 100, 69, 60, 51, 26, 65, 29, 41, 55, 77, 68, 30,
93, 61, 102, 33, 43, 43, 58, 38, 39, 7, 41, 66, 56, 56, 57, 64, 49,
44, 71, 51, 65, 39, 57, 77, 40, 63, 55, 58, 52, 14, 57, 91, 74, 52, 45, 66, 51, 48, 61, 88, 4, 73, 38, 81, 6,
55, 61, 51, 55, 26, 35, 42, 68, 47, 94, 63, 19, 50, 46, 62, 55, 54,
44, 59, 28, 74, 31, 45, 57, 66, 58, 99, 96, 74, 62, 42, 50, 29, 52, 50, 99, 77, 61, 54, 62, 77, 110, 67, 79,
75, 72, 82, 88, 61, 53, 65, 82, 87, 48, 80, 56, 53, 72, 81, 47, 27,
59, 72, 62, 95, 53, 78, 43, 32, 75, 29, 31, 47, 27, 72, 54, 31, 40, 38, 58, 6, 45, 29, 67, 73, 63, 58, 2,
61];

// Declare direction vectors

b_1 = vector(3);
b_2 = vector(3);
b_3 = vector(3);

x_1loc = vector(3);
x_2loc = vector(3);

```

```

x_3loc = vector(3);
x_1loc = vector(3);
dxi_loc = vector(3);

cross_13_23 = vector(3);
r_13 = vector(3);
r_23 = vector(3);

tau = vector(51);

itavec = vector(3);
lamvec = vector(3);
Fvec_loc = vector(3);
Fvec = vector(3);
x_Hloc = vector(3);
plane_dist_vec = vector(100000);
Sm = vector(100000);
Sm_t = vector(100000);

x_mjm_draw = vector(10);
y_mjm_draw = vector(10);
z_mjm_draw = vector(10);

//maximum array size before reset of the values
//peg vectors
position_x = vector(v_size);
velocity_x = vector(v_size);
//acceleration_x = vector(v_size);
position_y = vector(v_size);
velocity_y = vector(v_size);
//acceleration_y = vector(v_size);
position_z = vector(v_size);
velocity_z = vector(v_size);
//acceleration_z = vector(v_size);

//board vectors;
board_x = vector(v_size);
board_y = vector(v_size);
board_z = vector(v_size);

//tool vectors
t_position_x = vector(v_size);
t_velocity_x = vector(v_size);
//t_acceleration_x = vector(v_size);
t_position_y = vector(v_size);
t_velocity_y = vector(v_size);
//t_acceleration_y = vector(v_size);
t_position_z = vector(v_size);
t_velocity_z = vector(v_size);
//t_acceleration_z = vector(v_size);
temp_peg_pos = vector(3);
temp_hip_pos = vector(3);

position_x[0] = peg_pos_in[0];

```



```

position_x[1] = peg_pos_in[0];
position_y[0] = peg_pos_in[1];
position_y[1] = peg_pos_in[1];
position_z[0] = peg_pos_in[2];
position_z[1] = peg_pos_in[2];
t_position_x[0] = 0;
t_position_x[1] = 0;
t_position_y[0] = 0;
t_position_y[1] = 0;
t_position_z[0] = 0;
t_position_z[1] = 0;
board_x[0] = board_pos_in[0];
board_x[1] = board_pos_in[0];
board_y[0] = board_pos_in[1];
board_y[1] = board_pos_in[1];
board_z[0] = board_pos_in[2];
board_z[1] = board_pos_in[2];
//force vectors
F_x = vector(v_size);
F_y = vector(v_size);
F_z = vector(v_size);

//relative mov.
dx = vector(v_size);
dy = vector(v_size);
dz = vector(v_size);

```

```

Lx = 120;
Ly=40;
Lz=120;

```

```

// First Light
Luce1 = NewLight();
Luce1.SetDiffuse(0.5,0.5,0.5);
Luce1.SetSpecular(1.2,1.2,1.2);
Luce1.SetAmbient(5,5,5);
// Luce1.Enable();
Luce1.Disable();

// Second Light
Luce2 = NewLight();

Luce2.SetDiffuse(0.76,0.76,0.76,1);
Luce2.SetSpecular(0.51,0.51,0.51);
Luce2.SetAmbient(0.7,0.7,0.7);
Luce2.Enable();

```

```

//Luce2.Disable();
// Luce2.Enable();

// Lights initial position
Luce1.SetPosition(PosL1);
Luce2.SetPosition(PosL2);

/// Initialize the Haptic Extension
HLoadModule(false);
// Initialize the physics extension
//NLoadModule(false);

/// Create the World
h_world = HWorld(); //this is the root object for all haptic objects

// Load the Meshes

//Peg Mesh

peg_mesh1 = CvmNewMesh("peg.aam");
//peg_mesh1.ModulateMaterials(0.25,0.2,0.1,10); //peg_mesh1.centerBoundingBox();
peg_mesh2 = CvmNewMesh("peg.aam");
//peg_mesh2.ModulateMaterials(0.25,0.2,0.1,10); //peg_mesh1.centerBoundingBox();
peg_mesh3 = CvmNewMesh("peg.aam");
//peg_mesh3.ModulateMaterials(0.25,0.2,0.1,10); //peg_mesh1.centerBoundingBox();

peg_mesh4 = CvmNewMesh("peg.aam");
peg_mesh4.ModulateMaterials(0.9,0.2,0.1,10); //peg_mesh1.centerBoundingBox();

dong = CVmVRAWav("dong.wav");
dong.Load("dong.wav");

//Board Mesh

board_mesh = CvmNewMesh("board.aam");
board_mesh.ModulateMaterials(1,0.8,0.6,1);
board_mesh.centerBoundingBox();

h_board_mesh = HMesh(h_world);
h_board_mesh.loadFromFile("board.aam");
h_world.addChild(h_board_mesh);
h_board_mesh.stiffness = 2;
h_board_mesh.position([board_pos_in[0],board_pos_in[1],board_pos_in[2]]);
//h_board_mesh.mesh.stiffness = 2;

board_mesh2 = CvmNewMesh("board.aam");
board_mesh2.ModulateMaterials(0.6,0.9,0.9,0.5);

```

```

board_mesh2.centerBoundingBox();

//h_board_mesh2 = HGMesh(h_world,board_mesh);
//h_board_mesh2.rotate(180,[1,0,0]);
//h_board_mesh2.setPosition([board_pos_in[0]+120,board_pos_in[1],board_pos_in[2]]);
//h_board_mesh2.mesh.stiffness = 2;

sphere_mesh = CvmNewMesh("sphere.aam");
sphere_mesh.Scale(3,3,3);
sphere_mesh.ModulateMaterials(0.2,0.8,0.6,1);
sphere_mesh.centerBoundingBox();

sphere = CVmObj();
sphere.SetPosition(board_pos_in[0],board_pos_in[1]+100,board_pos_in[2]);
sphere.LinkToMesh(sphere_mesh);
//Create a Board object and link it to board mesh structure
board = CVmObj();
board.SetPosition(board_pos_in[0],board_pos_in[1],board_pos_in[2]);
board.LinkToMesh(board_mesh);

//Create a Board object and link it to board mesh structure
board2 = CVmObj();
board2.Rotate(180,[1,0,0]);
board2.SetPosition(board_pos_in[0]+120,board_pos_in[1],board_pos_in[2]);
board2.LinkToMesh(board_mesh2);

//Create the peg objects and link them to the peg mesh
peg = CVmObj();
peg.Rotate(90,[1,0,0]);
peg.SetPosition(peg_pos_in[0],peg_pos_in[1],peg_pos_in[2]);
peg.LinkToMesh(peg_mesh1);

peg_t = CVmObj();
peg_t.SetScale(1.01,1.01,0.5);
peg_t.Rotate(90,[1,0,0]);
peg_t.SetPosition(0,90,0);
peg_t.LinkToMesh(peg_mesh4);

//Create the peg objects and link them to the peg mesh
peg_0 = CVmObj();
peg_0.Rotate(90,[1,0,0]);
peg_0.SetPosition(board_pos_in[0] + Lx/2 + rand(100), board_pos_in[1] + Ly/2 + peg_height + tolerance + 2
,board_pos_in[2] + rand(50) - rand(50) );
peg_0.LinkToMesh(peg_mesh2);

//Create the peg objects and link them to the peg mesh
peg0 = CVmObj();
peg0.Rotate(90,[1,0,0]);
peg0.SetPosition(peg_pos_in[0],peg_pos_in[1],peg_pos_in[2]);
peg0.LinkToMesh(peg_mesh3);

peg0_t = CVmObj();

```

```

peg0_t.SetScale(1.01,1.01,0.5);
peg0_t.Rotate(90,[1,0,0]);
peg0_t.SetPosition(peg_pos_in[0],peg_pos_in[1],peg_pos_in[2]);
peg0_t.LinkToMesh(peg_mesh4);

peg1 = CVmObj();
peg1.Rotate(90,[1,0,0]);
peg1.SetPosition(peg_pos_in[0],peg_pos_in[1],peg_pos_in[2]);
peg1.LinkToMesh(peg_mesh3);

peg1_t = CVmObj();
peg1_t.SetScale(1.01,1.01,0.5);
peg1_t.Rotate(90,[1,0,0]);
peg1_t.SetPosition(peg_pos_in[0],peg_pos_in[1],peg_pos_in[2]);
peg1_t.LinkToMesh(peg_mesh4);

//Create the peg objects and link them to the peg mesh
peg2 = CVmObj();
peg2.Rotate(90,[1,0,0]);
peg2.SetPosition(peg_pos_in[0],peg_pos_in[1],peg_pos_in[2]);
peg2.LinkToMesh(peg_mesh3);

peg2_t = CVmObj();
peg2_t.SetScale(1.01,1.01,0.5);
peg2_t.Rotate(90,[1,0,0]);
peg2_t.SetPosition(peg_pos_in[0],peg_pos_in[1],peg_pos_in[2]);
peg2_t.LinkToMesh(peg_mesh4);

//Create the peg objects and link them to the peg mesh
peg3 = CVmObj();
peg3.Rotate(90,[1,0,0]);
peg3.SetPosition(peg_pos_in[0],peg_pos_in[1],peg_pos_in[2]);
peg3.LinkToMesh(peg_mesh3);

peg3_t = CVmObj();
peg3_t.SetScale(1.01,1.01,0.5);
peg3_t.Rotate(90,[1,0,0]);
peg3_t.SetPosition(peg_pos_in[0],peg_pos_in[1],peg_pos_in[2]);
peg3_t.LinkToMesh(peg_mesh4);

//Create the peg objects and link them to the peg mesh
peg4 = CVmObj();
peg4.Rotate(90,[1,0,0]);
peg4.SetPosition(peg_pos_in[0],peg_pos_in[1],peg_pos_in[2]);
peg4.LinkToMesh(peg_mesh3);

peg4_t = CVmObj();
peg4_t.SetScale(1.01,1.01,0.5);
peg4_t.Rotate(90,[1,0,0]);
peg4_t.SetPosition(peg_pos_in[0],peg_pos_in[1],peg_pos_in[2]);
peg4_t.LinkToMesh(peg_mesh4);

//Create the peg objects and link them to the peg mesh
peg5 = CVmObj();

```

```

peg5.Rotate(90, [1,0,0]);
peg5.SetPosition(peg_pos_in[0],peg_pos_in[1],peg_pos_in[2]);
peg5.LinkToMesh(peg_mesh3);

peg5_t = CVmObj();
peg5_t.SetScale(1.01,1.01,0.5);
peg5_t.Rotate(90, [1,0,0]);
peg5_t.SetPosition(peg_pos_in[0],peg_pos_in[1],peg_pos_in[2]);
peg5_t.LinkToMesh(peg_mesh4);

//Create the peg objects and link them to the peg mesh
peg6 = CVmObj();
peg6.Rotate(90, [1,0,0]);
peg6.SetPosition(peg_pos_in[0],peg_pos_in[1],peg_pos_in[2]);
peg6.LinkToMesh(peg_mesh3);

peg6_t = CVmObj();
peg6_t.SetScale(1.01,1.01,0.5);
peg6_t.Rotate(90, [1,0,0]);
peg6_t.SetPosition(peg_pos_in[0],peg_pos_in[1],peg_pos_in[2]);
peg6_t.LinkToMesh(peg_mesh4);

//Create the peg objects and link them to the peg mesh
peg7 = CVmObj();
peg7.Rotate(90, [1,0,0]);
peg7.SetPosition(peg_pos_in[0],peg_pos_in[1],peg_pos_in[2]);
peg7.LinkToMesh(peg_mesh3);

peg7_t = CVmObj();
peg7_t.SetScale(1.01,1.01,0.5);
peg7_t.Rotate(90, [1,0,0]);
peg7_t.SetPosition(peg_pos_in[0],peg_pos_in[1],peg_pos_in[2]);
peg7_t.LinkToMesh(peg_mesh4);

peg8 = CVmObj();
peg8.Rotate(90, [1,0,0]);
peg8.SetPosition(peg_pos_in[0],peg_pos_in[1],peg_pos_in[2]);
peg8.LinkToMesh(peg_mesh3);

peg8_t = CVmObj();
peg8_t.SetScale(1.01,1.01,0.5);
peg8_t.Rotate(90, [1,0,0]);
peg8_t.SetPosition(peg_pos_in[0],peg_pos_in[1],peg_pos_in[2]);
peg8_t.LinkToMesh(peg_mesh4);

hand_avatar = CvmAvatar("hand.aam");
hand_avatar.SetPosition(0,60,0);
hand_avatar.SetRotation(-90,1,0,0);
hand_avatar.SetScale(7,7,7);

```

```
th_0 = 0;
```

```

th_1 = 0;
th_2 = 0;
th_3 = 0;
th_4 = 0;
th_5 = 0;
th_6 = 0;
th_7 = 0;
th_8 = 0;

tth_0 = 0;
tth_1 = 0;
tth_2 = 0;
tth_3 = 0;
tth_4 = 0;
tth_5 = 0;
tth_6 = 0;
tth_7 = 0;
tth_8 = 0;

h_t=0;
h_all = 0;

//create the HIP

h_toolView = HToolGL(h_world,5);
{
    var tool = h_toolView.tool;
//    tool.setParameter("workspace", [83,83,83]);
    tool.setParameter("workspace", [83,83,83]);
    tool.updatePose();

    tool.start();
    tool.computeForces();

    tool.applyForces();
    tool.forceEnabled = true;
}

h_world.computeGlobalPositions(true);

h_hapticDisplay = true;

h_timerMode=true;

if(!h_timerMode)
    h_world.run();
}

var hand_rot = 0;
function OnFrame()
{

```

```

// Camera manager
CameraMoveMouse();

peg_mesh1.ModulateMaterials(1,1,1,10);
peg_mesh2.ModulateMaterials(1,1,1,10);

h_0outout = 0;

SceneBegin();

if (hand_rot == 0)
{
hand_avatar.Rotate(90,0,0,1);
}

hand_rot = hand_rot + 1;

// Draws grid
//DrawGrid();
DrawLines();
ShowNextHole();
//peg_0.draw();

// TO DO
// Luce1.SetPosition(board_x[h]-760,board_y[h]+42,board_z[h]-50);

peg.SetPosition(position_x[h],position_y[h],position_z[h]);
board.Draw();
board2.Draw();
// peg.Draw();
// sphere.Draw();
// peg_0.Draw();
// peg_t.Draw();
// peg0.Draw();

//peg_0.SetPosition(board_pos_in[0]+100 + apx*sin(fpx*h),board_pos_in[1] + Ly/2 + 19 + apy*sin(fpy*h),board_pos_in[2]
+ apz*sin(fpz*h));

if(abs(peg.GetPosition[0]-peg_0.GetPosition[0]) =<plane_tolerance && abs(peg.GetPosition[1]-peg_0.GetPosition[1] +
tolerance/2) <= tolerance/2 && abs(peg.GetPosition[2]-peg_0.GetPosition[2]) =<plane_tolerance)
{
// peg_mesh1.ModulateMaterials(0.25,0.2,0.1,1);

```

```

// peg_mesh2.ModulateMaterials(0.25,0.2,0.1,1);
  h_t = 1;
}

// else
// peg_mesh1.ModulateMaterials(0.45,0.3,0.2,0.5);

// if(h_t == 1)
// peg_0.SetPosition(board_pos_in[0] + rand(0,100) + apx*sin(fpx*h),board_pos_in[1] + Ly/2 + 19 +
apy*sin(fpy*h),board_pos_in[2] + rand(0,50) + apz*sin(fpz*h));

// else
// peg_mesh2.ModulateMaterials(0.45,0.3,0.2,0.5);

  for(var g=40; g<48; g+=1)
  {

    if (c==g && b==11 && (position_y[h]-peg_height)<=(board_y[h]+Ly/2-19))

    {
      // peg_mesh1.ModulateMaterials(0.25,0.2,0.1,1);
    }

  }

peg0.SetPosition(hole_x[0],hole_y[0]+19,hole_z[0]);
peg0_t.SetPosition(hole_x[0],hole_y[0]+19.1,hole_z[0]);
if (c==40 && b==11 && (position_y[h]-peg_height)<=(board_y[h]+Ly/2-19))
{
  h_0 = 1;
  th_0 = 1;
  h_t = 0;

  peg_0.SetPosition(board_pos_in[0] + Lx/2 + rand(100) + apx*sin(fpx*h), board_pos_in[1] + Ly/2 + peg_height + tolerance +
2 + apy*sin(fpy*h), board_pos_in[2] + rand(50) - rand(50) + apz*sin(fpz*h));
  if (peg_0.GetPosition[2] == peg0.GetPosition[2])
  {
    peg_0.GetPosition[2] = peg_0.GetPosition[2] + 4;
  }
}

peg1.SetPosition(hole_x[1],hole_y[1]+19,hole_z[1]);
peg1_t.SetPosition(hole_x[1],hole_y[1]+19.1,hole_z[1]);

if (c==41 && b==11 && (position_y[h]-peg_height)<=(board_y[h]+Ly/2-19))
{
  h_1 = 1;
  th_1 = 1;
  h_t = 0;
}

```



```

peg_0.SetPosition(board_pos_in[0] + Lx/2 + rand(100) + apx*sin(fpx*h),board_pos_in[1] + Ly/2 + peg_height + tolerance +
2 + apy*sin(fpy*h),board_pos_in[2] + rand(50) - rand(50) + apz*sin(fpz*h));
if (peg_0.GetPosition[2] == peg0.GetPosition[2])
{
peg_0.GetPosition[2] = peg_0.GetPosition[2] + 4;
}
}

peg2.SetPosition(hole_x[2],hole_y[2]+19,hole_z[2]);
peg2_t.SetPosition(hole_x[2],hole_y[2]+19.1,hole_z[2]);
if (c==42 && b==11 && (position_y[h]-peg_height)<=(board_y[h]+Ly/2-19))
{

h_2 = 1;
th_2 = 1;
h_t = 0;
peg_0.SetPosition(board_pos_in[0] + Lx/2 + rand(100) + apx*sin(fpx*h),board_pos_in[1] + Ly/2 + peg_height + tolerance +
2 + apy*sin(fpy*h),board_pos_in[2] + rand(50) - rand(50) + apz*sin(fpz*h));
if (peg_0.GetPosition[2] == peg0.GetPosition[2])
{
peg_0.GetPosition[2] = peg_0.GetPosition[2] + 4;
}
}
peg3.SetPosition(hole_x[3],hole_y[3]+19,hole_z[3]);
peg3_t.SetPosition(hole_x[3],hole_y[3]+19.1,hole_z[3]);
if (c==43 && b==11 && (position_y[h]-peg_height)<=(board_y[h]+Ly/2-19))
{

h_3 = 1;
th_3 = 1;
h_t = 0;
peg_0.SetPosition(board_pos_in[0] + Lx/2 + rand(100) + apx*sin(fpx*h),board_pos_in[1] + Ly/2 + peg_height + tolerance +
2 + apy*sin(fpy*h),board_pos_in[2] + rand(50) - rand(50) + apz*sin(fpz*h));
if (peg_0.GetPosition[2] == peg0.GetPosition[2])
{
peg_0.GetPosition[2] = peg_0.GetPosition[2] + 4;
}
}
peg4.SetPosition(hole_x[4],hole_y[4]+19,hole_z[4]);
peg4_t.SetPosition(hole_x[4],hole_y[4]+19.1,hole_z[4]);
if (c==44 && b==11 && (position_y[h]-peg_height)<=(board_y[h]+Ly/2-19))
{

h_4 = 1;
th_4 = 1;
h_t = 0;
peg_0.SetPosition(board_pos_in[0] + Lx/2 + rand(100) + apx*sin(fpx*h),board_pos_in[1] + Ly/2 + peg_height + tolerance +
2 + apy*sin(fpy*h),board_pos_in[2] + rand(50) - rand(50) + apz*sin(fpz*h));
if (peg_0.GetPosition[2] == peg0.GetPosition[2])
{
peg_0.GetPosition[2] = peg_0.GetPosition[2] + 4;
}
}
peg5.SetPosition(hole_x[5],hole_y[5]+19,hole_z[5]);

```

```

peg5_t.SetPosition(hole_x[5],hole_y[5]+19.1,hole_z[5]);
if (c==45 && b==11 && (position_y[h]-peg_height)<=(board_y[h]+Ly/2-19))
{
h_5 = 1;
th_5 = 1;
h_t = 0;
peg_0.SetPosition(board_pos_in[0] + Lx/2 + rand(100) + apx*sin(fpz*h),board_pos_in[1] + Ly/2 + peg_height + tolerance +
2 + apy*sin(fpy*h),board_pos_in[2] + rand(50) - rand(50) + apz*sin(fpz*h));
if (peg_0.GetPosition[2] == peg0.GetPosition[2])
{
peg_0.GetPosition[2] = peg_0.GetPosition[2] + 4;
}
}
peg6.SetPosition(hole_x[6],hole_y[6]+19,hole_z[6]);
peg6_t.SetPosition(hole_x[6],hole_y[6]+19.1,hole_z[6]);

if (c==46 && b==11 && (position_y[h]-peg_height)<=(board_y[h]+Ly/2-19))
{
h_6 = 1;
th_6 = 1;
h_t = 0;
peg_0.SetPosition(board_pos_in[0] + Lx/2 + rand(100) + apx*sin(fpz*h),board_pos_in[1] + Ly/2 + peg_height + tolerance +
2 + apy*sin(fpy*h),board_pos_in[2] + rand(50) - rand(50) + apz*sin(fpz*h));
if (peg_0.GetPosition[2] == peg0.GetPosition[2])
{
peg_0.GetPosition[2] = peg_0.GetPosition[2] + 4;
}
}
peg7.SetPosition(hole_x[7],hole_y[7]+19,hole_z[7]);
peg7_t.SetPosition(hole_x[7],hole_y[7]+19.1,hole_z[7]);

if (c==47 && b==11 && (position_y[h]-peg_height)<=(board_y[h]+Ly/2-19))
{
h_7 = 1;
th_7 = 1;
h_t = 0;
peg_0.SetPosition(board_pos_in[0] + Lx/2 + rand(100) + apx*sin(fpz*h),board_pos_in[1] + Ly/2 + peg_height + tolerance +
2 + apy*sin(fpy*h),board_pos_in[2] + rand(50) - rand(50) + apz*sin(fpz*h));
if (peg_0.GetPosition[2] == peg0.GetPosition[2])
{
peg_0.GetPosition[2] = peg_0.GetPosition[2] + 4;
}
}

peg8.SetPosition(hole_x[8],hole_y[8]+19,hole_z[8]);
peg8_t.SetPosition(hole_x[8],hole_y[8]+19.1,hole_z[8]);

if (c==48 && b==11 && (position_y[h]-peg_height)<=(board_y[h]+Ly/2-19))
{
h_8 = 1;
th_8 = 1;

```

```

    h_t = 0;
    peg_0.SetPosition(board_pos_in[0] + Lx/2 + rand(100) + apx*sin(fpz*h),board_pos_in[1] + Ly/2 + peg_height + tolerance +
2 + apy*sin(fpy*h),board_pos_in[2] + rand(50) - rand(50) + apz*sin(fpz*h));
    if (peg_0.GetPosition[2] == peg0.GetPosition[2])
    {
        peg_0.GetPosition[2] = peg_0.GetPosition[2] + 4;
    }
}

//FORCE VECTOR REPRESENTATION
h_toolView.render();
//sphere.SetPosition(h_toolView.tool.proxyPosition[0],t_position_y[h],h_toolView.tool.proxyPosition[2]);
sphere.SetPosition(t_position_x[h],t_position_y[h],t_position_z[h]);
if (abs(t_position_x[h] - peg_0.GetPosition[0]) <= 15 && abs(t_position_y[h] - peg_0.GetPosition[1]) <= 15 &&
abs(t_position_z[h] - peg_0.GetPosition[2]) <= 15 && h_all != 1)
{
    sphere.Draw();
}
sphere.Draw();
if (h_all == 1 && h_0outout != 1 && h_0drop != 1 && abs(t_position_x[h] - hole_x[0]) <= 15 && abs(t_position_y[h] -
hole_y[0] - 19) <= 15 && abs(t_position_z[h] - hole_z[0]) <= 15)
{
    sphere.Draw();
}

if (h_all == 1 && h_1outout != 1 && h_1drop != 1 && abs(t_position_x[h] - hole_x[1]) <= 15 && abs(t_position_y[h] -
hole_y[1] - 19) <= 15 && abs(t_position_z[h] - hole_z[1]) <= 15)
{
    sphere.Draw();
}

if (h_all == 1 && h_2outout != 1 && h_2drop != 1 && abs(t_position_x[h] - hole_x[2]) <= 15 && abs(t_position_y[h] -
hole_y[2] - 19) <= 15 && abs(t_position_z[h] - hole_z[2]) <= 15)
{
    sphere.Draw();
}

if (h_all == 1 && h_3outout != 1 && h_3drop != 1 && abs(t_position_x[h] - hole_x[0]) <= 15 && abs(t_position_y[h] -
hole_y[0] - 19) <= 15 && abs(t_position_z[h] - hole_z[3]) <= 15)
{
    sphere.Draw();
}

if (h_all == 1 && h_4outout != 1 && h_4drop != 1 && abs(t_position_x[h] - hole_x[4]) <= 15 && abs(t_position_y[h] -
hole_y[4] - 19) <= 15 && abs(t_position_z[h] - hole_z[4]) <= 15)
{
    sphere.Draw();
}

if (h_all == 1 && h_5outout != 1 && h_5drop != 1 && abs(t_position_x[h] - hole_x[5]) <= 15 && abs(t_position_y[h] -
hole_y[5] - 19) <= 15 && abs(t_position_z[h] - hole_z[5]) <= 15)
{
    sphere.Draw();
}

```

```

    if (h_all == 1 && h_6outout != 1 && h_6drop != 1 && abs(t_position_x[h] - hole_x[6]) <= 15 && abs(t_position_y[h] -
hole_y[6] - 19) <= 15 && abs(t_position_z[h] - hole_z[6]) <= 15)
    {
        sphere.Draw();
    }

    if (h_all == 1 && h_7outout != 1 && h_7drop != 1 && abs(t_position_x[h] - hole_x[7]) <= 15 && abs(t_position_y[h] -
hole_y[7] - 19) <= 15 && abs(t_position_z[h] - hole_z[7]) <= 15)
    {
        sphere.Draw();
    }

    if (h_all == 1 && h_8outout != 1 && h_8drop != 1 && abs(t_position_x[h] - hole_x[8]) <= 15 && abs(t_position_y[h] -
hole_y[8] - 19) <= 15 && abs(t_position_z[h] - hole_z[8]) <= 15)
    {
        sphere.Draw();
    }

    //if(h_t!=1)
    //peg_0.ModulateMaterials(0.6,0.7,0.9,100);

    // if(h_t==1)
    // peg_0.SetPosition(board_pos_in[0]+rand(0,100),board_pos_in[1]+100,board_pos_in[2]+rand(0,50));
    if (h_0 == 1 && h_1 == 1 && h_2 == 1 && h_3 == 1 && h_4 == 1 && h_5 == 1 && h_6 == 1 && h_7 == 1 && h_8 == 1 &&
t_position_x[h] >= 30)
        h_all_int = 1;

    if (h_0 == 1 && h_1 == 1 && h_2 == 1 && h_3 == 1 && h_4 == 1 && h_5 == 1 && h_6 == 1 && h_7 == 1 && h_8 == 1 &&
h_all_int == 1)
        h_all = 1;

    if (h_0 == 1 && h_1 == 1 && h_2 == 1 && h_3 == 1 && h_4 == 1 && h_5 == 1 && h_6 == 1 && h_7 == 1 && h_8 == 1)
        h_all_p = 1;

    if (h_all == 1)
        h_t = 0;

    if(abs(t_position_x[h]-peg0.GetPosition[0]) =< plane_tolerance && abs(t_position_y[h]-peg0.GetPosition[1] +
tolerance/2) <= tolerance/2 && abs(t_position_z[h]-peg0.GetPosition[2]) =<plane_tolerance && h_all == 1 && h_0drop != 1 &&
h_0outout != 1)
    {
        h_0out = 1; //grabs peg 0 for pulling it out
        hole_diameter2 = plane_tolerance*2;// + tolerance*1.5;
        // f_m_x = 1 - (abs(t_position_x[h] - hole_x[0])/(hole_diameter/2))^2;
        // f_m_z = 1 - (abs(t_position_z[h] - hole_z[0])/(hole_diameter/2))^2;
    }

    if(h_0out == 1 && h_0drop != 1 && h_0outout != 1 && abs(t_position_x[h] - position_x[h]) < 2 && abs(t_position_y[h] -
position_y[h]) < 2 && abs(t_position_z[h] - position_z[h]) < 2)

```

```

{
h_0in = 1; //grabs peg 0 for pulling it out
hole_diameter2 = plane_tolerance*2;// + tolerance*1.5;
// f_m_x = 1 - (abs(t_position_x[h] - hole_x[0])/(hole_diameter/2))^2;
// f_m_z = 1 - (abs(t_position_z[h] - hole_z[0])/(hole_diameter/2))^2;
}

if(abs(t_position_x[h]-peg1.GetPosition[0]) =< plane_tolerance && abs(t_position_y[h]-peg1.GetPosition[1] +
tolerance/2) <= tolerance/2 && abs(t_position_z[h]-peg1.GetPosition[2]) =<plane_tolerance && h_all == 1 && h_1drop != 1 &&
h_loutout != 1)
{
h_lout = 1; //grabs peg 0 for pulling it out
hole_diameter2 = plane_tolerance*2;// + tolerance*1.5;
// f_m_x = 1 - (abs(t_position_x[h] - hole_x[1])/(hole_diameter/2))^2;
// f_m_z = 1 - (abs(t_position_z[h] - hole_z[1])/(hole_diameter/2))^2;
}

if(h_lout == 1 && h_1drop != 1 && h_loutout != 1 && abs(t_position_x[h] - position_x[h]) < 2 && abs(t_position_y[h] -
position_y[h]) < 2 && abs(t_position_z[h] - position_z[h]) < 2)
{
h_lin = 1; //grabs peg 0 for pulling it out
hole_diameter2 = plane_tolerance*2;// + tolerance*1.5;
// f_m_x = 1 - (abs(t_position_x[h] - hole_x[0])/(hole_diameter/2))^2;
// f_m_z = 1 - (abs(t_position_z[h] - hole_z[0])/(hole_diameter/2))^2;
}

if(abs(t_position_x[h]-peg2.GetPosition[0]) =< plane_tolerance && abs(t_position_y[h]-peg2.GetPosition[1] +
tolerance/2) <= tolerance/2 && abs(t_position_z[h]-peg2.GetPosition[2]) =<plane_tolerance && h_all == 1 && h_2drop != 1)
{
h_2out = 1; //grabs peg 0 for pulling it out
hole_diameter2 = plane_tolerance*2;// + tolerance*1.5;
// f_m_x = 1 - (abs(t_position_x[h] - hole_x[2])/(hole_diameter/2))^2;
// f_m_z = 1 - (abs(t_position_z[h] - hole_z[2])/(hole_diameter/2))^2;
}

if(h_2out == 1 && h_2drop != 1 && h_2outout != 1 && abs(t_position_x[h] - position_x[h]) < 2 && abs(t_position_y[h] -
position_y[h]) < 2 && abs(t_position_z[h] - position_z[h]) < 2)
{
h_2in = 1; //grabs peg 0 for pulling it out
hole_diameter2 = plane_tolerance*2;// + tolerance*1.5;
// f_m_x = 1 - (abs(t_position_x[h] - hole_x[0])/(hole_diameter/2))^2;
// f_m_z = 1 - (abs(t_position_z[h] - hole_z[0])/(hole_diameter/2))^2;
}

if(abs(t_position_x[h]-peg3.GetPosition[0]) =< plane_tolerance && abs(t_position_y[h]-peg3.GetPosition[1] +
tolerance/2) <= tolerance/2 && abs(t_position_z[h]-peg3.GetPosition[2]) =<plane_tolerance && h_all == 1 && h_3drop != 1)
{
h_3out = 1; //grabs peg 0 for pulling it out
hole_diameter2 = plane_tolerance*2;// + tolerance*1.5;
// f_m_x = 1 - (abs(t_position_x[h] - hole_x[3])/(hole_diameter/2))^2;
// f_m_z = 1 - (abs(t_position_z[h] - hole_z[3])/(hole_diameter/2))^2;
}

```

```

    if(h_3out == 1 && h_3drop != 1 && h_3outout != 1 && abs(t_position_x[h] - position_x[h]) < 2 && abs(t_position_y[h] -
position_y[h]) < 2 && abs(t_position_z[h] - position_z[h]) < 2)
    {
        h_3in = 1; //grabs peg 0 for pulling it out
        hole_diameter2 = plane_tolerance*2;// + tolerance*1.5;
        // f_m_x = 1 - (abs(t_position_x[h] - hole_x[0])/(hole_diameter/2))^2;
        // f_m_z = 1 - (abs(t_position_z[h] - hole_z[0])/(hole_diameter/2))^2;
    }

    if(abs(t_position_x[h]-peg4.GetPosition[0]) =< plane_tolerance && abs(t_position_y[h]-peg4.GetPosition[1] +
tolerance/2) <= tolerance/2 && abs(t_position_z[h]-peg4.GetPosition[2]) =<plane_tolerance && h_all == 1 && h_4drop != 1)
    {
        h_4out = 1; //grabs peg 0 for pulling it out
        hole_diameter2 = plane_tolerance*2;// + tolerance*1.5;
        // f_m_x = 1 - (abs(t_position_x[h] - hole_x[4])/(hole_diameter/2))^2;
        // f_m_z = 1 - (abs(t_position_z[h] - hole_z[4])/(hole_diameter/2))^2;
    }

    if(h_4out == 1 && h_4drop != 1 && h_4outout != 1 && abs(t_position_x[h] - position_x[h]) < 2 && abs(t_position_y[h] -
position_y[h]) < 2 && abs(t_position_z[h] - position_z[h]) < 2)
    {
        h_4in = 1; //grabs peg 0 for pulling it out
        hole_diameter2 = plane_tolerance*2;// + tolerance*1.5;
        // f_m_x = 1 - (abs(t_position_x[h] - hole_x[0])/(hole_diameter/2))^2;
        // f_m_z = 1 - (abs(t_position_z[h] - hole_z[0])/(hole_diameter/2))^2;
    }

    if(abs(t_position_x[h]-peg5.GetPosition[0]) =< plane_tolerance && abs(t_position_y[h]-peg5.GetPosition[1] +
tolerance/2) <= tolerance/2 && abs(t_position_z[h]-peg5.GetPosition[2]) =<plane_tolerance && h_all == 1 && h_5drop != 1)
    {
        h_5out = 1; //grabs peg 0 for pulling it out
        hole_diameter2 = plane_tolerance*2;// + tolerance*1.5;
        // f_m_x = 1 - (abs(t_position_x[h] - hole_x[5])/(hole_diameter/2))^2;
        // f_m_z = 1 - (abs(t_position_z[h] - hole_z[5])/(hole_diameter/2))^2;
    }

    if(h_5out == 1 && h_5drop != 1 && h_5outout != 1 && abs(t_position_x[h] - position_x[h]) < 2 && abs(t_position_y[h] -
position_y[h]) < 2 && abs(t_position_z[h] - position_z[h]) < 2)
    {
        h_5in = 1; //grabs peg 0 for pulling it out
        hole_diameter2 = plane_tolerance*2;// + tolerance*1.5;
        // f_m_x = 1 - (abs(t_position_x[h] - hole_x[0])/(hole_diameter/2))^2;
        // f_m_z = 1 - (abs(t_position_z[h] - hole_z[0])/(hole_diameter/2))^2;
    }

    if(abs(t_position_x[h]-peg6.GetPosition[0]) =< plane_tolerance && abs(t_position_y[h]-peg6.GetPosition[1] +
tolerance/2) <= tolerance/2 && abs(t_position_z[h]-peg6.GetPosition[2]) =<plane_tolerance && h_all == 1 && h_6drop != 1)
    {
        h_6out = 1; //grabs peg 0 for pulling it out
        hole_diameter2 = plane_tolerance*2;// + tolerance*1.5;
        // f_m_x = 1 - (abs(t_position_x[h] - hole_x[6])/(hole_diameter/2))^2;
        // f_m_z = 1 - (abs(t_position_z[h] - hole_z[6])/(hole_diameter/2))^2;
    }

```

```

    if(h_6out == 1 && h_6drop != 1 && h_6outout != 1 && abs(t_position_x[h] - position_x[h]) < 2 && abs(t_position_y[h] -
position_y[h]) < 2 && abs(t_position_z[h] - position_z[h]) < 2)
    {
        h_6in = 1; //grabs peg 0 for pulling it out
        hole_diameter2 = plane_tolerance*2;// + tolerance*1.5;
// f_m_x = 1 - (abs(t_position_x[h] - hole_x[0])/(hole_diameter/2))^2;
// f_m_z = 1 - (abs(t_position_z[h] - hole_z[0])/(hole_diameter/2))^2;
    }

    if(abs(t_position_x[h]-peg7.GetPosition[0]) =< plane_tolerance && abs(t_position_y[h]-peg7.GetPosition[1] +
tolerance/2) <= tolerance/2 && abs(t_position_z[h]-peg7.GetPosition[2]) =<plane_tolerance && h_all == 1 && h_7drop != 1)
    {
        h_7out = 1; //grabs peg 0 for pulling it out
        hole_diameter2 = plane_tolerance*2;// + tolerance*1.5;
// f_m_x = 1 - (abs(t_position_x[h] - hole_x[7])/(hole_diameter/2))^2;
// f_m_z = 1 - (abs(t_position_z[h] - hole_z[7])/(hole_diameter/2))^2;
    }

    if(h_7out == 1 && h_7drop != 1 && h_7outout != 1 && abs(t_position_x[h] - position_x[h]) < 2 && abs(t_position_y[h] -
position_y[h]) < 2 && abs(t_position_z[h] - position_z[h]) < 2)
    {
        h_7in = 1; //grabs peg 0 for pulling it out
        hole_diameter2 = plane_tolerance*2;// + tolerance*1.5;
// f_m_x = 1 - (abs(t_position_x[h] - hole_x[0])/(hole_diameter/2))^2;
// f_m_z = 1 - (abs(t_position_z[h] - hole_z[0])/(hole_diameter/2))^2;
    }

    if(abs(t_position_x[h]-peg8.GetPosition[0]) =< plane_tolerance && abs(t_position_y[h]-peg8.GetPosition[1] +
tolerance/2) <= tolerance/2 && abs(t_position_z[h]-peg8.GetPosition[2]) =<plane_tolerance && h_all == 1 && h_8drop != 1)
    {
        h_8out = 1; //grabs peg 0 for pulling it out
        hole_diameter2 = plane_tolerance*2;// + tolerance*1.5;
// f_m_x = 1 - (abs(t_position_x[h] - hole_x[8])/(hole_diameter/2))^2;
//f_m_z = 1 - (abs(t_position_z[h] - hole_z[8])/(hole_diameter/2))^2;
    }

    if(h_8out == 1 && h_8drop != 1 && h_8outout != 1 && abs(t_position_x[h] - position_x[h]) < 2 && abs(t_position_y[h] -
position_y[h]) < 2 && abs(t_position_z[h] - position_z[h]) < 2)
    {
        h_8in = 1; //grabs peg 0 for pulling it out
        hole_diameter2 = plane_tolerance*2;// + tolerance*1.5;
// f_m_x = 1 - (abs(t_position_x[h] - hole_x[0])/(hole_diameter/2))^2;
// f_m_z = 1 - (abs(t_position_z[h] - hole_z[0])/(hole_diameter/2))^2;
    }

    if(h_all == 1 && h_0out == 1 && (position_y[h] - peg_height - 38) > 0)
    {
        h_0outout = 1; // peg 0 is out
// hole_diameter = hole_diameter2;//6.4 + tolerance*1.5;
        h_0in = 0;
    }

    if(h_all == 1 && h_1out == 1 && (position_y[h] - peg_height - 38) > 0)
    {

```

```

h_1outout = 1; // peg 0 is out
// hole_diameter = hole_diameter2;//6.4 + tolerance*1.5;
h_1in = 0;
}

if(h_all == 1 && h_2out == 1 && (position_y[h] - peg_height - 38) > 0)
{
h_2outout = 1; // peg 0 is out
// hole_diameter = hole_diameter2;//6.4 + tolerance*1.5;
h_2in = 0;
}

if(h_all == 1 && h_3out == 1 && (position_y[h] - peg_height - 38) > 0)
{
h_3outout = 1; // peg 0 is out
// hole_diameter = hole_diameter2;//6.4 + tolerance*1.5;
h_3in = 0;
}

if(h_all == 1 && h_4out == 1 && (position_y[h] - peg_height - 38) > 0)
{
h_4outout = 1; // peg 0 is out
// hole_diameter = hole_diameter2;//6.4 + tolerance*1.5;
h_4in = 0;
}

if(h_all == 1 && h_5out == 1 && (position_y[h] - peg_height - 38) > 0)
{
h_5outout = 1; // peg 0 is out
// hole_diameter = hole_diameter2;//6.4 + tolerance*1.5;
h_5in = 0;
}

if(h_all == 1 && h_6out == 1 && (position_y[h] - peg_height - 38) > 0)
{
h_6outout = 1; // peg 0 is out
// hole_diameter = hole_diameter2;//6.4 + tolerance*1.5;
h_6in = 0;
}

if(h_all == 1 && h_7out == 1 && (position_y[h] - peg_height - 38) > 0)
{
h_7outout = 1; // peg 0 is out
// hole_diameter = hole_diameter2;//6.4 + tolerance*1.5;
h_7in = 0;
}

if(h_all == 1 && h_8out == 1 && (position_y[h] - peg_height - 38) > 0)
{
h_8outout = 1; // peg 0 is out
// hole_diameter = hole_diameter2;//6.4 + tolerance*1.5;
h_8in = 0;
}

```



```

if (h_all == 1 && h_0out == 1 && h_0outout == 1 && t_position_x[h] >= 46)
{
h_0drop = 1;
h_0out = 0;
h_0outout = 0;
}
if (h_all == 1 && h_1out == 1 && h_1outout == 1 && t_position_x[h] >= 46)
{
h_1drop = 1;
h_1out = 0;
h_1outout = 0;
}
if (h_all == 1 && h_2out == 1 && h_2outout == 1 && t_position_x[h] >= 46)
{
h_2drop = 1;
h_2out = 0;
h_2outout = 0;
}
if (h_all == 1 && h_3out == 1 && h_3outout == 1 && t_position_x[h] >= 46)
{
h_3drop = 1;
h_3out = 0;
h_3outout = 0;
}
if (h_all == 1 && h_4out == 1 && h_4outout == 1 && t_position_x[h] >= 46)
{
h_4drop = 1;
h_4out = 0;
h_4outout = 0;
}
if (h_all == 1 && h_5out == 1 && h_5outout == 1 && t_position_x[h] >= 46)
{
h_5drop = 1;
h_5out = 0;
h_5outout = 0;
}
if (h_all == 1 && h_6out == 1 && h_6outout == 1 && t_position_x[h] >= 46)
{
h_6drop = 1;
h_6out = 0;
h_6outout = 0;
}
if (h_all == 1 && h_7out == 1 && h_7outout == 1 && t_position_x[h] >= 46)
{
h_7drop = 1;
h_7out = 0;
h_7outout = 0;
}
if (h_all == 1 && h_8out == 1 && h_8outout == 1 && t_position_x[h] >= 46)
{
h_8drop = 1;

```

```

    h_8out = 0;
    h_8outout = 0;
}

if((h_0outout == 1 || h_1outout == 1 || h_2outout == 1 || h_3outout == 1 || h_4outout == 1 || h_5outout == 1 ||
h_6outout == 1 || h_7outout == 1 || h_8outout == 1))
{
    hole_diameter2 = 0;
}

if (t_position_x[h] > (hole_x[0] - 15) && t_position_x[h] < (hole_x[0] + 15) && t_position_z[h] > (hole_z[0] - 15) &&
t_position_z[h] < (hole_z[0] + 15))
{
    in_hole = 0;
}
if (t_position_x[h] > (hole_x[1] - 15) && t_position_x[h] < (hole_x[1] + 15) && t_position_z[h] > (hole_z[1] - 15) &&
t_position_z[h] < (hole_z[1] + 15))
{
    in_hole = 1;
}
if (t_position_x[h] > (hole_x[2] - 15) && t_position_x[h] < (hole_x[2] + 15) && t_position_z[h] > (hole_z[2] - 15) &&
t_position_z[h] < (hole_z[2] + 15))
{
    in_hole = 2;
}
if (t_position_x[h] > (hole_x[3] - 15) && t_position_x[h] < (hole_x[3] + 15) && t_position_z[h] > (hole_z[3] - 15) &&
t_position_z[h] < (hole_z[3] + 15))
{
    in_hole = 3;
}
if (t_position_x[h] > (hole_x[4] - 15) && t_position_x[h] < (hole_x[4] + 15) && t_position_z[h] > (hole_z[4] - 15) &&
t_position_z[h] < (hole_z[4] + 15))
{
    in_hole = 4;
}
if (t_position_x[h] > (hole_x[5] - 15) && t_position_x[h] < (hole_x[5] + 15) && t_position_z[h] > (hole_z[5] - 15) &&
t_position_z[h] < (hole_z[5] + 15))
{
    in_hole = 5;
}
if (t_position_x[h] > (hole_x[6] - 15) && t_position_x[h] < (hole_x[6] + 15) && t_position_z[h] > (hole_z[6] - 15) &&
t_position_z[h] < (hole_z[6] + 15))
{
    in_hole = 6;
}
if (t_position_x[h] > (hole_x[7] - 15) && t_position_x[h] < (hole_x[7] + 15) && t_position_z[h] > (hole_z[7] - 15) &&
t_position_z[h] < (hole_z[7] + 15))
{
    in_hole = 7;
}
if (t_position_x[h] > (hole_x[8] - 15) && t_position_x[h] < (hole_x[8] + 15) && t_position_z[h] > (hole_z[8] - 15) &&
t_position_z[h] < (hole_z[8] + 15))
{
    in_hole = 8;
}

```

```

}

if (t_position_x[h] < (hole_x[0] - 15) || t_position_x[h] > (hole_x[0] + 75) || t_position_z[h] < (hole_z[0] - 15) ||
t_position_z[h] > (hole_z[0] + 75))
{
in_hole = -1;
}

if (th_0 != 1 && th_1 != 1 && th_2 != 1 && th_3 != 1 && th_4 != 1 && th_5 != 1 && th_6 != 1 && th_7 != 1 && th_8 != 1
&& h_t != 1 && h_all !=1 && h_all_p != 1)
{
peg_t.SetPosition(peg_0.GetPosition[0], peg_0.GetPosition[1]+0.1, peg_0.GetPosition[2]);
peg_t.draw();
}

if ((h_t != 0 || h_0out == 1 || h_1out ==1 || h_2out == 1 || h_3out ==1 || h_4out == 1 || h_5out ==1 || h_6out == 1 ||
h_7out ==1 || h_8out ==1))
{
peg_t.SetPosition(peg.GetPosition[0], peg.GetPosition[1]+0.1, peg.GetPosition[2]);
peg_t.draw();
}

if (th_0 != 1 && th_1 != 1 && th_2 != 1 && th_3 != 1 && th_4 != 1 && th_5 != 1 && th_6 != 1 && th_7 != 1 && th_8 != 1
&& h_t != 1 && h_all_int != 1 && h_all_p != 1)
peg_0.draw();

if ((h_t != 0 || h_0out == 1 || h_1out ==1 || h_2out == 1 || h_3out ==1 || h_4out == 1 || h_5out ==1 || h_6out == 1 ||
h_7out ==1 || h_8out ==1))
peg.Draw();

if (h_all == 1)
{
for (i=0; i<9; i+=1)
{
if (in_hole == i)
{
angle_3 = 90 - 90*(sqrt( (t_position_x[h] - hole_x[i])^2 + (t_position_y[h] - hole_y[i] - peg_height/2)^2 +
(t_position_z[h] - hole_z[i])^2 ))/100;
angle_4 = 90 - 90*(sqrt( (t_position_x[h] - hole_x[i])^2 + (t_position_y[h] - hole_y[i] - peg_height/2)^2 +
(t_position_z[h] - hole_z[i])^2 ))/100;
angle_6 = 90 - 90*(sqrt( (t_position_x[h] - hole_x[i])^2 + (t_position_y[h] - hole_y[i] - peg_height/2)^2 +
(t_position_z[h] - hole_z[i])^2 ))/100;

angle_t1 = 50 - 50*(sqrt( (t_position_x[h] - hole_x[i])^2 + (t_position_y[h] - hole_y[i] - peg_height/2)^2 +
(t_position_z[h] - hole_z[i])^2 ))/100;
angle_t2 = 35 - 35*(sqrt( (t_position_x[h] - hole_x[i])^2 + (t_position_y[h] - hole_y[i] - peg_height/2)^2 +
(t_position_z[h] - hole_z[i])^2 ))/100;
angle_t3 = (-30) - (-30)*(sqrt( (t_position_x[h] - hole_x[i])^2 + (t_position_y[h] - hole_y[i] - peg_height/2)^2 +
(t_position_z[h] - hole_z[i])^2 ))/100;

```

```

        angle_i1 = 70 - 70*(sqrt( (t_position_x[h] - hole_x[i])^2 + (t_position_y[h] - hole_y[i] - peg_height/2)^2 +
(t_position_z[h] - hole_z[i])^2 ))/100;
        angle_i2 = 35 - 35*(sqrt( (t_position_x[h] - hole_x[i])^2 + (t_position_y[h] - hole_y[i] - peg_height/2)^2 +
(t_position_z[h] - hole_z[i])^2 ))/100;
        angle_i3 = 50 - 50*(sqrt( (t_position_x[h] - hole_x[i])^2 + (t_position_y[h] - hole_y[i] - peg_height/2)^2 +
(t_position_z[h] - hole_z[i])^2 ))/100;
    }
}

}

    if(h_t == 0 && (position_y[h]-peg_height) > (board_y[h]+Ly/2-19) && h_all != 1 && (h_0out != 1 || h_1out != 1 ||
h_2out != 1 || h_3out != 1 || h_4out != 1 || h_5out != 1 || h_6out != 1 || h_7out != 1 || h_8out != 1))// && (h_0drop != 1
|| h_1drop != 1 || h_2drop != 1 || h_3drop != 1 || h_4drop != 1 || h_5drop != 1 || h_6drop != 1 || h_7drop != 1 || h_8drop
!= 1) )
    {
        angle_3 = 90 - 90*(sqrt( (peg.GetPosition[0] - peg_0.GetPosition[0])^2 + (peg.GetPosition[1] -
peg_0.GetPosition[1])^2 + (peg.GetPosition[2] - peg_0.GetPosition[2])^2 ))/200;
        angle_4 = 90 - 90*(sqrt( (peg.GetPosition[0] - peg_0.GetPosition[0])^2 + (peg.GetPosition[1] -
peg_0.GetPosition[1])^2 + (peg.GetPosition[2] - peg_0.GetPosition[2])^2 ))/200;
        angle_6 = 90 - 90*(sqrt( (peg.GetPosition[0] - peg_0.GetPosition[0])^2 + (peg.GetPosition[1] -
peg_0.GetPosition[1])^2 + (peg.GetPosition[2] - peg_0.GetPosition[2])^2 ))/200;

        angle_t1 = 50 - 50*(sqrt( (peg.GetPosition[0] - peg_0.GetPosition[0])^2 + (peg.GetPosition[1] -
peg_0.GetPosition[1])^2 + (peg.GetPosition[2] - peg_0.GetPosition[2])^2 ))/200;
        angle_t2 = 35 - 35*(sqrt( (peg.GetPosition[0] - peg_0.GetPosition[0])^2 + (peg.GetPosition[1] -
peg_0.GetPosition[1])^2 + (peg.GetPosition[2] - peg_0.GetPosition[2])^2 ))/200;
        angle_t3 = (-30) - (-30)*(sqrt( (peg.GetPosition[0] - peg_0.GetPosition[0])^2 + (peg.GetPosition[1] -
peg_0.GetPosition[1])^2 + (peg.GetPosition[2] - peg_0.GetPosition[2])^2 ))/200;

        angle_i1 = 70 - 70*(sqrt( (peg.GetPosition[0] - peg_0.GetPosition[0])^2 + (peg.GetPosition[1] -
peg_0.GetPosition[1])^2 + (peg.GetPosition[2] - peg_0.GetPosition[2])^2 ))/200;
        angle_i2 = 35 - 35*(sqrt( (peg.GetPosition[0] - peg_0.GetPosition[0])^2 + (peg.GetPosition[1] -
peg_0.GetPosition[1])^2 + (peg.GetPosition[2] - peg_0.GetPosition[2])^2 ))/200;
        angle_i3 = 50 - 50*(sqrt( (peg.GetPosition[0] - peg_0.GetPosition[0])^2 + (peg.GetPosition[1] -
peg_0.GetPosition[1])^2 + (peg.GetPosition[2] - peg_0.GetPosition[2])^2 ))/200;
    }

/*

    var angle_3y = 90 - 90*(abs(h_toolView.tool.proxyPosition[1] - peg_0.GetPosition[0])/abs(120 -
peg_0.GetPosition[1]));
    var angle_4y = 90 - 90*(abs(h_toolView.tool.proxyPosition[1] - peg_0.GetPosition[0])/abs(120 -
peg_0.GetPosition[1]));
    var angle_6y = 90 - 90*(abs(h_toolView.tool.proxyPosition[1] - peg_0.GetPosition[0])/abs(120 -
peg_0.GetPosition[1]));

    var angle_tly = 50 - 50*(abs(h_toolView.tool.proxyPosition[1] - peg_0.GetPosition[0])/abs(120 -
peg_0.GetPosition[1]));
    var angle_t2y = 35 - 35*(abs(h_toolView.tool.proxyPosition[1] - peg_0.GetPosition[0])/abs(120 -
peg_0.GetPosition[1]));

```

```

    var angle_t3y = (-30) - (-30)*(abs(h_toolView.tool.proxyPosition[1] - peg_0.GetPosition[0])/abs(120 -
peg_0.GetPosition[1]));

    var angle_i1y = 70 - 70*(abs(h_toolView.tool.proxyPosition[1] - peg_0.GetPosition[0])/abs(120 -
peg_0.GetPosition[1]));
    var angle_i2y = 35 - 35*(abs(h_toolView.tool.proxyPosition[1] - peg_0.GetPosition[0])/abs(120 -
peg_0.GetPosition[1]));
    var angle_i3y = 50 - 50*(abs(h_toolView.tool.proxyPosition[1] - peg_0.GetPosition[0])/abs(120 -
peg_0.GetPosition[1]));
    //z

    var angle_3z = 90 - 90*(abs(h_toolView.tool.proxyPosition[2] - peg_0.GetPosition[0])/abs(hole_z[0] -
peg_0.GetPosition[2]));
    var angle_4z = 90 - 90*(abs(h_toolView.tool.proxyPosition[2] - peg_0.GetPosition[0])/abs(hole_z[0] -
peg_0.GetPosition[2]));
    var angle_6z = 90 - 90*(abs(h_toolView.tool.proxyPosition[2] - peg_0.GetPosition[0])/abs(hole_z[0] -
peg_0.GetPosition[2]));

    var angle_t1z = 50 - 50*(abs(h_toolView.tool.proxyPosition[2] - peg_0.GetPosition[0])/abs(hole_z[0] -
peg_0.GetPosition[2]));
    var angle_t2z = 35 - 35*(abs(h_toolView.tool.proxyPosition[2] - peg_0.GetPosition[0])/abs(hole_z[0] -
peg_0.GetPosition[2]));
    var angle_t3z = (-30) - (-30)*(abs(h_toolView.tool.proxyPosition[2] - peg_0.GetPosition[0])/abs(hole_z[0] -
peg_0.GetPosition[2]));

    var angle_i1z = 70 - 70*(abs(h_toolView.tool.proxyPosition[2] - peg_0.GetPosition[0])/abs(hole_z[0] -
peg_0.GetPosition[2]));
    var angle_i2z = 35 - 35*(abs(h_toolView.tool.proxyPosition[2] - peg_0.GetPosition[0])/abs(hole_z[0] -
peg_0.GetPosition[2]));
    var angle_i3z = 50 - 50*(abs(h_toolView.tool.proxyPosition[2] - peg_0.GetPosition[0])/abs(hole_z[0] -
peg_0.GetPosition[2]));
    */
    if (h_0out == 1 || h_1out == 1 || h_2out == 1 || h_3out == 1 || h_4out == 1 || h_5out == 1 || h_6out == 1 || h_7out
== 1 || h_8out ==1)
    {
        test = 1;

    }
    else
    {
        test = 0;
    }

    if (test != 1)
    {
        hand_avatar.JointSetRotation(3,angle_t1,0,1,0);
        hand_avatar.JointSetRotation(4,-angle_t2,0,0,1);
        hand_avatar.JointSetRotation(5,angle_t3,0,0,1);

        hand_avatar.JointSetRotation(8,angle_i1,0,1,0);
        hand_avatar.JointSetRotation(9,angle_i2,0,1,0);
        hand_avatar.JointSetRotation(10,angle_i3,0,1,0);

        hand_avatar.JointSetRotation(13,angle_3,0,1,0);

```

```

hand_avatar.JointSetRotation(14,angle_6,0,1,0);
hand_avatar.JointSetRotation(15,angle_4,0,1,0);

hand_avatar.JointSetRotation(18,angle_3,0,1,0);
hand_avatar.JointSetRotation(19,angle_6,0,1,0);
hand_avatar.JointSetRotation(20,angle_4,0,1,0);

hand_avatar.JointSetRotation(23,angle_3,0,1,0);
hand_avatar.JointSetRotation(24,angle_6,0,1,0);
hand_avatar.JointSetRotation(25,angle_4,0,1,0);
}

hand_avatar.SetPosition(h_toolView.tool.proxyPosition[0]+27 ,h_toolView.tool.proxyPosition[1] +
40,h_toolView.tool.proxyPosition[2] - 23);

    if (h_0==1 && h_0out !=1 && h_0drop != 1)
    {
peg0_t.Draw();
peg0.Draw();

    }
    if (h_1==1 && h_1out !=1 && h_1drop != 1)
    {
peg1_t.Draw();
peg1.Draw();

    }
    if (h_2==1 && h_2out !=1 && h_2drop != 1)
    {
peg2_t.Draw();
peg2.Draw();

    }
    if (h_3==1 && h_3out !=1 && h_3drop != 1)
    {
peg3_t.Draw();
peg3.Draw();

    }
    if (h_4==1 && h_4out !=1 && h_4drop != 1)
    {
peg4_t.Draw();
peg4.Draw();

    }
    if (h_5==1 && h_5out !=1 && h_5drop != 1)
    {
peg5_t.Draw();
peg5.Draw();

    }
    if (h_6==1 && h_6out !=1 && h_6drop != 1)

```

```
    {
    peg6_t.Draw();
    peg6.Draw();

    }
    if (h_7==1 && h_7out !=1 && h_7drop != 1)
    {
    peg7_t.Draw();
    peg7.Draw();

    }
    if (h_8==1 && h_8out !=1 && h_8drop != 1)
    {
    peg8_t.Draw();
    peg8.Draw();

    }
}
```

```
if (th_0 == 1 && tth_0 != 1)
dong.Play(false);
```

```
if (th_1 == 1 && tth_1 != 1)
dong.Play(false);
```

```
if (th_2 == 1 && tth_2 != 1)
dong.Play(false);
```

```
if (th_3 == 1 && tth_3 != 1)
dong.Play(false);
```

```
if (th_4 == 1 && tth_4 != 1)
dong.Play(false);
```

```
if (th_5 == 1 && tth_5 != 1)
dong.Play(false);
```

```
if (th_6 == 1 && tth_6 != 1)
dong.Play(false);
```

```
if (th_7 == 1 && tth_7 != 1)
dong.Play(false);
```

```
if (th_8 == 1 && tth_8 != 1)
dong.Play(false);
```

```
if(h_0 == 1)
tth_0 = 1;
```

```
if(h_1 == 1)
tth_1 = 1;
```

```

if(h_2 == 1)
tth_2 = 1;

if(h_3 == 1)
tth_3 = 1;

if(h_4 == 1)
tth_4 = 1;

if(h_5 == 1)
tth_5 = 1;

if(h_6 == 1)
tth_6 = 1;

if(h_7 == 1)
tth_7 = 1;

if(h_8 == 1)
tth_8 = 1;

if (h_all == 1)
th_all = 1;

sphere.SetPosition(h_toolView.tool.proxyPosition[0],t_position_y[h],h_toolView.tool.proxyPosition[2]);
sphere.SetPosition(peg.GetPosition[0],peg.GetPosition[1],peg.GetPosition[2]);

// real_rand = rand(32767)/328.0;

if (hold == 1)
{
outputLN(norm_dist,",",d_norm_dist,",",k_norm,",",b_norm,",",RMS0,",",S0_I_mjm,",",D_dist_mjm,",",k_time,",",b_mjm,",",RMS1
}
ConsoleText(0.7,0.88, Sprintf(" S0_I_mjm %f", (S0_I_mjm)));
//ConsoleText(0.7,0.86, Sprintf(" T-ts %f", (GetTime()-tau_start)));

/*

ConsoleText(0.7,0.9, Sprintf(" sign %f", (delta_tau)));
ConsoleText(0.7,0.88, Sprintf("yH_loc %f", (yH_loc)));
ConsoleText(0.7,0.86, Sprintf("zH_loc %f", (zH_loc)));
ConsoleText(0.7,0.84, Sprintf("X_Iloc[0] %f", (X_Iloc[0]));
ConsoleText(0.7,0.82, Sprintf("X_Iloc[0] %f", (X_Iloc[1]));
ConsoleText(0.7,0.80, Sprintf("X_Iloc[0] %f", (X_Iloc[2]));
ConsoleText(0.7,0.78, Sprintf("sign %f", ((D_norm_dist*abs(D_norm_dist)^-1))));
ConsoleText(0.7,0.76, Sprintf("F_help_x %f", (F_help_x)));
ConsoleText(0.7,0.74, Sprintf("F_help_y %f", (F_help_y)));
ConsoleText(0.7,0.72, Sprintf("F_help_z %f", (F_help_z)));
ConsoleText(0.7,0.70, Sprintf("RMS0 %f", (RMS0)));

```



```

    ConsoleText(0.7,0.68, Sprintf("epsilon %f", (epsilon)));
    ConsoleText(0.7,0.66, Sprintf("hn %f", (hn)));
    ConsoleText(0.7,0.64, Sprintf("abs(tau[h+1]-tau[h]) %f", (abs(tau[h+1]-tau[h]))));
    ConsoleText(0.7,0.62, Sprintf("abs(tau[h]) %f", (abs(tau[h]))));
    ConsoleText(0.7,0.60, Sprintf("abs(tau[h+1]-tau[h]) * abs(tau[h])^-1 %f", (abs(tau[h+1]-tau[h]) *
abs(tau[h])^-1)));
    ConsoleText(0.7,0.58, Sprintf("Ratio5 %f", (Ratio5)));
    ConsoleText(0.7,0.56, Sprintf("Ratio6 %f", (Ratio6)));
    ConsoleText(0.7,0.54, Sprintf("D_plane_dist %f", (D_plane_dist)));
    ConsoleText(0.7,0.52, Sprintf("D_norm_dist %f", ((D_norm_dist))));
*/

// ConsoleText(0.7,0.9, Sprintf("x %f", (peg_t.GetPosition[0])));
// ConsoleText(0.7,0.88, Sprintf("y %f", (peg_t.GetPosition[1])));
// ConsoleText(0.7,0.86, Sprintf("z %f", (peg_t.GetPosition[2])));
// ConsoleText(0.7,0.84, Sprintf("time %f", (time)));
// ConsoleText(0.7,0.82, Sprintf("hold %f", (hold)));
// ConsoleText(0.7,0.80, Sprintf("x %f", (position_x[h]));
// ConsoleText(0.7,0.78, Sprintf("y %f", (position_y[h]));
// ConsoleText(0.7,0.76, Sprintf("z %f", (position_z[h]));
// ConsoleText(0.7,0.74, Sprintf("peg_0_x %f", (peg0.GetPosition[0]));
// ConsoleText(0.7,0.72, Sprintf("peg_0_x %f", (h_toolView.tool.lastForce[0]));
// +(-0.2*0.1*F_m_x*(peg0.GetPosition[0]-velocity_x[h])*dam_peg*2 +
(peg0.GetPosition[0]-position_x[h])*stif_peg*0.1)
// ConsoleText(0.7,0.6, Sprintf("test %f", (test)));

// ConsoleText(0.1,0.9, Sprintf("hole_0 %f", (h_0)));
// ConsoleText(0.1,0.88, Sprintf("hole_1 %f", (h_1)));
// ConsoleText(0.1,0.86, Sprintf("hole_2 %f", (h_2)));
// ConsoleText(0.1,0.84, Sprintf("hole_3 %f", (h_3)));
// ConsoleText(0.1,0.82, Sprintf("hole_4 %f", (h_4)));
// ConsoleText(0.1,0.80, Sprintf("hole_5 %f", (h_5)));
// ConsoleText(0.1,0.78, Sprintf("hole_6 %f", (h_6)));
// ConsoleText(0.1,0.76, Sprintf("hole_7 %f", (h_7)));
// ConsoleText(0.1,0.74, Sprintf("hole_8 %f", (h_8)));

// ConsoleText(0.1,0.72, Sprintf("next_hole %f", (next_hole)));

// ConsoleText(0.1,0.70, Sprintf("h_0out %f", (h_0out)));
// ConsoleText(0.1,0.68, Sprintf("h_0outout %f", (h_0outout)));

// ConsoleText(0.1,0.66, Sprintf("h_t %f", (h_t)));
/*

ConsoleText(0.1,0.66, Sprintf("x_1 %f", (x_1)));
ConsoleText(0.1,0.64, Sprintf("y_1 %f", (y_1)));

ConsoleText(0.1,0.60, Sprintf("z_1 %f", (z_1)));
ConsoleText(0.1,0.58, Sprintf("x_2 %f", (x_2)));

ConsoleText(0.1,0.54, Sprintf("y_2 %f", (y_2)));
ConsoleText(0.1,0.52, Sprintf("z_2 %f", (z_2)));

```

```

ConsoleText(0.1,0.50, Sprintf("x_1d %f", (x_1d)));
ConsoleText(0.1,0.48, Sprintf("y_1d %f", (y_1d)));

ConsoleText(0.1,0.44, Sprintf("z_1d %f", (z_1d)));
ConsoleText(0.1,0.42, Sprintf("phid %f", (phid)));

ConsoleText(0.1,0.38, Sprintf("L_1 %f", (L_1)));
ConsoleText(0.1,0.36, Sprintf("L_2 %f", (L_2)));

ConsoleText(0.1,0.32, Sprintf("L_3 %f", (L_3)));
ConsoleText(0.1,0.30, Sprintf("L_4 %f", (L_4)));

ConsoleText(0.1,0.26, Sprintf("Alpha3 %f", (Alpha3)));

ConsoleText(0.1,0.22, Sprintf("theta2 %f", (theta2)));

ConsoleText(0.1,0.20, Sprintf("thetad %f", (thetad)));
ConsoleText(0.1,0.18, Sprintf("x_3 %f", (x_3)));
ConsoleText(0.1,0.16, Sprintf("y_3 %f", (y_3)));

ConsoleText(0.1,0.14, Sprintf("z_3 %f", (z_3)));
ConsoleText(0.1,0.12, Sprintf("AA %f", (AA)));
ConsoleText(0.1,0.10, Sprintf("BB %f", (BB)));
ConsoleText(0.1,0.08, Sprintf("CC %f", (CC)));

ConsoleText(0.1,0.06, Sprintf("DD %f", (DD)));
ConsoleText(0.1,0.04, Sprintf("b1x %f", (b1x)));
ConsoleText(0.1,0.02, Sprintf("b1y %f", (b1y)));

    ConsoleText(0.1,0.70, Sprintf("b1z %f", (b1z)));

    ConsoleText(0.9,0.9, Sprintf("b2x %f", (b2x)));
    ConsoleText(0.9,0.88, Sprintf("b2y %f", (b2y)));
    ConsoleText(0.9,0.86, Sprintf("b2z %f", (b2z)));

    ConsoleText(0.9,0.84, Sprintf("b3x %f", (b3x)));
    ConsoleText(0.9,0.82, Sprintf("b3y %f", (b3y)));
    ConsoleText(0.9,0.8, Sprintf("b3z %f", (b3z)));

    ConsoleText(0.9,0.78, Sprintf(" X_1loc[0] %f", ( X_1loc[0])));
    ConsoleText(0.9,0.76, Sprintf(" X_1loc[1] %f", ( X_1loc[1])));
    ConsoleText(0.9,0.74, Sprintf(" X_1loc[2] %f", ( X_1loc[2])));

// ConsoleText(0.9,0.8, Sprintf("b1x %f", (b_1[0])));
// ConsoleText(0.9,0.78, Sprintf("b1y %f", (b_1[1])));
// ConsoleText(0.9,0.76, Sprintf("b1z %f", (b_1[2])));
// ConsoleText(0.9,0.74, Sprintf("Check b1 %f", (sqrt(b_1[0]^2 + b_1[1]^2 + b_1[2]^2))));

    ConsoleText(0.9,0.7, Sprintf(" X_2loc[0] %f", ( X_2loc[0])));
    ConsoleText(0.9,0.68, Sprintf(" X_2loc[1] %f", ( X_2loc[1])));

```

```

ConsoleText(0.9,0.66, Sprintf(" X_2loc[2] %f", ( X_2loc[2])));
ConsoleText(0.9,0.64, Sprintf("Check b2 %f", (sqrt(b_2[0]^2 + b_2[1]^2 + b_2[2]^2))));

ConsoleText(0.9,0.6, Sprintf(" X_3loc[0] %f", ( X_3loc[0])));
ConsoleText(0.9,0.58, Sprintf(" X_3loc[1] %f", ( X_3loc[1])));
ConsoleText(0.9,0.56, Sprintf(" X_3loc[2] %f", ( X_3loc[2])));
ConsoleText(0.9,0.54, Sprintf("Check b3 %f", (sqrt(b_3[0]^2 + b_3[1]^2 + b_3[2]^2))));

ConsoleText(0.9,0.42, Sprintf("mjm_a3 %f", (mjm_a3)));
ConsoleText(0.9,0.40, Sprintf("mjm_a4 %f", (mjm_a4)));
ConsoleText(0.9,0.38, Sprintf("mjm_a5 %f", (mjm_a5)));

ConsoleText(0.9,0.34, Sprintf("mjm_b0 %f", (mjm_b0)));
ConsoleText(0.9,0.32, Sprintf("mjm_b1 %f", (mjm_b1)));
ConsoleText(0.9,0.30, Sprintf("mjm_b2 %f", (mjm_b2)));

ConsoleText(0.9,0.28, Sprintf("mjm_b3 %f", (mjm_b3)));
ConsoleText(0.9,0.26, Sprintf("mjm_b4 %f", (mjm_b4)));
ConsoleText(0.9,0.24, Sprintf("mjm_b5 %f", (mjm_b5)));

ConsoleText(0.9,0.20, Sprintf("Alpha3 %f", (Alpha3)));

ConsoleText(0.9,0.18, Sprintf("theta2 %f", (theta2)));

ConsoleText(0.9,0.14, Sprintf("theta2_low %f", (theta2_low)));
ConsoleText(0.9,0.12, Sprintf("fn %f", (fn)));
ConsoleText(0.9,0.10, Sprintf("dfn %f", (dfn)));

ConsoleText(0.9,0.5, Sprintf("x_2 %f", (x_2)));

ConsoleText(0.5,0.5, Sprintf("fn"));
ConsoleText(0.5,0.48, Sprintf("tau[hn]^9 %f", ((10*mjm_a5^2 + 10*mjm_b5^2))));
ConsoleText(0.5,0.46, Sprintf("tau[hn]^8 %f", ((18*mjm_a4*mjm_a5 + 18*mjm_b4*mjm_b5)*0^8));
ConsoleText(0.5,0.44, Sprintf("tau[hn]^7 %f", ((8*mjm_a4^2 + 16*mjm_a3*mjm_a5 + 8*mjm_b4^2 + 16*mjm_b3*mjm_b5)*0^7));
ConsoleText(0.5,0.42, Sprintf("tau[hn]^6 %f", (14*mjm_a3*mjm_a4 + 14*mjm_a2*mjm_a5 + 14*mjm_b3*mjm_b4 +
14*mjm_b2*mjm_b5)*0^6));
ConsoleText(0.5,0.40, Sprintf("tau[hn]^5 %f", (6*mjm_a3^2 + 12*mjm_a2*mjm_a4 + 12*mjm_a1*mjm_a5 + 6*mjm_b3^2 +
12*mjm_b2*mjm_b4 + 12*mjm_b1*mjm_b5)*0^5));
ConsoleText(0.5,0.38, Sprintf("tau[hn]^4 %f", (10*mjm_a2*mjm_a3 + 10*mjm_a1*mjm_a4 + 10*mjm_a0*mjm_a5 +
10*mjm_b2*mjm_b3 + 10*mjm_b1*mjm_b4 + 10*mjm_b0*mjm_b5 - 10*mjm_a5*xHI_loc - 10*mjm_b5*yHI_loc)*0^4));
ConsoleText(0.5,0.36, Sprintf("tau[hn]^3 %f", (4*mjm_a2^2 + 8*mjm_a1*mjm_a3 + 8*mjm_a0*mjm_a4 + 4*mjm_b2^2 +
8*mjm_b1*mjm_b3 + 8*mjm_b0*mjm_b4 - 8*mjm_a4*xHI_loc - 8*mjm_b4*yHI_loc)*0^3));
ConsoleText(0.5,0.34, Sprintf("tau[hn]^2 %f", (6*mjm_a1*mjm_a2 + 6*mjm_a0*mjm_a3 + 6*mjm_b1*mjm_b2 + 6*mjm_b0*mjm_b3 -
6*mjm_a3*xHI_loc - 6*mjm_b3*yHI_loc)*0^2));
ConsoleText(0.5,0.32, Sprintf("tau[hn] %f", (2*mjm_a1^2 + 4*mjm_a0*mjm_a2 + 2*mjm_b1^2 + 4*mjm_b0*mjm_b2 -
4*mjm_a2*xHI_loc - 4*mjm_b2*yHI_loc)*0^1));
ConsoleText(0.5,0.30, Sprintf("tau[hn]^0 %f", (2*mjm_a0*mjm_a1 + 2*mjm_b0*mjm_b1 - 2*mjm_a1*xHI_loc -
2*mjm_b1*yHI_loc)*0^0));

```

```

    ConsoleText(0.5,0.28, Sprintf("tau[hn] - add %f",((10*mjm_a5^2 + 10*mjm_b5^2) + (18*mjm_a4*mjm_a5 + 18*mjm_b4*mjm_b5)
+ (8*mjm_a4^2 + 16*mjm_a3*mjm_a5 + 8*mjm_b4^2 + 16*mjm_b3*mjm_b5) + (14*mjm_a3*mjm_a4 + 14*mjm_a2*mjm_a5 +
14*mjm_b3*mjm_b4 + 14*mjm_b2*mjm_b5) + (6*mjm_a3^2 + 12*mjm_a2*mjm_a4 + 12*mjm_a1*mjm_a5 + 6*mjm_b3^2 + 12*mjm_b2*mjm_b4 +
12*mjm_b1*mjm_b5) + (10*mjm_a2*mjm_a3 + 10*mjm_a1*mjm_a4 + 10*mjm_a0*mjm_a5 + 10*mjm_b2*mjm_b3 + 10*mjm_b1*mjm_b4 +
10*mjm_b0*mjm_b5 - 10*mjm_a5*xHI_loc - 10*mjm_b5*yHI_loc) + (4*mjm_a2^2 + 8*mjm_a1*mjm_a3 + 8*mjm_a0*mjm_a4 + 4*mjm_b2^2 +
8*mjm_b1*mjm_b3 + 8*mjm_b0*mjm_b4 - 8*mjm_a4*xHI_loc - 8*mjm_b4*yHI_loc) + (6*mjm_a1*mjm_a2 + 6*mjm_a0*mjm_a3 +
6*mjm_b1*mjm_b2 + 6*mjm_b0*mjm_b3 - 6*mjm_a3*xHI_loc - 6*mjm_b3*yHI_loc) + (2*mjm_a1^2 + 4*mjm_a0*mjm_a2 + 2*mjm_b1^2 +
4*mjm_b0*mjm_b2 - 4*mjm_a2*xHI_loc - 4*mjm_b2*yHI_loc) + (2*mjm_a0*mjm_a1 + 2*mjm_b0*mjm_b1 - 2*mjm_a1*xHI_loc -
2*mjm_b1*yHI_loc))));

    ConsoleText(0.3,0.5, Sprintf("dfn"));
    ConsoleText(0.3,0.48, Sprintf("tau[hn]^8 %f", (9*(10*mjm_a5^2 + 10*mjm_b5^2))));
    ConsoleText(0.3,0.46, Sprintf("tau[hn]^7 %f", (8*(18*mjm_a4*mjm_a5 + 18*mjm_b4*mjm_b5))));
    ConsoleText(0.3,0.44, Sprintf("tau[hn]^6 %f", (7*(8*mjm_a4^2 + 16*mjm_a3*mjm_a5 + 8*mjm_b4^2 + 16*mjm_b3*mjm_b5))));
    ConsoleText(0.3,0.42, Sprintf("tau[hn]^5 %f", (6*(14*mjm_a3*mjm_a4 + 14*mjm_a2*mjm_a5 + 14*mjm_b3*mjm_b4 +
14*mjm_b2*mjm_b5))));
    ConsoleText(0.3,0.40, Sprintf("tau[hn]^4 %f", (5*(6*mjm_a3^2 + 12*mjm_a2*mjm_a4 + 12*mjm_a1*mjm_a5 + 6*mjm_b3^2 +
12*mjm_b2*mjm_b4 + 12*mjm_b1*mjm_b5))));
    ConsoleText(0.3,0.38, Sprintf("tau[hn]^3 %f", (4*(10*mjm_a2*mjm_a3 + 10*mjm_a1*mjm_a4 + 10*mjm_a0*mjm_a5 +
10*mjm_b2*mjm_b3 + 10*mjm_b1*mjm_b4 + 10*mjm_b0*mjm_b5 - 10*mjm_a5*xHI_loc - 10*mjm_b5*yHI_loc))));
    ConsoleText(0.3,0.36, Sprintf("tau[hn]^2 %f", (3*(4*mjm_a2^2 + 8*mjm_a1*mjm_a3 + 8*mjm_a0*mjm_a4 + 4*mjm_b2^2 +
8*mjm_b1*mjm_b3 + 8*mjm_b0*mjm_b4 - 8*mjm_a4*xHI_loc - 8*mjm_b4*yHI_loc))));
    ConsoleText(0.3,0.34, Sprintf("tau[hn]^1 %f", (2*(6*mjm_a1*mjm_a2 + 6*mjm_a0*mjm_a3 + 6*mjm_b1*mjm_b2 + 6*mjm_b0*mjm_b3
- 6*mjm_a3*xHI_loc - 6*mjm_b3*yHI_loc))));
    ConsoleText(0.3,0.32, Sprintf("tau[hn]^0 %f", (2*mjm_a1^2 + 4*mjm_a0*mjm_a2 + 2*mjm_b1^2 + 4*mjm_b0*mjm_b2 -
4*mjm_a2*xHI_loc - 4*mjm_b2*yHI_loc));
    ConsoleText(0.3,0.30, Sprintf("tau[hn] - add %f",((9*(10*mjm_a5^2 + 10*mjm_b5^2)) + (8*(18*mjm_a4*mjm_a5 +
18*mjm_b4*mjm_b5)) + (7*(8*mjm_a4^2 + 16*mjm_a3*mjm_a5 + 8*mjm_b4^2 + 16*mjm_b3*mjm_b5)) + (6*(14*mjm_a3*mjm_a4 +
14*mjm_a2*mjm_a5 + 14*mjm_b3*mjm_b4 + 14*mjm_b2*mjm_b5)) + (5*(6*mjm_a3^2 + 12*mjm_a2*mjm_a4 + 12*mjm_a1*mjm_a5 +
6*mjm_b3^2 + 12*mjm_b2*mjm_b4 + 12*mjm_b1*mjm_b5)) + (4*(10*mjm_a2*mjm_a3 + 10*mjm_a1*mjm_a4 + 10*mjm_a0*mjm_a5 +
10*mjm_b2*mjm_b3 + 10*mjm_b1*mjm_b4 + 10*mjm_b0*mjm_b5 - 10*mjm_a5*xHI_loc - 10*mjm_b5*yHI_loc)) + (3*(4*mjm_a2^2 +
8*mjm_a1*mjm_a3 + 8*mjm_a0*mjm_a4 + 4*mjm_b2^2 + 8*mjm_b1*mjm_b3 + 8*mjm_b0*mjm_b4 - 8*mjm_a4*xHI_loc - 8*mjm_b4*yHI_loc))
+ (2*(6*mjm_a1*mjm_a2 + 6*mjm_a0*mjm_a3 + 6*mjm_b1*mjm_b2 + 6*mjm_b0*mjm_b3 - 6*mjm_a3*xHI_loc - 6*mjm_b3*yHI_loc)) +
(2*mjm_a1^2 + 4*mjm_a0*mjm_a2 + 2*mjm_b1^2 + 4*mjm_b0*mjm_b2 - 4*mjm_a2*xHI_loc - 4*mjm_b2*yHI_loc))));

*/

/*
ConsoleText(0.9,0.22, Sprintf("R_safe %f", (R_safe)));
ConsoleText(0.9,0.2, Sprintf("R_H %f", (R_H)));

ConsoleText(0.9,0.5, Sprintf("dxi_loc[0] %f", (dxi_loc[0]));
ConsoleText(0.9,0.48, Sprintf("dxi_loc[1] %f", (dxi_loc[1]));
ConsoleText(0.9,0.46, Sprintf("dxi_loc[2] %f", (dxi_loc[2]));

ConsoleText(0.7,0.5, Sprintf("inst_vel_x %f", (inst_vel_x));
ConsoleText(0.7,0.48, Sprintf("inst_vel_y %f", (inst_vel_y));
Text(0.7,0.46, Sprintf("inst_vel_z %f", (inst_vel_z));
*/

```

```

//      dfn = (9*(10*mjm_a5^2 + 10*mjm_b5^2))*tau[hn]^8 + (8*(18*mjm_a4*mjm_a5 + 18*mjm_b4*mjm_b5))*tau[hn]^7 +
(7*(8*mjm_a4^2 + 16*mjm_a3*mjm_a5 + 8*mjm_b4^2 + 16*mjm_b3*mjm_b5))*tau[hn]^6 + (6*(14*mjm_a3*mjm_a4 + 14*mjm_a2*mjm_a5 +
14*mjm_b3*mjm_b4 + 14*mjm_b2*mjm_b5))*tau[hn]^5 + (5*(6*mjm_a3^2 + 12*mjm_a2*mjm_a4 + 12*mjm_a1*mjm_a5 + 6*mjm_b3^2 +
12*mjm_b2*mjm_b4 + 12*mjm_b1*mjm_b5))*tau[hn]^4 + (4*(10*mjm_a2*mjm_a3 + 10*mjm_a1*mjm_a4 + 10*mjm_a0*mjm_a5 +
10*mjm_b2*mjm_b3 + 10*mjm_b1*mjm_b4 + 10*mjm_b0*mjm_b5 - 10*mjm_a5*xHI_loc - 10*mjm_b5*yHI_loc))*tau[hn]^3 +
(3*(4*mjm_a2^2 + 8*mjm_a1*mjm_a3 + 8*mjm_a0*mjm_a4 + 4*mjm_b2^2 + 8*mjm_b1*mjm_b3 + 8*mjm_b0*mjm_b4 - 8*mjm_a4*xHI_loc -
8*mjm_b4*yHI_loc))*tau[hn]^2 + (2*(6*mjm_a1*mjm_a2 + 6*mjm_a0*mjm_a3 + 6*mjm_b1*mjm_b2 + 6*mjm_b0*mjm_b3 -
6*mjm_a3*xHI_loc - 6*mjm_b3*yHI_loc))*tau[hn] + (2*mjm_a1^2 + 4*mjm_a0*mjm_a2 + 2*mjm_b1^2 + 4*mjm_b0*mjm_b2 -
4*mjm_a2*xHI_loc - 4*mjm_b2*yHI_loc);

```

```

th_0 = 0;
th_1 = 0;
th_2 = 0;
th_3 = 0;
th_4 = 0;
th_5 = 0;
th_6 = 0;
th_7 = 0;
th_8 = 0;

```

```

hand_avatar.Draw();
SceneEnd();

```

```

}

```

```

function DownloadReady( RequestID )
{
    // TO DO
}

```

```

function OnTimer()
{
    m=10;
    c=0;
    b=10;
    a=1;

    h=h+1;

    time = GetTime();

```

```

//reset the arrays if exceed v_size maximum. (saving memory);
if(time == 300000)
{
    h=2;

```

```
position_x[h-2] = position_x[v_size-2]; // mm
position_x[h-1] = position_x[v_size-1];
position_x[h] = position_x[v_size];
velocity_x[h] = velocity_x[v_size]; //mm/s
```

```
position_y[h-2] = position_y[v_size-2]; // mm
position_y[h-1] = position_y[v_size-1];
position_y[h] = position_y[v_size];
velocity_y[h] = velocity_y[v_size]; //mm/s
```

```
position_z[h-1] = position_z[v_size-2]; // mm
position_z[h-2] = position_z[v_size-1]; // mm
position_z[h] = position_z[v_size];
velocity_z[h] = velocity_z[v_size]; //mm/s
```

```
t_position_x[h] = t_position_x[v_size]; // mm
t_position_x[h-1] = t_position_x[v_size-1]; // mm
t_position_x[h-2] = t_position_x[v_size-2]; // mm
t_velocity_x[h] = t_velocity_x[v_size]; //mm/s
```

```
t_position_y[h] = t_position_y[v_size]; // mm
t_position_y[h-1] = t_position_y[v_size-1]; // mm
t_position_y[h-2] = t_position_y[v_size-2]; // mm
t_velocity_y[h] = t_velocity_y[v_size]; //mm/s
```

```
t_position_z[h] = t_position_z[v_size]; // mm
t_position_z[h-1] = t_position_z[v_size-1]; // mm
t_position_z[h-2] = t_position_z[v_size-2]; // mm
t_velocity_z[h] = t_velocity_z[v_size]; //mm/s
```

```
board_x[h-2] = board_x[v_size-2];
board_x[h-1] = board_x[v_size-1];
board_x[h] = board_x[v_size];
```

```
board_y[h-2] = board_y[v_size-2];
board_y[h-1] = board_y[v_size-1];
board_y[h] = board_y[v_size];
```

```
board_z[h-2] = board_z[v_size-2];
board_z[h-1] = board_z[v_size-1];
board_z[h] = board_z[v_size];
```

```
}
```

```
//board position... for now it's static
board_x[h] = board_pos_in[0];
```

```

board_y[h] = board_pos_in[1];
board_z[h] = board_pos_in[2];

board.SetPosition(board_x[h], board_y[h], board_z[h]);
//first determine the position of far-left hole (0,0). We must always know or be able to calculate the position of
this hole
//position of centre hole = board_pos_in.
//position of the 0,0 hole in the x-dir = board_pos_in[0]-hole_dis
//position of the 0,0 hole in the y-dir = Ly
//position of the 0,0 hole in the z-dir = board_pos_in[2]-hole_dis
//or:
hole_x_in[0]=board_x[h]-hole_dis;
hole_y_in[0]=board_y[h] + 20;
hole_z_in[0]=board_z[h]-hole_dis;

var k=0;
for( j=0; j<nx; j+=1)
{
hole_x_in[j] = hole_x_in[0]+hole_dis*k;
hole_y_in[j] = hole_y_in[0];
hole_z_in[j] = hole_z_in[0];
k=k+1;
}

var kk=0;
for( j=nx; j<nx+nx; j+=1)
{
hole_x_in[j] = hole_x_in[0]+hole_dis*kk;
hole_y_in[j] = hole_y_in[0];
hole_z_in[j] = hole_z_in[0]+hole_dis;
kk=kk+1;
}

var kkk=0;
for( j=nx+nx; j<nx+nx+nx; j+=1)
{
hole_x_in[j] = hole_x_in[0]+hole_dis*kkk;
hole_y_in[j] = hole_y_in[0];
hole_z_in[j] = hole_z_in[0]+2*hole_dis;
kkk=kkk+1;
}

hole_x[0] = hole_x_in[0];
hole_y[0] = hole_y_in[0];
hole_z[0] = hole_z_in[0];

hole_x[1] = hole_x_in[3];
hole_y[1] = hole_y_in[3];
hole_z[1] = hole_z_in[3];

hole_x[2] = hole_x_in[6];
hole_y[2] = hole_y_in[6];
hole_z[2] = hole_z_in[6];

```

```
hole_x[3] = hole_x_in[7];
hole_y[3] = hole_y_in[7];
hole_z[3] = hole_z_in[7];
```

```
hole_x[4] = hole_x_in[4];
hole_y[4] = hole_y_in[4];
hole_z[4] = hole_z_in[4];
```

```
hole_x[5] = hole_x_in[1];
hole_y[5] = hole_y_in[1];
hole_z[5] = hole_z_in[1];
```

```
hole_x[6] = hole_x_in[2];
hole_y[6] = hole_y_in[2];
hole_z[6] = hole_z_in[2];
```

```
hole_x[7] = hole_x_in[5];
hole_y[7] = hole_y_in[5];
hole_z[7] = hole_z_in[5];
```

```
hole_x[8] = hole_x_in[8];
hole_y[8] = hole_y_in[8];
hole_z[8] = hole_z_in[8];
```

```
//determine numerically the positions, velocities and accelerations of the peg and tool
```

```
//if((abs(t_position_x[h] - position_x[h]) =< 4) && (abs(t_position_y[h] - position_y[h]) =< 6) && (abs(t_position_z[h] -
position_z[h]) =< 4) && (h_toolView.tool.buttonPressed)) //&& peg.GetPosition[1] != cubel.GetPosition[1])
if((abs(h_toolView.tool.proxyPosition[0] - peg.GetPosition[0]) =< 4) && (abs(h_toolView.tool.proxyPosition[1] -
peg.GetPosition[1]) =< 4) && (abs(h_toolView.tool.proxyPosition[2] - peg.GetPosition[2]) =< 4) &&
(h_toolView.tool.buttonPressed)) //&& peg.GetPosition[1] != cubel.GetPosition[1])
```

```
a=1;
else
a=1;
```

```
if(a==1)
{
```

```
t_position_x[h] = h_toolView.tool.proxyPosition[0];
t_velocity_x[h] = (t_position_x[h] - t_position_x[h-1])/T; //m/s
```

```
t_position_y[h] = h_toolView.tool.proxyPosition[1];
t_velocity_y[h] = (t_position_y[h] - t_position_y[h-1])/T; //m/s
```

```
t_position_z[h] = h_toolView.tool.proxyPosition[2];
t_velocity_z[h] = (t_position_z[h] - t_position_z[h-1])/T; //m/s
```



```

    position_x[h]=( position_x[h-1]* (2*m_peg/T^2 + dam_peg/T) - position_x[h-2]*m_peg/T^2 + t_velocity_x[h]*dam_peg +
t_position_x[h] * stif_peg) / (m_peg/T^2 + dam_peg/T + stif_peg) ;
    position_y[h]=( position_y[h-1]* (2*m_peg/T^2 + dam_peg/T) - position_y[h-2]*m_peg/T^2 + t_velocity_y[h]*dam_peg +
t_position_y[h] * stif_peg - m_peg*9810) / (m_peg/T^2 + dam_peg/T + stif_peg) ;
    position_z[h]=( position_z[h-1]* (2*m_peg/T^2 + dam_peg/T) - position_z[h-2]*m_peg/T^2 + t_velocity_z[h]*dam_peg +
t_position_z[h] * stif_peg) / (m_peg/T^2 + dam_peg/T + stif_peg) ;
    peg.SetPosition(position_x[h], position_y[h] + y_cor, position_z[h]);

```

```

velocity_x[h] = (position_x[h] - position_x[h-1])/T; //m/s
velocity_y[h] = (position_y[h] - position_y[h-1])/T;
velocity_z[h] = (position_z[h] - position_z[h-1])/T;

```

```

vxH = velocity_x[h];
vyH = velocity_y[h];
vzH = velocity_z[h];

```

```

L_1x = board_x[h] + Lx/2 + Lx;
L_1x_in = board_x[h] + Lx/2 + Lx - peg_diam/6;
L_2x = board_x[h] - Lx/2;
L_2x_in = board_x[h] - Lx/2 + peg_diam/6;
L_3z = board_z[h] + Lz/2;
L_3z_in = board_z[h] + Lz/2 - peg_diam/6;
L_4z = board_z[h] - Lz/2;
L_4z_in = board_z[h] - Lz/2 + peg_diam/6;
L_5y = board_y[h] - Ly/2;
L_5y_in = board_y[h] - Ly/2 + peg_height/6;
L_6y = board_y[h] + Ly/2;
    //check were the peg is

```

```

    if (position_y[h]-peg_height =< L_6y && t_position_y[h] > L_5y)
        b=11;
    if (position_y[h]-peg_height => L_5y && t_position_y[h] <= L_6y)
        b=11;;
    if (position_y[h] =< L_5y)
        b=10;
    if (position_y[h] =< L_5y_in && position_y[h] >= L_5y)
        b=12;

```

```

    if ((position_z[h] + peg_diam/2) < L_4z && (position_x[h] + peg_diam/2) < L_2x)
        c=11;
    if ((position_z[h] + peg_diam/2) < L_4z && (position_x[h]-peg_diam/2) < L_1x && (position_x[h] + peg_diam/2) > L_2x)
        c=12;
    if ((position_z[h] + peg_diam/2) =< L_4z_in && (position_z[h] + peg_diam/2) => L_4z && (position_x[h]-peg_diam/2) =<
L_1x && (position_x[h] + peg_diam/2) => L_2x)
        c=122;
    if ((position_z[h] + peg_diam/2) < L_4z && (position_x[h] - peg_diam/2) >= L_1x)
        c=13;

```

```

    if ((position_z[h] - peg_diam/2) < L_3z && (position_z[h] + peg_diam/2) > L_4z && (position_x[h] + peg_diam/2) < L_2x)
        c=21;
    if ((position_z[h] - peg_diam/2) =< L_3z && (position_z[h] + peg_diam/2) => L_4z && (position_x[h] + peg_diam/2) =<
L_2x_in && (position_x[h] + peg_diam/2) => L_2x)
        c=212;
    if ((position_z[h] - peg_diam/2) =< L_3z_in && (position_z[h] + peg_diam/2) => L_4z_in && (position_x[h] - peg_diam/2)
=< L_1x_in && (position_x[h] + peg_diam/2) >= L_2x_in)
        c=22;
    if ((position_z[h] - peg_diam/2) =< L_3z && (position_z[h] + peg_diam/2) >= L_4z && (position_x[h] - peg_diam/2) >=
L_1x)
        c=23;
    if ((position_z[h] - peg_diam/2) =< L_3z && (position_z[h] + peg_diam/2) >= L_4z && (position_x[h] - peg_diam/2) >=
L_1x_in && (position_x[h] - peg_diam/2) < L_1x)
        c=232;
    if ((position_z[h] - peg_diam/2) => L_3z && (position_x[h] + peg_diam/2) =< L_2x)
        c=31;
    if ((position_z[h] - peg_diam/2) >= L_3z && (position_x[h] - peg_diam/2) =< L_1x && (position_x[h] + peg_diam/2) =>
L_2x)
        c=32;
    if ((position_z[h] - peg_diam/2) >= L_3z_in && (position_z[h] - peg_diam/2) =< L_3z && (position_x[h] - peg_diam/2) =<
L_1x && (position_x[h] + peg_diam/2) => L_2x)
        c=322;
    if ((position_z[h] - peg_diam/2) >= L_3z && (position_x[h] - peg_diam/2) >= L_1x)
        c=33;

// for(var g=40; g<48; g+=1)

    if (abs(position_x[h] - hole_x[0]) =< (hole_diam-peg_diam) && abs(position_z[h] - hole_z[0]) =< (hole_diam-peg_diam) &&
(position_y[h] - peg_height - 40) <= 0)
        c=40;

    if (abs(position_x[h] - hole_x[1]) =< (hole_diam-peg_diam) && abs(position_z[h] - hole_z[1]) =< (hole_diam-peg_diam) &&
(position_y[h] - peg_height - 40) <= 0)
        c=41;
    if (abs(position_x[h] - hole_x[2]) =< (hole_diam-peg_diam) && abs(position_z[h] - hole_z[2]) =< (hole_diam-peg_diam) &&
(position_y[h] - peg_height - 40) <= 0)
        c=42;
    if (abs(position_x[h] - hole_x[3]) =< (hole_diam-peg_diam) && abs(position_z[h] - hole_z[3]) =< (hole_diam-peg_diam) &&
(position_y[h] - peg_height - 40) <= 0)
        c=43;
    if (abs(position_x[h] - hole_x[4]) =< (hole_diam-peg_diam) && abs(position_z[h] - hole_z[4]) =< (hole_diam-peg_diam) &&
(position_y[h] - peg_height - 40) <= 0)
        c=44;
    if (abs(position_x[h] - hole_x[5]) =< (hole_diam-peg_diam) && abs(position_z[h] - hole_z[5]) =< (hole_diam-peg_diam) &&
(position_y[h] - peg_height - 40) <= 0)
        c=45;
    if (abs(position_x[h] - hole_x[6]) =< (hole_diam-peg_diam) && abs(position_z[h] - hole_z[6]) =< (hole_diam-peg_diam) &&
(position_y[h] - peg_height - 40) <= 0)

```

```

c=46;
if (abs(position_x[h] - hole_x[7]) =< (hole_diam-peg_diam) && abs(position_z[h] - hole_z[7]) =< (hole_diam-peg_diam) &&
(position_y[h] - peg_height - 40) <= 0)
c=47;
if (abs(position_x[h] - hole_x[8]) =< (hole_diam-peg_diam) && abs(position_z[h] - hole_z[8]) =< (hole_diam-peg_diam) &&
(position_y[h] - peg_height - 40) <= 0)
c=48;

if(h_t == 0 && h_all != 1)
hole_diameter = 0;
else
hole_diameter = hole_diameter2;

if(h_all == 1 && (h_0outout != 1 || h_1outout != 1 || h_2outout != 1 || h_3outout != 1 || h_4outout != 1 || h_5outout !=
1 || h_6outout != 1 || h_7outout != 1 || h_8outout != 1))
hole_diameter = 15;

if (h_0outout == 1 || h_1outout == 1 || h_2outout == 1 || h_3outout == 1 || h_4outout == 1 || h_5outout == 1 || h_6outout
== 1 || h_7outout == 1 || h_8outout == 1)
hole_diameter = hole_diameter2;

if (h_0out!= 1 && (h_t != 0 || h_1out ==1 || h_2out == 1 || h_3out ==1 || h_4out == 1 || h_5out ==1 || h_6out == 1 ||
h_7out ==1 || h_8out ==1) && h_0==1 && h_0in != 1 && h_0outout!= 1 && h_0drop != 1 && (sqrt( (t_position_x[h] -
hole_x[0])^2 + (t_position_z[h] - hole_z[0])^2)) < 6.4 && (t_position_y[h]-peg_height) < (40+((peg_height - peak_tol)/2)))
{
col_0 = 1;
}
else
col_0 = 0;

if (h_0out!= 1 && (h_t != 0 || h_1out ==1 || h_2out == 1 || h_3out ==1 || h_4out == 1 || h_5out ==1 || h_6out == 1 ||
h_7out ==1 || h_8out ==1) && h_0==1 && h_0in != 1 && h_0outout!= 1 && h_0drop != 1 && (sqrt( (t_position_x[h] -
hole_x[0])^2 + (t_position_z[h] - hole_z[0])^2)) < 6.4 && (t_position_y[h]-peg_height) >= (40+((peg_height - peak_tol)/2))
&& (t_position_y[h]-peg_height) <= (40+((peg_height)/2)))
{
col_0 = 2;
}

if (h_1out != 1 && (h_t != 0 || h_0out == 1 || h_2out == 1 || h_3out ==1 || h_4out == 1 || h_5out ==1 || h_6out == 1 ||
h_7out ==1 || h_8out ==1) && h_1==1 && h_1in != 1 && h_1outout!= 1 && h_1drop != 1 && (sqrt( (t_position_x[h] -
hole_x[1])^2 + (t_position_z[h] - hole_z[1])^2)) < 6.4 && (t_position_y[h]-peg_height) < (40+((peg_height - peak_tol)/2)))
{
col_1 = 1;
}
else
col_1 = 0;

if (h_1out != 1 && (h_t != 0 || h_0out == 1 || h_2out == 1 || h_3out ==1 || h_4out == 1 || h_5out ==1 || h_6out == 1 ||
h_7out ==1 || h_8out ==1) && h_1==1 && h_1in != 1 && h_1outout!= 1 && h_1drop != 1 && (sqrt( (t_position_x[h] -
hole_x[1])^2 + (t_position_z[h] - hole_z[1])^2)) < 6.4 && (t_position_y[h]-peg_height) >= (40+((peg_height - peak_tol)/2))
&& (t_position_y[h]-peg_height) <= (40+((peg_height)/2)))
{

```

```

col_1 = 2;
}

if (h_2out != 1 && (h_t != 0 || h_0out == 1 || h_1out ==1 || h_3out ==1 || h_4out == 1 || h_5out ==1 || h_6out == 1 ||
h_7out ==1 || h_8out ==1) && h_2==1 && h_2in != 1 && h_2outout!= 1 && h_2drop != 1 && (sqrt( (t_position_x[h] -
hole_x[2])^2 + (t_position_z[h] - hole_z[2])^2)) < 6.4 && (t_position_y[h]-peg_height) < (40+((peg_height - peak_tol)/2)))
{
col_2 = 1;
}
else
col_2 = 0;

if (h_2out != 1 && (h_t != 0 || h_0out == 1 || h_1out ==1 || h_3out ==1 || h_4out == 1 || h_5out ==1 || h_6out == 1 ||
h_7out ==1 || h_8out ==1) && h_2==1 && h_2in != 1 && h_2outout!= 1 && h_2drop != 1 && (sqrt( (t_position_x[h] -
hole_x[2])^2 + (t_position_z[h] - hole_z[2])^2)) < 6.4 && (t_position_y[h]-peg_height) >= (40+((peg_height - peak_tol)/2))
&& (t_position_y[h]-peg_height) <= (40+((peg_height)/2)))
{
col_2 = 2;
}

if (h_3out!= 1 && (h_t != 0 || h_0out == 1 || h_1out ==1 || h_2out == 1 || h_4out == 1 || h_5out ==1 || h_6out == 1 ||
h_7out ==1 || h_8out ==1) && h_3==1 && h_3in != 1 && h_3outout!= 1 && h_3drop != 1 && (sqrt( (t_position_x[h] -
hole_x[3])^2 + (t_position_z[h] - hole_z[3])^2)) < 6.4 && (t_position_y[h]-peg_height) < (40+((peg_height - peak_tol)/2)))
{
col_3 = 1;
}
else
col_3 = 0;

if (h_3out!= 1 && (h_t != 0 || h_0out == 1 || h_1out ==1 || h_2out == 1 || h_4out == 1 || h_5out ==1 || h_6out == 1 ||
h_7out ==1 || h_8out ==1) && h_3==1 && h_3in != 1 && h_3outout!= 1 && h_3drop != 1 && (sqrt( (t_position_x[h] -
hole_x[3])^2 + (t_position_z[h] - hole_z[3])^2)) < 6.4 && (t_position_y[h]-peg_height) >= (40+((peg_height - peak_tol)/2))
&& (t_position_y[h]-peg_height) <= (40+((peg_height)/2)))
{
col_3 = 2;
}

if (h_4out != 1 && (h_t != 0 || h_0out == 1 || h_1out ==1 || h_2out == 1 || h_3out ==1 || h_5out ==1 || h_6out == 1 ||
h_7out ==1 || h_8out ==1) && h_4==1 && h_4in != 1 && h_4outout!= 1 && h_4drop != 1 && (sqrt( (t_position_x[h] -
hole_x[4])^2 + (t_position_z[h] - hole_z[4])^2)) < 6.4 && (t_position_y[h]-peg_height) < (40+((peg_height - peak_tol)/2)))
{
col_4 = 1;
}
else
col_4 = 0;

if (h_4out != 1 && (h_t != 0 || h_0out == 1 || h_1out ==1 || h_2out == 1 || h_3out ==1 || h_5out ==1 || h_6out == 1 ||
h_7out ==1 || h_8out ==1) && h_4==1 && h_4in != 1 && h_4outout!= 1 && h_4drop != 1 && (sqrt( (t_position_x[h] -
hole_x[4])^2 + (t_position_z[h] - hole_z[4])^2)) < 6.4 && (t_position_y[h]-peg_height) >= (40+((peg_height - peak_tol)/2))
&& (t_position_y[h]-peg_height) <= (40+((peg_height)/2)))
{
col_4 = 2;
}

```

```

if ( h_5out != 1 && (h_t != 0 || h_0out == 1 || h_1out ==1 || h_2out == 1 || h_3out ==1 || h_4out == 1 || h_6out == 1 ||
h_7out ==1 || h_8out ==1) && h_5==1 && h_5in != 1 && h_5outout!= 1 && h_5drop != 1 && (sqrt( (t_position_x[h] -
hole_x[5])^2 + (t_position_z[h] - hole_z[5])^2)) < 6.4 && (t_position_y[h]-peg_height) < (40+((peg_height - peak_tol)/2)))
{
col_5 = 1;
}
else
col_5 = 0;

if ( h_5out != 1 && (h_t != 0 || h_0out == 1 || h_1out ==1 || h_2out == 1 || h_3out ==1 || h_4out == 1 || h_6out == 1 ||
h_7out ==1 || h_8out ==1) && h_5==1 && h_5in != 1 && h_5outout!= 1 && h_5drop != 1 && (sqrt( (t_position_x[h] -
hole_x[5])^2 + (t_position_z[h] - hole_z[5])^2)) < 6.4 && (t_position_y[h]-peg_height) >= (40+((peg_height - peak_tol)/2))
&& (t_position_y[h]-peg_height) <= (40+((peg_height)/2)))
{
col_5 = 2;
}

if (h_6out!= 1 && (h_t != 0 || h_0out == 1 || h_1out ==1 || h_2out == 1 || h_3out ==1 || h_4out == 1 || h_5out ==1 ||
h_7out ==1 || h_8out ==1) && h_6==1 && h_6in != 1 && h_6outout!= 1 && h_6drop != 1 && (sqrt( (t_position_x[h] -
hole_x[6])^2 + (t_position_z[h] - hole_z[6])^2)) < 6.4 && (t_position_y[h]-peg_height) < (40+((peg_height - peak_tol)/2)))
{
col_6 = 1;
}
else
col_6 = 0;

if (h_6out!= 1 && (h_t != 0 || h_0out == 1 || h_1out ==1 || h_2out == 1 || h_3out ==1 || h_4out == 1 || h_5out ==1 ||
h_7out ==1 || h_8out ==1) && h_6==1 && h_6in != 1 && h_6outout!= 1 && h_6drop != 1 && (sqrt( (t_position_x[h] -
hole_x[6])^2 + (t_position_z[h] - hole_z[6])^2)) < 6.4 && (t_position_y[h]-peg_height) >= (40+((peg_height - peak_tol)/2))
&& (t_position_y[h]-peg_height) <= (40+((peg_height)/2)))
{
col_6 = 2;
}

if (h_7out!= 1 && (h_t != 0 || h_0out == 1 || h_1out ==1 || h_2out == 1 || h_3out ==1 || h_4out == 1 || h_5out ==1 ||
h_6out == 1 || h_8out ==1) && h_7==1 && h_7in != 1 && h_7outout!= 1 && h_7drop != 1 && (sqrt( (t_position_x[h] -
hole_x[7])^2 + (t_position_z[h] - hole_z[7])^2)) < 6.4 && (t_position_y[h]-peg_height) < (40+((peg_height - peak_tol)/2)))
{
col_7 = 1;
}
else
col_7 = 0;

if (h_7out!= 1 && (h_t != 0 || h_0out == 1 || h_1out ==1 || h_2out == 1 || h_3out ==1 || h_4out == 1 || h_5out ==1 ||
h_6out == 1 || h_8out ==1) && h_7==1 && h_7in != 1 && h_7outout!= 1 && h_7drop != 1 && (sqrt( (t_position_x[h] -
hole_x[7])^2 + (t_position_z[h] - hole_z[7])^2)) < 6.4 && (t_position_y[h]-peg_height) >= (40+((peg_height - peak_tol)/2))
&& (t_position_y[h]-peg_height) <= (40+((peg_height)/2)))
{
col_7 = 2;
}

if (h_8out!= 1 && (h_t != 0 || h_0out == 1 || h_1out ==1 || h_2out == 1 || h_3out ==1 || h_4out == 1 || h_5out ==1 ||
h_6out == 1 || h_7out ==1) && h_8==1 && h_8in != 1 && h_8outout!= 1 && h_8drop != 1 && (sqrt( (t_position_x[h] -
hole_x[8])^2 + (t_position_z[h] - hole_z[8])^2)) < 6.4 && (t_position_y[h]-peg_height) < (40+((peg_height - peak_tol)/2)))

```

```

{
col_8 = 1;
}
else
col_8 = 0;

if (h_8out!= 1 && (h_t != 0 || h_0out == 1 || h_1out ==1 || h_2out == 1 || h_3out ==1 || h_4out == 1 || h_5out ==1 ||
h_6out == 1 || h_7out ==1) && h_8==1 && h_8in != 1 && h_8outout!= 1 && h_8drop != 1 && (sqrt( (t_position_x[h] -
hole_x[8])^2 + (t_position_z[h] - hole_z[8])^2)) < 6.4 && (t_position_y[h]-peg_height) >= (40+((peg_height - peak_tol)/2))
&& (t_position_y[h]-peg_height) <= (40+((peg_height)/2)))
{
col_8 = 2;
}

if(h_all != 1)
{

if (h_0 != 1)
{
if ((sqrt( (t_position_x[h] - hole_x[0])^2 + (t_position_z[h] - hole_z[0])^2)) > hole_diameter2 - hole_diameter + (0.3))
{
ro_0 =5;
}
if ((sqrt( (t_position_x[h] - hole_x[0])^2 + (t_position_z[h] - hole_z[0])^2)) > hole_diameter2 - hole_diameter + (0.6))
{
ro_0 =4;
}
if ((sqrt( (t_position_x[h] - hole_x[0])^2 + (t_position_z[h] - hole_z[0])^2)) > hole_diameter2 - hole_diameter + (0.9))
{
ro_0 =3;
}
if ((sqrt( (t_position_x[h] - hole_x[0])^2 + (t_position_z[h] - hole_z[0])^2)) > hole_diameter2 - hole_diameter + (1.2))
{
ro_0 =2;
}
if ((sqrt( (t_position_x[h] - hole_x[0])^2 + (t_position_z[h] - hole_z[0])^2)) > hole_diameter2 - hole_diameter + (1.5))
{
ro_0 =1;
}
}

if (h_1 != 1)
{
if ((sqrt( (t_position_x[h] - hole_x[1])^2 + (t_position_z[h] - hole_z[1])^2)) > hole_diameter2 - hole_diameter + (0.3))
{
ro_1 =5;
}
if ((sqrt( (t_position_x[h] - hole_x[1])^2 + (t_position_z[h] - hole_z[1])^2)) > hole_diameter2 - hole_diameter + (0.6))
{
ro_1 =4;
}
if ((sqrt( (t_position_x[h] - hole_x[1])^2 + (t_position_z[h] - hole_z[1])^2)) > hole_diameter2 - hole_diameter + (0.9))

```

```

{
ro_1 =3;
}
if ((sqrt( (t_position_x[h] - hole_x[1])^2 + (t_position_z[h] - hole_z[1])^2)) > hole_diameter2 - hole_diameter + (1.2))
{
ro_1 =2;
}
if ((sqrt( (t_position_x[h] - hole_x[1])^2 + (t_position_z[h] - hole_z[1])^2)) > hole_diameter2 - hole_diameter + (1.5))
{
ro_1 =1;
}
}

if (h_2 != 1)
{
if ((sqrt( (t_position_x[h] - hole_x[2])^2 + (t_position_z[h] - hole_z[2])^2)) > hole_diameter2 - hole_diameter + (0.3))
{
ro_2 =5;
}
if ((sqrt( (t_position_x[h] - hole_x[2])^2 + (t_position_z[h] - hole_z[2])^2)) > hole_diameter2 - hole_diameter + (0.6))
{
ro_2 =4;
}
if ((sqrt( (t_position_x[h] - hole_x[2])^2 + (t_position_z[h] - hole_z[2])^2)) > hole_diameter2 - hole_diameter + (0.9))
{
ro_2 =3;
}
if ((sqrt( (t_position_x[h] - hole_x[2])^2 + (t_position_z[h] - hole_z[2])^2)) > hole_diameter2 - hole_diameter + (1.2))
{
ro_2 =2;
}
if ((sqrt( (t_position_x[h] - hole_x[2])^2 + (t_position_z[h] - hole_z[2])^2)) > hole_diameter2 - hole_diameter + (1.5))
{
ro_2 =1;
}
}

if (h_3 != 1)
{
if ((sqrt( (t_position_x[h] - hole_x[3])^2 + (t_position_z[h] - hole_z[3])^2)) > hole_diameter2 - hole_diameter + (0.3))
{
ro_3 =5;
}
if ((sqrt( (t_position_x[h] - hole_x[3])^2 + (t_position_z[h] - hole_z[3])^2)) > hole_diameter2 - hole_diameter + (0.6))
{
ro_3 =4;
}
if ((sqrt( (t_position_x[h] - hole_x[3])^2 + (t_position_z[h] - hole_z[3])^2)) > hole_diameter2 - hole_diameter + (0.9))
{
ro_3 =3;
}
if ((sqrt( (t_position_x[h] - hole_x[3])^2 + (t_position_z[h] - hole_z[3])^2)) > hole_diameter2 - hole_diameter + (1.2))
{

```

```

ro_3 =2;
}
if ((sqrt( (t_position_x[h] - hole_x[3])^2 + (t_position_z[h] - hole_z[3])^2)) > hole_diameter2 - hole_diameter + (1.5))
{
ro_3 =1;
}
}

if (h_4 != 1)
{
if ((sqrt( (t_position_x[h] - hole_x[4])^2 + (t_position_z[h] - hole_z[4])^2)) > hole_diameter2 - hole_diameter + (0.3))
{
ro_4 =5;
}
if ((sqrt( (t_position_x[h] - hole_x[4])^2 + (t_position_z[h] - hole_z[4])^2)) > hole_diameter2 - hole_diameter + (0.6))
{
ro_4 =4;
}
if ((sqrt( (t_position_x[h] - hole_x[4])^2 + (t_position_z[h] - hole_z[4])^2)) > hole_diameter2 - hole_diameter + (0.9))
{
ro_4 =3;
}
if ((sqrt( (t_position_x[h] - hole_x[4])^2 + (t_position_z[h] - hole_z[4])^2)) > hole_diameter2 - hole_diameter + (1.2))
{
ro_4 =2;
}
if ((sqrt( (t_position_x[h] - hole_x[4])^2 + (t_position_z[h] - hole_z[4])^2)) > hole_diameter2 - hole_diameter + (1.5))
{
ro_4 =1;
}
}
}

if (h_5 != 1)
{
if ((sqrt( (t_position_x[h] - hole_x[5])^2 + (t_position_z[h] - hole_z[5])^2)) > hole_diameter2 - hole_diameter + (0.3))
{
ro_5 =5;
}
if ((sqrt( (t_position_x[h] - hole_x[5])^2 + (t_position_z[h] - hole_z[5])^2)) > hole_diameter2 - hole_diameter + (0.6))
{
ro_5 =4;
}
if ((sqrt( (t_position_x[h] - hole_x[5])^2 + (t_position_z[h] - hole_z[5])^2)) > hole_diameter2 - hole_diameter + (0.9))
{
ro_5 =3;
}
if ((sqrt( (t_position_x[h] - hole_x[5])^2 + (t_position_z[h] - hole_z[5])^2)) > hole_diameter2 - hole_diameter + (1.2))
{
ro_5 =2;
}
if ((sqrt( (t_position_x[h] - hole_x[5])^2 + (t_position_z[h] - hole_z[5])^2)) > hole_diameter2 - hole_diameter + (1.5))
{
ro_5 =1;
}
}
}

```



```

}

if(h_6 != 1)
{
if ((sqrt( (t_position_x[h] - hole_x[6])^2 + (t_position_z[h] - hole_z[6])^2)) > hole_diameter2 - hole_diameter + (0.3))
{
ro_6 =5;
}
if ((sqrt( (t_position_x[h] - hole_x[6])^2 + (t_position_z[h] - hole_z[6])^2)) > hole_diameter2 - hole_diameter + (0.6))
{
ro_6 =4;
}
if ((sqrt( (t_position_x[h] - hole_x[6])^2 + (t_position_z[h] - hole_z[6])^2)) > hole_diameter2 - hole_diameter + (0.9))
{
ro_6 =3;
}
if ((sqrt( (t_position_x[h] - hole_x[6])^2 + (t_position_z[h] - hole_z[6])^2)) > hole_diameter2 - hole_diameter + (1.2))
{
ro_6 =2;
}
if ((sqrt( (t_position_x[h] - hole_x[6])^2 + (t_position_z[h] - hole_z[6])^2)) > hole_diameter2 - hole_diameter + (1.5))
{
ro_6 =1;
}
}

if(h_7 != 1)
{
if ((sqrt( (t_position_x[h] - hole_x[7])^2 + (t_position_z[h] - hole_z[7])^2)) > hole_diameter2 - hole_diameter + (0.3))
{
ro_7 =5;
}
if ((sqrt( (t_position_x[h] - hole_x[7])^2 + (t_position_z[h] - hole_z[7])^2)) > hole_diameter2 - hole_diameter + (0.6))
{
ro_7 =4;
}
if ((sqrt( (t_position_x[h] - hole_x[7])^2 + (t_position_z[h] - hole_z[7])^2)) > hole_diameter2 - hole_diameter + (0.9))
{
ro_7 =3;
}
if ((sqrt( (t_position_x[h] - hole_x[7])^2 + (t_position_z[h] - hole_z[7])^2)) > hole_diameter2 - hole_diameter + (1.2))
{
ro_7 =2;
}
if ((sqrt( (t_position_x[h] - hole_x[7])^2 + (t_position_z[h] - hole_z[7])^2)) > hole_diameter2 - hole_diameter + (1.5))
{
ro_7 =1;
}
}

if(h_8 != 1)
{
if ((sqrt( (t_position_x[h] - hole_x[8])^2 + (t_position_z[h] - hole_z[8])^2)) > hole_diameter2 - hole_diameter + (0.3))
{

```

```

ro_8 =5;
}
if ((sqrt( (t_position_x[h] - hole_x[8])^2 + (t_position_z[h] - hole_z[8])^2)) > hole_diameter2 - hole_diameter + (0.6))
{
ro_8 =4;
}
if ((sqrt( (t_position_x[h] - hole_x[8])^2 + (t_position_z[h] - hole_z[8])^2)) > hole_diameter2 - hole_diameter + (0.9))
{
ro_8 =3;
}
if ((sqrt( (t_position_x[h] - hole_x[8])^2 + (t_position_z[h] - hole_z[8])^2)) > hole_diameter2 - hole_diameter + (1.2))
{
ro_8 =2;
}
if ((sqrt( (t_position_x[h] - hole_x[8])^2 + (t_position_z[h] - hole_z[8])^2)) > hole_diameter2 - hole_diameter + (1.5))
{
ro_8 =1;
}
}

for(var g=40; g<=48; g+=1)
{
    if (c==g && b==11)

    {
        hole_diam = 10;

        fr_k = sqrt(h_toolView.tool.lastForce[0]^2+h_toolView.tool.lastForce[2]^2)*mk_h;
        // hole_diam = 6.4;
        position_x[h]=hole_x[g-40];
        position_y[h]=(position_y[h-1]* (2*m_peg/T^2 + (fr_k+dam_peg)/T) - position_y[h-2]*m_peg/T^2 + t_velocity_y[h]*dam_peg
+ t_position_y[h] * stif_peg - m_peg*9810) / (m_peg/T^2 + (fr_k+dam_peg)/T + stif_peg);
        position_z[h]=hole_z[g-40];
        peg.SetPosition(position_x[h], position_y[h] + y_cor, position_z[h]);
    }

}

for(var g=40; g<=48; g+=1)
{
    if (c==g && b==11 && (position_y[h]-peg_height)<=(board_y[h]+Ly/2-19))

    {

// hole_diam = 6.4;
position_x[h]=hole_x[g-40];
position_y[h]= peg_height + board_y[h] + Ly/2-19;
position_z[h]=hole_z[g-40];
peg.SetPosition(position_x[h], position_y[h] + y_cor, position_z[h]);

```

```

}
}

for (var ri=0; ri<10; ri+=0.05)
{
if (c != 40 /* && (r_0 == 0+ri) */ && (t_position_y[h] <= (board_y[h] + Ly/2 + peg_height - 3*abs(10-r_0)/(r_0+1))) &&
c==22)
{

fr_k = abs(h_toolView.tool.lastForce[1]*mk);
position_x[h]=( position_x[h-1]* (2*m_peg/T^2 + (fr_k+dam_peg)/T) - position_x[h-2]*m_peg/T^2 +
t_velocity_x[h]*(dam_peg) + t_position_x[h] * stif_peg) / (m_peg/T^2 + (fr_k+dam_peg)/T + stif_peg) ;
position_y[h]= board_y[h] + Ly/2 + peg_height - 3*abs((10-r_0)/(r_0+1));//(-(1/(m_peg/T^2 + dam_board/T + dam_peg/T +
stif_peg + stif_board))*(-position_y[h-1]*(2*m_peg/T^2 - dam_board/T - dam_peg/T) - position_y[h-2]*m_peg/T^2 +
t_position_y[h]*stif_peg + t_velocity_y[h]*dam_peg) - peg_height + board_y[h] + Ly/2);
position_z[h]=( position_z[h-1]* (2*m_peg/T^2 + (fr_k+dam_peg)/T) - position_z[h-2]*m_peg/T^2 +
t_velocity_z[h]*(dam_peg) + t_position_z[h] * stif_peg) / (m_peg/T^2 + (fr_k+dam_peg)/T + stif_peg) ;
peg.SetPosition(position_x[h], position_y[h] + y_cor, position_z[h]);

}
}

if(g!=40 && g!=41 && g!=42 && g!=43 && g!=44 && g!=45 && g!=46 && g!=47 && g!=48 && c!=40 && c!=41 && c!=42 && c!=43 &&
c!=44 && c!=45 && c!=46 && c!=47 && c!=48)
hole_diam = hole_diameter;

if (c==22 && b==11 && (r_0 == 11))
{

fr_k = abs(h_toolView.tool.lastForce[1]*mk);
position_x[h]=( position_x[h-1]* (2*m_peg/T^2 + (fr_k+dam_peg)/T) - position_x[h-2]*m_peg/T^2 +
t_velocity_x[h]*(dam_peg) + t_position_x[h] * stif_peg) / (m_peg/T^2 + (fr_k+dam_peg)/T + stif_peg) ;
position_y[h]= board_y[h] + Ly/2 + peg_height;//(-(1/(m_peg/T^2 + dam_board/T + dam_peg/T + stif_peg +
stif_board))*(-position_y[h-1]*(2*m_peg/T^2 - dam_board/T - dam_peg/T) - position_y[h-2]*m_peg/T^2 +
t_position_y[h]*stif_peg + t_velocity_y[h]*dam_peg) - peg_height + board_y[h] + Ly/2);
position_z[h]=( position_z[h-1]* (2*m_peg/T^2 + (fr_k+dam_peg)/T) - position_z[h-2]*m_peg/T^2 +
t_velocity_z[h]*(dam_peg) + t_position_z[h] * stif_peg) / (m_peg/T^2 + (fr_k+dam_peg)/T + stif_peg) ;
peg.SetPosition(position_x[h], position_y[h] + y_cor, position_z[h]);
}

if (c==22 && b==12)
{

fr_k = abs(h_toolView.tool.lastForce[1]*mk);
position_x[h]=( position_x[h-1]* (2*m_peg/T^2 + (fr_k+dam_peg)/T) - position_x[h-2]*m_peg/T^2 + t_velocity_x[h]*dam_peg
+ t_position_x[h] * stif_peg) / (m_peg/T^2 + (fr_k+dam_peg)/T + stif_peg);

```

```

    position_y[h]= board_y[h] - Ly/2;//((-1/(m_peg/T^2 + dam_board/T + dam_peg/T + stif_peg +
stif_board))*(-position_y[h-1]*(2*m_peg/T^2 - dam_board/T - dam_peg/T) - position_y[h-2]*m_peg/T^2 +
t_position_y[h]*stif_peg + t_velocity_y[h]*dam_peg) - peg_height + board_y[h] + Ly/2);
    position_z[h]=( position_z[h-1]* (2*m_peg/T^2 + (fr_k+dam_peg)/T) - position_z[h-2]*m_peg/T^2 + t_velocity_z[h]*dam_peg
+ t_position_z[h] * stif_peg) / (m_peg/T^2 + (fr_k+dam_peg)/T + stif_peg);
    peg.SetPosition(position_x[h], position_y[h] + y_cor, position_z[h]);
}

if (c==232 && b==11)
{

fr_k = abs(h_toolView.tool.lastForce[1]*mk);

    position_x[h]= board_x[h] + Lx/2 + peg_diam/2 + Lx;//((1/(m_peg/T^2 + dam_board/T + dam_peg/T + stif_peg +
stif_board))*(-position_x[h-1]*(2*m_peg/T^2 - dam_board/T - dam_peg/T) - position_x[h-2]*m_peg/T^2 +
t_position_x[h]*stif_peg + t_velocity_x[h]*dam_peg) + peg_diam/2 + board_x[h] + Lx/2);
    position_y[h]=( position_y[h-1]* (2*m_peg/T^2 + (fr_k+dam_peg)/T) - position_y[h-2]*m_peg/T^2 +
t_velocity_y[h]*dam_peg + t_position_y[h] * stif_peg) / (m_peg/T^2 + (fr_k+dam_peg)/T + stif_peg) ;
    position_z[h]=( position_z[h-1]* (2*m_peg/T^2 + (fr_k+dam_peg)/T) - position_z[h-2]*m_peg/T^2 +
t_velocity_z[h]*dam_peg + t_position_z[h] * stif_peg) / (m_peg/T^2 + (fr_k+dam_peg)/T + stif_peg) ;
    peg.SetPosition(position_x[h], position_y[h] + y_cor, position_z[h]);
}

if (c==322 && b==11)
{

fr_k = abs(h_toolView.tool.lastForce[1]*mk);
    position_x[h]=( position_x[h-1]* (2*m_peg/T^2 + (fr_k+dam_peg)/T) - position_x[h-2]*m_peg/T^2 +
t_velocity_x[h]*dam_peg + t_position_x[h] * stif_peg) / (m_peg/T^2 + (fr_k+dam_peg)/T + stif_peg) ;
    position_y[h]=( position_y[h-1]* (2*m_peg/T^2 + (fr_k+dam_peg)/T) - position_y[h-2]*m_peg/T^2 +
t_velocity_y[h]*dam_peg + t_position_y[h] * stif_peg) / (m_peg/T^2 + (fr_k+dam_peg)/T + stif_peg) ;
    position_z[h]=board_z[h] + Lz/2 + peg_diam/2;//((1/(m_peg/T^2 + dam_board/T + dam_peg/T + stif_peg +
stif_board))*(-position_z[h-1]*(2*m_peg/T^2 - dam_board/T - dam_peg/T) - position_z[h-2]*m_peg/T^2 +
t_position_z[h]*stif_peg + t_velocity_z[h]*dam_peg) + peg_diam/2 + board_z[h] + Lz/2);
    peg.SetPosition(position_x[h], position_y[h] + y_cor, position_z[h]);
}

if (c==122 && b==11)
{

fr_k = abs(h_toolView.tool.lastForce[1]*mk);
    position_x[h]=( position_x[h-1]* (2*m_peg/T^2 + (fr_k+dam_peg)/T) - position_x[h-2]*m_peg/T^2 +
t_velocity_x[h]*dam_peg + t_position_x[h] * stif_peg) / (m_peg/T^2 + (fr_k+dam_peg)/T + stif_peg) ;
    position_y[h]=( position_y[h-1]* (2*m_peg/T^2 + (fr_k+dam_peg)/T) - position_y[h-2]*m_peg/T^2 +
t_velocity_y[h]*dam_peg + t_position_y[h] * stif_peg) / (m_peg/T^2 + (fr_k+dam_peg)/T + stif_peg) ;
    position_z[h]=board_z[h] - Lz/2 - peg_diam/2;//((1/(m_peg/T^2 + dam_board/T + dam_peg/T + stif_peg +
stif_board))*(-position_z[h-1]*(2*m_peg/T^2 - dam_board/T - dam_peg/T) - position_z[h-2]*m_peg/T^2 +
t_position_z[h]*stif_peg + t_velocity_z[h]*dam_peg) - peg_diam/2 + board_z[h] - Lz/2);
    peg.SetPosition(position_x[h], position_y[h] + y_cor, position_z[h]);
}

if (c==212 && b==11)
{

```

```

    fr_k = abs(h_toolView.tool.lastForce[1]*mk);
    position_x[h]=board_x[h] - Lx/2 - peg_diam/2;//((1/(m_peg/T^2 + dam_board/T + dam_peg/T + stif_peg +
stif_board))*(-position_x[h-1]*(2*m_peg/T^2 - dam_board/T - dam_peg/T) - position_x[h-2]*m_peg/T^2 +
t_position_x[h]*stif_peg + t_velocity_x[h]*dam_peg) - peg_diam/2 + board_x[h] - Lx/2);
    position_y[h]=( position_y[h-1]* (2*m_peg/T^2 + (fr_k+dam_peg)/T) - position_y[h-2]*m_peg/T^2 +
t_velocity_y[h]*dam_peg + t_position_y[h] * stif_peg) / (m_peg/T^2 + (fr_k+dam_peg)/T + stif_peg) ;
    position_z[h]=( position_z[h-1]* (2*m_peg/T^2 + (fr_k+dam_peg)/T) - position_z[h-2]*m_peg/T^2 +
t_velocity_z[h]*dam_peg + t_position_z[h] * stif_peg) / (m_peg/T^2 + (fr_k+dam_peg)/T + stif_peg) ;
    peg.SetPosition(position_x[h], position_y[h] + y_cor, position_z[h]);
}

```

```

    if(col_0 == 1)
    {
//1st quadrant
if (t_position_x[h] > hole_x[0] && t_position_z[h] > hole_z[0])
{
theta = atan(abs((t_position_x[h] - hole_x[0])/(t_position_z[h] - hole_z[0])));
position_x[h] = hole_x[0] + sin(theta)*6.4;
position_z[h] = hole_z[0] + cos(theta)*6.4;
}
//2nd quadrant
if (t_position_x[h] > hole_x[0] && t_position_z[h] < hole_z[0])
{
theta = 3.14 - atan(abs((t_position_x[h] - hole_x[0])/(t_position_z[h] - hole_z[0])));
position_x[h] = hole_x[0] + cos(theta - (3.14/2))*6.4;
position_z[h] = hole_z[0] - sin(theta - (3.14/2))*6.4;
}
//3rd quadrant
if (t_position_x[h] < hole_x[0] && t_position_z[h] < hole_z[0])
{
theta = 3.14 + atan(abs((t_position_x[h] - hole_x[0])/(t_position_z[h] - hole_z[0])));
position_x[h] = hole_x[0] - sin(theta - 3.14)*6.4;
position_z[h] = hole_z[0] - cos(theta - 3.14)*6.4;
}
//4nd quadrant
if (t_position_x[h] < hole_x[0] && t_position_z[h] > hole_z[0])
{
theta = 3.14*2 - atan(abs((t_position_x[h] - hole_x[0])/(t_position_z[h] - hole_z[0])));
position_x[h] = hole_x[0] - cos(theta - (3.14 + (3.14/2)))*6.4;
position_z[h] = hole_z[0] + sin(theta - (3.14 + (3.14/2)))*6.4;
}

```

```

peg.SetPosition(position_x[h], position_y[h] + y_cor, position_z[h]);
}

```

```

    if(col_1 == 1)
    {
//1st quadrant
if (t_position_x[h] > hole_x[1] && t_position_z[h] > hole_z[1])
{
theta = atan(abs((t_position_x[h] - hole_x[1])/(t_position_z[h] - hole_z[1])));

```

```

position_x[h] = hole_x[1] + sin(theta)*6.4;
position_z[h] = hole_z[1] + cos(theta)*6.4;
}
//2nd quadrant
if (t_position_x[h] > hole_x[1] && t_position_z[h] < hole_z[1])
{
theta = 3.14 - atan(abs((t_position_x[h] - hole_x[1])/(t_position_z[h] - hole_z[1])));
position_x[h] = hole_x[1] + cos(theta - (3.14/2))*6.4;
position_z[h] = hole_z[1] - sin(theta - (3.14/2))*6.4;
}
//3rd quadrant
if (t_position_x[h] < hole_x[1] && t_position_z[h] < hole_z[1])
{
theta = 3.14 + atan(abs((t_position_x[h] - hole_x[1])/(t_position_z[h] - hole_z[1])));
position_x[h] = hole_x[1] - sin(theta - 3.14)*6.4;
position_z[h] = hole_z[1] - cos(theta - 3.14)*6.4;
}
//4nd quadrant
if (t_position_x[h] < hole_x[1] && t_position_z[h] > hole_z[1])
{
theta = 3.14*2 - atan(abs((t_position_x[h] - hole_x[1])/(t_position_z[h] - hole_z[1])));
position_x[h] = hole_x[1] - cos(theta - (3.14 + (3.14/2)))*6.4;
position_z[h] = hole_z[1] + sin(theta - (3.14 + (3.14/2)))*6.4;
}

peg.SetPosition(position_x[h], position_y[h] + y_cor, position_z[h]);
}

if(col_2 == 1)
{
//1st quadrant
if (t_position_x[h] > hole_x[2] && t_position_z[h] > hole_z[2])
{
theta = atan(abs((t_position_x[h] - hole_x[2])/(t_position_z[h] - hole_z[2])));
position_x[h] = hole_x[2] + sin(theta)*6.4;
position_z[h] = hole_z[2] + cos(theta)*6.4;
}
//2nd quadrant
if (t_position_x[h] > hole_x[2] && t_position_z[h] < hole_z[2])
{
theta = 3.14 - atan(abs((t_position_x[h] - hole_x[2])/(t_position_z[h] - hole_z[2])));
position_x[h] = hole_x[2] + cos(theta - (3.14/2))*6.4;
position_z[h] = hole_z[2] - sin(theta - (3.14/2))*6.4;
}
//3nd quadrant
if (t_position_x[h] < hole_x[2] && t_position_z[h] < hole_z[2])
{
theta = 3.14 + atan(abs((t_position_x[h] - hole_x[2])/(t_position_z[h] - hole_z[2])));
position_x[h] = hole_x[2] - sin(theta - 3.14)*6.4;
position_z[h] = hole_z[2] - cos(theta - 3.14)*6.4;
}
//4nd quadrant

```

```

if (t_position_x[h] < hole_x[2] && t_position_z[h] > hole_z[2])
{
theta = 3.14*2 - atan(abs((t_position_x[h] - hole_x[2])/(t_position_z[h] - hole_z[2])));
position_x[h] = hole_x[2] - cos(theta - (3.14 + (3.14/2)))*6.4;
position_z[h] = hole_z[2] + sin(theta - (3.14 + (3.14/2)))*6.4;
}

peg.SetPosition(position_x[h], position_y[h] + y_cor, position_z[h]);
}

if(col_3 == 1)
{
//1st quadrant
if (t_position_x[h] > hole_x[3] && t_position_z[h] > hole_z[3])
{
theta = atan(abs((t_position_x[h] - hole_x[3])/(t_position_z[h] - hole_z[3])));
position_x[h] = hole_x[3] + sin(theta)*6.4;
position_z[h] = hole_z[3] + cos(theta)*6.4;
}
//2nd quadrant
if (t_position_x[h] > hole_x[3] && t_position_z[h] < hole_z[3])
{
theta = 3.14 - atan(abs((t_position_x[h] - hole_x[3])/(t_position_z[h] - hole_z[3])));
position_x[h] = hole_x[3] + cos(theta - (3.14/2))*6.4;
position_z[h] = hole_z[3] - sin(theta - (3.14/2))*6.4;
}
//3rd quadrant
if (t_position_x[h] < hole_x[3] && t_position_z[h] < hole_z[3])
{
theta = 3.14 + atan(abs((t_position_x[h] - hole_x[3])/(t_position_z[h] - hole_z[3])));
position_x[h] = hole_x[3] - sin(theta - 3.14)*6.4;
position_z[h] = hole_z[3] - cos(theta - 3.14)*6.4;
}
//4nd quadrant
if (t_position_x[h] < hole_x[3] && t_position_z[h] > hole_z[3])
{
theta = 3.14*2 - atan(abs((t_position_x[h] - hole_x[3])/(t_position_z[h] - hole_z[3])));
position_x[h] = hole_x[3] - cos(theta - (3.14 + (3.14/2)))*6.4;
position_z[h] = hole_z[3] + sin(theta - (3.14 + (3.14/2)))*6.4;
}

peg.SetPosition(position_x[h], position_y[h] + y_cor, position_z[h]);
}

if(col_4 == 1)
{
//1st quadrant
if (t_position_x[h] > hole_x[4] && t_position_z[h] > hole_z[4])
{
theta = atan(abs((t_position_x[h] - hole_x[4])/(t_position_z[h] - hole_z[4])));

```

```

position_x[h] = hole_x[4] + sin(theta)*6.4;
position_z[h] = hole_z[4] + cos(theta)*6.4;
}
//2nd quadrant
if (t_position_x[h] > hole_x[4] && t_position_z[h] < hole_z[4])
{
theta = 3.14 - atan(abs((t_position_x[h] - hole_x[4])/(t_position_z[h] - hole_z[4])));
position_x[h] = hole_x[4] + cos(theta - (3.14/2))*6.4;
position_z[h] = hole_z[4] - sin(theta - (3.14/2))*6.4;
}
//3rd quadrant
if (t_position_x[h] < hole_x[4] && t_position_z[h] < hole_z[4])
{
theta = 3.14 + atan(abs((t_position_x[h] - hole_x[4])/(t_position_z[h] - hole_z[4])));
position_x[h] = hole_x[4] - sin(theta - 3.14)*6.4;
position_z[h] = hole_z[4] - cos(theta - 3.14)*6.4;
}
//4nd quadrant
if (t_position_x[h] < hole_x[4] && t_position_z[h] > hole_z[4])
{
theta = 3.14*2 - atan(abs((t_position_x[h] - hole_x[4])/(t_position_z[h] - hole_z[4])));
position_x[h] = hole_x[4] - cos(theta - (3.14 + (3.14/2)))*6.4;
position_z[h] = hole_z[4] + sin(theta - (3.14 + (3.14/2)))*6.4;
}

peg.SetPosition(position_x[h], position_y[h] + y_cor, position_z[h]);
}

if(col_5 == 1)
{
//1st quadrant
if (t_position_x[h] > hole_x[5] && t_position_z[h] > hole_z[5])
{
theta = atan(abs((t_position_x[h] - hole_x[5])/(t_position_z[h] - hole_z[5])));
position_x[h] = hole_x[5] + sin(theta)*6.4;
position_z[h] = hole_z[5] + cos(theta)*6.4;
}
//2nd quadrant
if (t_position_x[h] > hole_x[5] && t_position_z[h] < hole_z[5])
{
theta = 3.14 - atan(abs((t_position_x[h] - hole_x[5])/(t_position_z[h] - hole_z[5])));
position_x[h] = hole_x[5] + cos(theta - (3.14/2))*6.4;
position_z[h] = hole_z[5] - sin(theta - (3.14/2))*6.4;
}
//3nd quadrant
if (t_position_x[h] < hole_x[5] && t_position_z[h] < hole_z[5])
{
theta = 3.14 + atan(abs((t_position_x[h] - hole_x[5])/(t_position_z[h] - hole_z[5])));
position_x[h] = hole_x[5] - sin(theta - 3.14)*6.4;
position_z[h] = hole_z[5] - cos(theta - 3.14)*6.4;
}
//4nd quadrant
if (t_position_x[h] < hole_x[5] && t_position_z[h] > hole_z[5])

```



```

{
theta = 3.14*2 - atan(abs((t_position_x[h] - hole_x[5])/(t_position_z[h] - hole_z[5]]));
position_x[h] = hole_x[5] - cos(theta - (3.14 + (3.14/2)))*6.4;
position_z[h] = hole_z[5] + sin(theta - (3.14 + (3.14/2)))*6.4;
}

peg.SetPosition(position_x[h], position_y[h] + y_cor, position_z[h]);
}

if(col_6 == 1)
{
//1st quadrant
if (t_position_x[h] > hole_x[6] && t_position_z[h] > hole_z[6])
{
theta = atan(abs((t_position_x[h] - hole_x[6])/(t_position_z[h] - hole_z[6]]));
position_x[h] = hole_x[6] + sin(theta)*6.4;
position_z[h] = hole_z[6] + cos(theta)*6.4;
}
//2nd quadrant
if (t_position_x[h] > hole_x[6] && t_position_z[h] < hole_z[6])
{
theta = 3.14 - atan(abs((t_position_x[h] - hole_x[6])/(t_position_z[h] - hole_z[6]]));
position_x[h] = hole_x[6] + cos(theta - (3.14/2))*6.4;
position_z[h] = hole_z[6] - sin(theta - (3.14/2))*6.4;
}
//3rd quadrant
if (t_position_x[h] < hole_x[6] && t_position_z[h] < hole_z[6])
{
theta = 3.14 + atan(abs((t_position_x[h] - hole_x[6])/(t_position_z[h] - hole_z[6]]));
position_x[h] = hole_x[6] - sin(theta - 3.14)*6.4;
position_z[h] = hole_z[6] - cos(theta - 3.14)*6.4;
}
//4nd quadrant
if (t_position_x[h] < hole_x[6] && t_position_z[h] > hole_z[6])
{
theta = 3.14*2 - atan(abs((t_position_x[h] - hole_x[6])/(t_position_z[h] - hole_z[6]]));
position_x[h] = hole_x[6] - cos(theta - (3.14 + (3.14/2)))*6.4;
position_z[h] = hole_z[6] + sin(theta - (3.14 + (3.14/2)))*6.4;
}

peg.SetPosition(position_x[h], position_y[h] + y_cor, position_z[h]);
}

if(col_7 == 1)
{
//1st quadrant
if (t_position_x[h] > hole_x[7] && t_position_z[h] > hole_z[7])
{
theta = atan(abs((t_position_x[h] - hole_x[7])/(t_position_z[h] - hole_z[7]]));
position_x[h] = hole_x[7] + sin(theta)*6.4;
}
}

```

```

position_z[h] = hole_z[7] + cos(theta)*6.4;
}
//2nd quadrant
if (t_position_x[h] > hole_x[7] && t_position_z[h] < hole_z[7])
{
theta = 3.14 - atan(abs((t_position_x[h] - hole_x[7])/(t_position_z[h] - hole_z[7]]));
position_x[h] = hole_x[7] + cos(theta - (3.14/2))*6.4;
position_z[h] = hole_z[7] - sin(theta - (3.14/2))*6.4;
}
//3rd quadrant
if (t_position_x[h] < hole_x[7] && t_position_z[h] < hole_z[7])
{
theta = 3.14 + atan(abs((t_position_x[h] - hole_x[7])/(t_position_z[h] - hole_z[7]]));
position_x[h] = hole_x[7] - sin(theta - 3.14)*6.4;
position_z[h] = hole_z[7] - cos(theta - 3.14)*6.4;
}
//4nd quadrant
if (t_position_x[h] < hole_x[7] && t_position_z[h] > hole_z[7])
{
theta = 3.14*2 - atan(abs((t_position_x[h] - hole_x[7])/(t_position_z[h] - hole_z[7]]));
position_x[h] = hole_x[7] - cos(theta - (3.14 + (3.14/2)))*6.4;
position_z[h] = hole_z[7] + sin(theta - (3.14 + (3.14/2)))*6.4;
}

peg.SetPosition(position_x[h], position_y[h] + y_cor, position_z[h]);
}

if(col_8 == 1)
{
//1st quadrant
if (t_position_x[h] > hole_x[8] && t_position_z[h] > hole_z[8])
{
theta = atan(abs((t_position_x[h] - hole_x[8])/(t_position_z[h] - hole_z[8]]));
position_x[h] = hole_x[8] + sin(theta)*6.4;
position_z[h] = hole_z[8] + cos(theta)*6.4;
}
//2nd quadrant
if (t_position_x[h] > hole_x[8] && t_position_z[h] < hole_z[8])
{
theta = 3.14 - atan(abs((t_position_x[h] - hole_x[8])/(t_position_z[h] - hole_z[8]]));
position_x[h] = hole_x[8] + cos(theta - (3.14/2))*6.4;
position_z[h] = hole_z[8] - sin(theta - (3.14/2))*6.4;
}
//3nd quadrant
if (t_position_x[h] < hole_x[8] && t_position_z[h] < hole_z[8])
{
theta = 3.14 + atan(abs((t_position_x[h] - hole_x[8])/(t_position_z[h] - hole_z[8]]));
position_x[h] = hole_x[8] - sin(theta - 3.14)*6.4;
position_z[h] = hole_z[8] - cos(theta - 3.14)*6.4;
}
//4nd quadrant

```

```
if (t_position_x[h] < hole_x[8] && t_position_z[h] > hole_z[8])
{
theta = 3.14*2 - atan(abs((t_position_x[h] - hole_x[8])/(t_position_z[h] - hole_z[8])));
position_x[h] = hole_x[8] - cos(theta - (3.14 + (3.14/2)))*6.4;
position_z[h] = hole_z[8] + sin(theta - (3.14 + (3.14/2)))*6.4;
}
```

```
peg.SetPosition(position_x[h], position_y[h] + y_cor, position_z[h]);
}
```

```
if (col_0 == 2)
{
position_y[h] = 40 + peg_height/2 +peg_height;
peg.SetPosition(position_x[h], position_y[h] + y_cor, position_z[h]);
}
```

```
if (col_1 == 2)
{
position_y[h] = 40 + peg_height/2 +peg_height;
peg.SetPosition(position_x[h], position_y[h] + y_cor, position_z[h]);
}
```

```
if (col_2 == 2)
{
position_y[h] = 40 + peg_height/2 +peg_height;
peg.SetPosition(position_x[h], position_y[h] + y_cor, position_z[h]);
}
```

```
if (col_3 == 2)
{
position_y[h] = 40 + peg_height/2 +peg_height;
peg.SetPosition(position_x[h], position_y[h] + y_cor, position_z[h]);
}
```

```
if (col_4 == 2)
{
position_y[h] = 40 + peg_height/2 +peg_height;
peg.SetPosition(position_x[h], position_y[h] + y_cor, position_z[h]);
}
```

```
if (col_5 == 2)
{
position_y[h] = 40 + peg_height/2 +peg_height;
peg.SetPosition(position_x[h], position_y[h] + y_cor, position_z[h]);
}
```

```
if (col_6 == 2)
{
position_y[h] = 40 + peg_height/2 +peg_height;
peg.SetPosition(position_x[h], position_y[h] + y_cor, position_z[h]);
}
```

```
if (col_7 == 2)
```

```

{
position_y[h] = 40 + peg_height/2 +peg_height;
peg.SetPosition(position_x[h], position_y[h] + y_cor, position_z[h]);
}

if (col_8 == 2)
{
position_y[h] = 40 + peg_height/2 +peg_height;
peg.SetPosition(position_x[h], position_y[h] + y_cor, position_z[h]);
}

}
else
{
position_x[h] = position_x[h-1];
position_y[h] = position_y[h-1];
position_z[h] = position_z[h-1];
velocity_x[h] = velocity_x[h-1];
velocity_y[h] = velocity_y[h-1];
velocity_z[h] = velocity_z[h-1];
c=0;
peg.SetPosition(position_x[h], position_y[h] + y_cor, position_z[h]);

}

if(h_timerMode)
{
var tool1 = h_toolView.tool;

if (h_0 == 1 && h_0_count <= 2)
{

next_hole = 1;
peg_touch0 = 1;
root0 = 0.01;
epsilon0 = 100;
x_mjm_loc_test = 0;
touch0 = 0;
tau_start_stop = 0;
h_0_count = h_0_count + 1;
phid = 0;

count1 = count1 + 1;

tau_start_stop = 0;

if (h_0_count == 2)
{

```

```

k_plane0 = (RMS0/RMSmax)*k_plane;//2000*10^-3;//stif_peg;
k_plane3 = 0;//0.001;
k_time = (RMS1/RMSmax_time)*k_time;//stif_peg;

if (k_plane0 > k_plane)
{
k_plane0 = k_plane;
}

k_plane1 = k_plane0;
RMS0 = 0;

plane_dist_sum0 = 0;

S0_I_mjm_sum = 0;
RMS1 = 0;
h_h1 = 0;
h_h0 = 0;
}

}

if (h_1 == 1 && h_1_count <= 2)
{
next_hole = 2;
peg_touch0 = 1;
root0 = 0.01;
epsilon0 = 100;
x_mjm_loc_test = 0;
touch0 = 0;
h_1_count = h_1_count + 1;
phid = 0;
tau_start_stop = 0;

// outputLN(" ",k_plane0," ");

// outputLN(" ",k_plane0," ");
if (h_1_count == 2)
{

k_plane0 = (RMS0/RMSmax)*k_plane;//2000*10^-3;//stif_peg;
k_plane3 = 0;//0.001;
k_time = (RMS1/RMSmax_time)*k_time;//stif_peg;

if (k_plane0 > 2000*10^-3)
{
k_plane0 = 2000*10^-3;
}

k_plane2 = k_plane0;
RMS0 = 0;

```

```

        plane_dist_sum0 = 0;

        S0_I_mjm_sum = 0;
        RMS1 = 0;
        h_h1 = 0;
        h_h0 = 0;
    }

}

if (h_2 == 1 && h_2_count <= 2)
{
    next_hole = 3;
    peg_touch0 = 1;
    root0 = 0.01;
    epsilon0 = 100;
    x_mjm_loc_test = 0;
    touch0 = 0;
    h_2_count = h_2_count + 1;
    tau_start_stop = 0;
    phid = 0;

    if (h_2_count == 2)
    {
        k_plane0 = (RMS0/RMSmax)*k_plane;//2000*10^-3;//stif_peg;
        k_plane3 = 0;//0.001;
        k_time = (RMS1/RMSmax_time)*k_time;//stif_peg;

        if (k_plane0 > 2000*10^-3)
        {
            k_plane0 = 2000*10^-3;
        }

        k_plane33 = k_plane0;
        RMS0 = 0;

        plane_dist_sum0 = 0;

        S0_I_mjm_sum = 0;
        RMS1 = 0;
        h_h1 = 0;
        h_h0 = 0;

    }

}

if (h_3 == 1 && h_3_count <= 2)

```

```

{
next_hole = 4;
peg_touch0 = 1;
root0 = 0.01;
epsilon0 = 100;
x_mjm_loc_test = 0;
touch0 = 0;
h_3_count = h_3_count + 1;
phid = 0;
tau_start_stop = 0;

    if (h_3_count == 2)
    {

        k_plane0 = (RMS0/RMSmax)*k_plane;//2000*10^-3;//stif_peg;
        k_plane3 = 0;//0.001;
        k_time = (RMS1/RMSmax_time)*k_time;//stif_peg;

        if (k_plane0 > 2000*10^-3)
        {
            k_plane0 = 2000*10^-3;
        }

        k_plane4 = k_plane0;
        RMS0 = 0;
        plane_dist_sum0 = 0;
        S0_I_mjm_sum = 0;
        RMS1 = 0;
        h_h1 = 0;
        h_h0 = 0;
    }

}
if (h_4 == 1 && h_4_count <= 2)
{
next_hole = 5;
peg_touch0 = 1;
root0 = 0.01;
epsilon0 = 100;
x_mjm_loc_test = 0;
touch0 = 0;
phid = 0;
h_4_count = h_4_count + 1;
tau_start_stop = 0;

    if (h_4_count == 2)
    {
        k_plane0 = (RMS0/RMSmax)*k_plane;//2000*10^-3;//stif_peg;

```

```

    k_plane3 = 0;//0.001;
    k_time = (RMS1/RMSmax_time)*k_time;//stif_peg;

    if (k_plane0 > 2000*10^-3)
    {
        k_plane0 = 2000*10^-3;
    }
    k_plane5 = k_plane0;
    RMS0 = 0;
    plane_dist_sum0 = 0;
    S0_I_mjm_sum = 0;
    RMS1 = 0;
    h_h1 = 0;
    h_h0 = 0;

}

}
if (h_5 == 1 && h_5_count <= 2)
{
    next_hole = 6;
    peg_touch0 = 1;
    root0 = 0.01;
    epsilon0 = 100;
    x_mjm_loc_test = 0;
    touch0 = 0;
    phid = 0;
    h_5_count = h_5_count + 1;
    tau_start_stop = 0;

    if (h_5_count == 2)
    {
        k_plane0 = (RMS0/RMSmax)*k_plane;//2000*10^-3;//stif_peg;
        k_plane3 = 0;//0.001;
        k_time = (RMS1/RMSmax_time)*k_time;//stif_peg;
        if (k_plane0 > 2000*10^-3)
        {
            k_plane0 = 2000*10^-3;
        }

        k_plane6 = k_plane0;
        RMS0 = 0;

        plane_dist_sum0 = 0;
        S0_I_mjm_sum = 0;
        RMS1 = 0;

        h_h1 = 0;
        h_h0 = 0;
    }
}

```



```

    }

}

if (h_6 == 1 && h_6_count <= 2)
{
    next_hole = 7;
    peg_touch0 = 1;
    root0 = 0.01;
    epsilon0 = 100;
    x_mjm_loc_test = 0;
    touch0 = 0;
    phid = 0;
    h_6_count = h_6_count + 1;
    tau_start_stop = 0;

    if (h_6_count == 2)
    {
        k_plane0 = (RMS0/RMSmax)*k_plane;//2000*10^-3;//stif_peg;
        k_plane3 = 0;//0.001;
        k_time = (RMS1/RMSmax_time)*k_time;//stif_peg;

        if (k_plane0 > 2000*10^-3)
        {
            k_plane0 = 2000*10^-3;
        }

        k_plane7 = k_plane0;
        RMS0 = 0;
        plane_dist_sum0 = 0;
        S0_I_mjm_sum = 0;
        RMS1 = 0;

        h_h1 = 0;
        h_h0 = 0;
    }

}

if (h_7 == 1 && h_7_count <= 2)
{
    next_hole = 8;
    peg_touch0 = 1;
    root0 = 0.01;
    epsilon0 = 100;
    x_mjm_loc_test = 0;
    touch0 = 0;
    phid = 0;
    h_7_count = h_7_count + 1;
    tau_start_stop = 0;

```

```

k_plane0 = (RMS0/RMSmax)*k_plane;//2000*10^-3;//stif_peg;
k_plane3 = 0;//0.001;
k_time = (RMS1/RMSmax_time)*k_time;//stif_peg;

if (k_plane0 > 2000*10^-3)
{
k_plane0 = 2000*10^-3;
}
}

//forces from peg to hole

// trig
if(next_hole >= 0 && next_hole <= 8)
{
//1
if(next_hole == 0)
{
k_norm = 2000*10^-3;//stif_peg;
k_plane0 = k_plane;//2000*10^-3;//stif_peg;
k_plane3 = 0;//0.001;
k_time = 2000*10^-3;//stif_peg;
}

if (hold == 1 && peg_touch0 <= 100 && abs(peg.GetPosition[1]-peg_0.GetPosition[1] + tolerance/2) <=
(tolerance/2)*1.2)
//2
{
if (tau_start_stop == 0)
//3
{
tau_start = GetTime();
tau_start_stop = 1;
}
//3

counter = counter + 1;

x_1 = peg_0.GetPosition[0];
y_1 = peg_0.GetPosition[1] - 38;
z_1 = peg_0.GetPosition[2];

if (next_hole == 0)
{
x_2 = peg0.GetPosition[0];
y_2 = peg0.GetPosition[1]-(38/2);
z_2 = peg0.GetPosition[2];
}
}

```

```
peg_touch0 = peg_touch0 + 1;

}

if (next_hole == 1)
{

x_2 = peg1.GetPosition[0];
y_2 = peg1.GetPosition[1]-(38/2);
z_2 = peg1.GetPosition[2];
peg_touch0 = peg_touch0 + 1;

}

if (next_hole == 2)
{

x_2 = peg2.GetPosition[0];
y_2 = peg2.GetPosition[1]-(38/2);
z_2 = peg2.GetPosition[2];
peg_touch0 = peg_touch0 + 1;
}

if (next_hole == 3)
{

x_2 = peg3.GetPosition[0];
y_2 = peg3.GetPosition[1]-(38/2);
z_2 = peg3.GetPosition[2];
peg_touch0 = peg_touch0 + 1;
}

if (next_hole == 4)
{

x_2 = peg4.GetPosition[0];
y_2 = peg4.GetPosition[1]-(38/2);
z_2 = peg4.GetPosition[2];
peg_touch0 = peg_touch0 + 1;
}

if (next_hole == 5)
{

x_2 = peg5.GetPosition[0];
y_2 = peg5.GetPosition[1]-(38/2);
z_2 = peg5.GetPosition[2];
peg_touch0 = peg_touch0 + 1;
}

if (next_hole == 6)
{

x_2 = peg6.GetPosition[0];
```

```

y_2 = peg6.GetPosition[1]-(38/2);
z_2 = peg6.GetPosition[2];
peg_touch0 = peg_touch0 + 1;
}

if (next_hole == 7)
{

x_2 = peg7.GetPosition[0];
y_2 = peg7.GetPosition[1]-(38/2);
z_2 = peg7.GetPosition[2];
peg_touch0 = peg_touch0 + 1;
}

if (next_hole == 8)
{

x_2 = peg8.GetPosition[0];
y_2 = peg8.GetPosition[1]-(38/2);
z_2 = peg8.GetPosition[2];
peg_touch0 = peg_touch0 + 1;
}

x_1d = x_1;
y_1d = y_2;
z_1d = z_1;

while(phid <= 20 || phid >= 130 || phid == 90/* || rand_angle == rand_angle8*/) //4
{

rand_angle = rand(500);

phid = angles_vector[rand_angle];
phi = (phid*Pi)/180;
}

//4

L_1 = y_1 - y_2;
L_2 = L_1/tan(phi);
L_3 = sqrt((x_1d - x_2)^2 + (z_1d - z_2)^2);

L_4 = sqrt(L_3^2 - L_2^2);

Alpha3 = atan(L_2/L_4);

if (z_1 > z_2) //5
{
theta2 = atan((x_1 - x_2)*(z_1 - z_2)^-1) + pi;
}

//5

```

```

if (z_1 < z_2)
//6
{
theta2 = atan((x_1 - x_2)*(z_1 - z_2)^-1);
}
//6

theta2_low = atan(L_1/L_3);
thetad = (theta2*180)/3.14;

x_3 = x_2 - L_4*sin(theta2 - Alpha3);
z_3 = z_2 - L_4*cos(theta2 - Alpha3);
y_3 = y_2;

fitts_dist = sqrt((x_2 - x_1)^2+(z_2 - z_1)^2);
t_fitts = ( A_1_healthy + A_2_healthy* log((fitts_dist/6.4)+1)/log(2) )*1000;
t_fitts_MS = ( A_1_MS + A_2_MS* log((fitts_dist/6.4)+1)/log(2) )*1000;

AA = y_1*(z_2 - z_3) + y_2*(z_3 - z_1) + y_3*(z_1 - z_2);
BB = z_1*(x_2 - x_3) + z_2*(x_3 - x_1) + z_3*(x_1 - x_2);
CC = x_1*(y_2 - y_3) + x_2*(y_3 - y_1) + x_3*(y_1 - y_2);
DD = - x_1*(y_2*z_3 - y_3*z_2) - x_2*(y_3*z_1 - y_1*z_3) - x_3*(y_1*z_2 - y_2*z_1);

// Plane normal
xn = AA/sqrt(AA^2 + BB^2 + CC^2);
yn = BB/sqrt(AA^2 + BB^2 + CC^2);
zn = CC/sqrt(AA^2 + BB^2 + CC^2);

// Define rotation axes

// Along '1' - '3'
blx = (x_1 - x_3)/ sqrt((x_1 - x_3)^2+(y_1 - y_3)^2+(z_1 - z_3)^2);
bly = (y_1 - y_3)/ sqrt((x_1 - x_3)^2+(y_1 - y_3)^2+(z_1 - z_3)^2);
blz = (z_1 - z_3)/ sqrt((x_1 - x_3)^2+(y_1 - y_3)^2+(z_1 - z_3)^2);

// Along '2' - '3' line
b2x = (x_2 - x_3)/ sqrt((x_2 - x_3)^2+(y_2 - y_3)^2+(z_2 - z_3)^2);
b2y = (y_2 - y_3)/ sqrt((x_2 - x_3)^2+(y_2 - y_3)^2+(z_2 - z_3)^2);
b2z = (z_2 - z_3)/ sqrt((x_2 - x_3)^2+(y_2 - y_3)^2+(z_2 - z_3)^2);

r_23[0] = (x_2 - x_3);
r_23[1] = (y_2 - y_3);
r_23[2] = (z_2 - z_3);

r_13[0] = (x_1 - x_3);
r_13[1] = (y_1 - y_3);
r_13[2] = (z_1 - z_3);

check_dot_12_23 = r_23[0]*r_13[0] + r_23[1]*r_13[1] + r_23[2]*r_13[2];

// Perpendicular to the plane

```

```

cross_13_23[0] = (r_13[1]*r_23[2] - r_13[2]*r_23[1])/sqrt((r_13[1]*r_23[2] - r_13[2]*r_23[1])^2+(r_13[2]*r_23[0] -
r_13[0]*r_23[2])^2+(r_13[0]*r_23[1] - r_13[1]*r_23[0])^2);
cross_13_23[1] = (r_13[2]*r_23[0] - r_13[0]*r_23[2])/sqrt((r_13[1]*r_23[2] - r_13[2]*r_23[1])^2+(r_13[2]*r_23[0] -
r_13[0]*r_23[2])^2+(r_13[0]*r_23[1] - r_13[1]*r_23[0])^2);
cross_13_23[2] = (r_13[0]*r_23[1] - r_13[1]*r_23[0])/sqrt((r_13[1]*r_23[2] - r_13[2]*r_23[1])^2+(r_13[2]*r_23[0] -
r_13[0]*r_23[2])^2+(r_13[0]*r_23[1] - r_13[1]*r_23[0])^2);

b3x = cross_13_23[0];
b3y = cross_13_23[1];
b3z = cross_13_23[2];

b_1 = [b1x, b1y, b1z];
b_2 = [b2x, b2y, b2z];
b_3 = [b3x, b3y, b3z];

X_1loc[0] = b1x*(x_1 - x_3) + b1y*(y_1 - y_3) + b1z*(z_1 - z_3);
X_1loc[1] = b2x*(x_1 - x_3) + b2y*(y_1 - y_3) + b2z*(z_1 - z_3);
X_1loc[2] = b3x*(x_1 - x_3) + b3y*(y_1 - y_3) + b3z*(z_1 - z_3);

X_2loc[0] = b1x*(x_2 - x_3) + b1y*(y_2 - y_3) + b1z*(z_2 - z_3);
X_2loc[1] = b2x*(x_2 - x_3) + b2y*(y_2 - y_3) + b2z*(z_2 - z_3);
X_2loc[2] = b3x*(x_2 - x_3) + b3y*(y_2 - y_3) + b3z*(z_2 - z_3);

// point three should be (0,0,0)
X_3loc[0] = b1x*(x_3 - x_3) + b1y*(y_3 - y_3) + b1z*(z_3 - z_3);
X_3loc[1] = b2x*(x_3 - x_3) + b2y*(y_3 - y_3) + b2z*(z_3 - z_3);
X_3loc[2] = b3x*(x_3 - x_3) + b3y*(y_3 - y_3) + b3z*(z_3 - z_3);

dxi = ((0.58-0.61)*(X_1loc[0]/0.6))/0.1;

dyi = ((0.03)*(X_2loc[1]/1))/0.1;

dyf = ((0.01)*(X_2loc[1]/1))/0.1;

dx = ((0-0.2)*(X_1loc[0]/0.6))/0.1;

xi = x_1loc[0];
xf = 0;
mjm_a0 = xi;
mjm_a1 = dxi;
mjm_a2 = 0;
mjm_a3 = -2*(5*xi - 5*xf + 3*dxi + 2*dx);
mjm_a4 = (15*xi - 15*xf + 8*dxi + 7*dx);
mjm_a5 = -3*(2*xi - 2*xf + dxi + dx);

yi = 0;
yf = x_2loc[1];
mjm_b0 = yi;
mjm_b1 = dyi;
mjm_b2 = 0;
mjm_b3 = -2*(5*yi - 5*yf + 3*dyi + 2*dyf);
mjm_b4 = (15*yi - 15*yf + 8*dyi + 7*dyf);
mjm_b5 = -3*(2*yi - 2*yf + dyi + dyf);

```

```

for (i_mjm_draw = 0; i_mjm_draw <= draw_points; i_mjm_draw+=1)
{
    tau_draw = i_mjm_draw*draw_points^-1;
    x_mjm_draw[i_mjm_draw] = mjm_a5*tau_draw^5 + mjm_a4*tau_draw^4 + mjm_a3*tau_draw^3 + mjm_a2*tau_draw^2 +
mjm_a1*tau_draw + mjm_a0;
    y_mjm_draw[i_mjm_draw] = mjm_b5*tau_draw^5 + mjm_b4*tau_draw^4 + mjm_b3*tau_draw^3 + mjm_b2*tau_draw^2 +
mjm_b1*tau_draw + mjm_b0;
}
//7

real_rand = rand(100);
}
//2

if (hold == 1)
{
    Check10 = AA*x_3 + BB*y_3 + CC*z_3 + DD;
    x_H = t_position_x[h];
    y_H = t_position_y[h]-38;
    z_H = t_position_z[h];

    X_Hloc[0] = b1x*(x_H - x_3) + b1y*(y_H - y_3) + b1z*(z_H - z_3);
    X_Hloc[1] = b2x*(x_H - x_3) + b2y*(y_H - y_3) + b2z*(z_H - z_3);
    X_Hloc[2] = b3x*(x_H - x_3) + b3y*(y_H - y_3) + b3z*(z_H - z_3);

    xH_loc = X_Hloc[0];
    yH_loc = X_Hloc[1];
    zH_loc = X_Hloc[2];

    // this is xHI in the matlab code
    x_I = (-DD * xn + BB * yn * x_H + CC * zn * x_H - BB * xn * y_H - CC * xn * z_H)/(AA*xn + BB * yn + CC * zn);
    y_I = (-DD * yn - AA * yn * x_H + AA * xn * y_H + CC * zn * y_H - CC * yn * z_H)/(AA*xn + BB * yn + CC * zn);
    z_I = (-DD * zn - AA * zn * x_H - BB * zn * y_H + AA * xn * z_H + BB * yn * z_H)/(AA*xn + BB * yn + CC * zn);

    // this is xHI_loc in the matlab code
    X_Iloc[0] = b1x*(x_I - x_3) + b1y*(y_I - y_3) + b1z*(z_I - z_3);
    X_Iloc[1] = b2x*(x_I - x_3) + b2y*(y_I - y_3) + b2z*(z_I - z_3);
    X_Iloc[2] = b3x*(x_I - x_3) + b3y*(y_I - y_3) + b3z*(z_I - z_3);

    if (hold == 1)
    {
        catch0 = catch0 + 1;
    }
}
//9

if (catch0 == 1)
}
//10

```

```

{
inst_vel_x = t_velocity_x[h];
inst_vel_y = t_velocity_y[h];
inst_vel_z = t_velocity_z[h];
}

//10

dxi_loc[0] = b1x*inst_vel_x + b1y*inst_vel_y + b1z*inst_vel_z;
dxi_loc[1] = b2x*inst_vel_x + b2y*inst_vel_y + b2z*inst_vel_z;
dxi_loc[2] = b3x*inst_vel_x + b3y*inst_vel_y + b3z*inst_vel_z;

xHI_loc = X_Iloc[0];
yHI_loc = X_Iloc[1];
zHI_loc = X_Iloc[2];

tau[hn] = root0;

if (h > 100)
{
while(hn <= 40 && epsilon0 >= 10^-4)
{
fn = (10*mjm_a5^2 + 10*mjm_b5^2)*tau[hn]^9 + (18*mjm_a4*mjm_a5 + 18*mjm_b4*mjm_b5)*tau[hn]^8 + (8*mjm_a4^2 +
16*mjm_a3*mjm_a5 + 8*mjm_b4^2 + 16*mjm_b3*mjm_b5)*tau[hn]^7 + (14*mjm_a3*mjm_a4 + 14*mjm_a2*mjm_a5 + 14*mjm_b3*mjm_b4 +
14*mjm_b2*mjm_b5)*tau[hn]^6 + (6*mjm_a3^2 + 12*mjm_a2*mjm_a4 + 12*mjm_a1*mjm_a5 + 6*mjm_b3^2 + 12*mjm_b2*mjm_b4 +
12*mjm_b1*mjm_b5)*tau[hn]^5 + (10*mjm_a2*mjm_a3 + 10*mjm_a1*mjm_a4 + 10*mjm_a0*mjm_a5 + 10*mjm_b2*mjm_b3 +
10*mjm_b1*mjm_b4 + 10*mjm_b0*mjm_b5 - 10*mjm_a5*xHI_loc - 10*mjm_b5*yHI_loc)*tau[hn]^4 + (4*mjm_a2^2 + 8*mjm_a1*mjm_a3 +
8*mjm_a0*mjm_a4 + 4*mjm_b2^2 + 8*mjm_b1*mjm_b3 + 8*mjm_b0*mjm_b4 - 8*mjm_a4*xHI_loc - 8*mjm_b4*yHI_loc)*tau[hn]^3 +
(6*mjm_a1*mjm_a2 + 6*mjm_a0*mjm_a3 + 6*mjm_b1*mjm_b2 + 6*mjm_b0*mjm_b3 - 6*mjm_a3*xHI_loc - 6*mjm_b3*yHI_loc)*tau[hn]^2 +
(2*mjm_a1^2 + 4*mjm_a0*mjm_a2 + 2*mjm_b1^2 + 4*mjm_b0*mjm_b2 - 4*mjm_a2*xHI_loc - 4*mjm_b2*yHI_loc)*tau[hn] +
(2*mjm_a0*mjm_a1 + 2*mjm_b0*mjm_b1 - 2*mjm_a1*xHI_loc - 2*mjm_b1*yHI_loc);

dfn = (9*(10*mjm_a5^2 + 10*mjm_b5^2))*tau[hn]^8 + (8*(18*mjm_a4*mjm_a5 + 18*mjm_b4*mjm_b5))*tau[hn]^7 +
(7*(8*mjm_a4^2 + 16*mjm_a3*mjm_a5 + 8*mjm_b4^2 + 16*mjm_b3*mjm_b5))*tau[hn]^6 + (6*(14*mjm_a3*mjm_a4 + 14*mjm_a2*mjm_a5 +
14*mjm_b3*mjm_b4 + 14*mjm_b2*mjm_b5))*tau[hn]^5 + (5*(6*mjm_a3^2 + 12*mjm_a2*mjm_a4 + 12*mjm_a1*mjm_a5 + 6*mjm_b3^2 +
12*mjm_b2*mjm_b4 + 12*mjm_b1*mjm_b5))*tau[hn]^4 + (4*(10*mjm_a2*mjm_a3 + 10*mjm_a1*mjm_a4 + 10*mjm_a0*mjm_a5 +
10*mjm_b2*mjm_b3 + 10*mjm_b1*mjm_b4 + 10*mjm_b0*mjm_b5 - 10*mjm_a5*xHI_loc - 10*mjm_b5*yHI_loc))*tau[hn]^3 +
(3*(4*mjm_a2^2 + 8*mjm_a1*mjm_a3 + 8*mjm_a0*mjm_a4 + 4*mjm_b2^2 + 8*mjm_b1*mjm_b3 + 8*mjm_b0*mjm_b4 - 8*mjm_a4*xHI_loc -
8*mjm_b4*yHI_loc))*tau[hn]^2 + (2*(6*mjm_a1*mjm_a2 + 6*mjm_a0*mjm_a3 + 6*mjm_b1*mjm_b2 + 6*mjm_b0*mjm_b3 -
6*mjm_a3*xHI_loc - 6*mjm_b3*yHI_loc))*tau[hn] + (2*mjm_a1^2 + 4*mjm_a0*mjm_a2 + 2*mjm_b1^2 + 4*mjm_b0*mjm_b2 -
4*mjm_a2*xHI_loc - 4*mjm_b2*yHI_loc);

tau[hn+1] = tau[hn] - fn/dfn;

epsilon0 = abs(tau[hn+1]-tau[hn]) * abs(tau[hn])^-1;

hn = hn + 1;
}
}
}

```



```

}
//12

count0 = count0 + 1;

root0 = tau[hn-1];

if (root0 < 0)
    root0 = 0;

if (hn ==> 2)
//13
{
    delta_tau = root0 - tau[hn-2];
}
//13

    hn = 0;
    epsilon0 = 100;
}
//11

xP_loc = mjm_a5*root0^5 + mjm_a4*root0^4 + mjm_a3*root0^3 + mjm_a2*root0^2 + mjm_a1*root0 + mjm_a0;
yP_loc = mjm_b5*root0^5 + mjm_b4*root0^4 + mjm_b3*root0^3 + mjm_b2*root0^2 + mjm_b1*root0 + mjm_b0;
zP_loc = 0;

xP = b1x*(xP_loc) + b2x*(yP_loc) + b3x*(zP_loc) + x_3;
yP = b1y*(xP_loc) + b2y*(yP_loc) + b3y*(zP_loc) + y_3;
zP = b1z*(xP_loc) + b2z*(yP_loc) + b3z*(zP_loc) + z_3;

if (x_mjm_loc_test == 0)
//14
{
    tau0 = (GetTime() - tau_start)*t_fitts^-1;

    if (tau0 > 1)
//15
    {
        tau0 = 1;
    }
//15

    x_mjm_loc = mjm_a5*tau0^5 + mjm_a4*tau0^4 + mjm_a3*tau0^3 + mjm_a2*tau0^2 + mjm_a1*tau0 + mjm_a0;
    y_mjm_loc = mjm_b5*tau0^5 + mjm_b4*tau0^4 + mjm_b3*tau0^3 + mjm_b2*tau0^2 + mjm_b1*tau0 + mjm_b0;
    z_mjm_loc = 0;

    x_mjm = b1x*(x_mjm_loc) + b2x*(y_mjm_loc) + b3x*(z_mjm_loc) + x_3;
    y_mjm = b1y*(x_mjm_loc) + b2y*(y_mjm_loc) + b3y*(z_mjm_loc) + y_3;
    z_mjm = b1z*(x_mjm_loc) + b2z*(y_mjm_loc) + b3z*(z_mjm_loc) + z_3;

    if (x_mjm_loc <= 0)
//16
    {
        x_mjm_loc_test = 1;
    }
}

```

```

}
//16

}
//14

if (x_mjm_loc_test_MS == 0)
//17
{
tau0_MS = (GetTime() - tau_start)*t_fitts_MS^-1;
x_mjm_loc_MS = mjm_a5*tau0_MS^5 + mjm_a4*tau0_MS^4 + mjm_a3*tau0_MS^3 + mjm_a2*tau0_MS^2 + mjm_a1*tau0_MS +
mjm_a0;
y_mjm_loc_MS = mjm_b5*tau0_MS^5 + mjm_b4*tau0_MS^4 + mjm_b3*tau0_MS^3 + mjm_b2*tau0_MS^2 + mjm_b1*tau0_MS +
mjm_b0;
z_mjm_loc_MS = 0;

x_mjm_MS = b1x*(x_mjm_loc_MS) + b2x*(y_mjm_loc_MS) + b3x*(z_mjm_loc_MS) + x_3;
y_mjm_MS = b1y*(x_mjm_loc_MS) + b2y*(y_mjm_loc_MS) + b3y*(z_mjm_loc_MS) + y_3;
z_mjm_MS = b1z*(x_mjm_loc_MS) + b2z*(y_mjm_loc_MS) + b3z*(z_mjm_loc_MS) + z_3;

if (x_mjm_loc_MS <= 0)
//18
{
x_mjm_loc_test_MS = 1;
}
//18

}
//17

norm_dist = sqrt((DD + AA * x_H + BB * y_H + CC * z_H)^2/(AA^2 + BB^2 + CC^2));
D_norm_dist = norm_dist*(AA * vxH + BB * vyH + CC * vzH) / (DD + AA*x_H + BB * y_H + CC * z_H);

vxHI_loc = b1x*vxH + b1y*vyH + b1z*vzH;
vyHI_loc = b2x*vxH + b2y*vyH + b2z*vzH;
vzHI_loc = b3x*vxH + b3y*vyH + b3z*vzH;

vH_mag = sqrt((vxHI_loc)^2+(vyHI_loc)^2+(vzHI_loc)^2);

delta_time = GetTime() - prev_time;

plane_dist = sqrt((xP_loc - xHI_loc)^2 + (yP_loc - yHI_loc)^2);
plane_dist_vec[h_h0] = abs(plane_dist);

if (delta_time == 0)
//19
{
delta_time = 1;
}
//19

D_plane_dist = ((mjm_a1 + 3*mjm_a3*root0^2 + 4*mjm_a4*root0^3 + 5*mjm_a5*root0^4)*(xP_loc -
X_Hloc[0])*delta_tau*delta_time^-1 +
(mjm_b1 + 3*mjm_b3*root0^2 + 4*mjm_b4*root0^3 + 5*mjm_b5*root0^4)*(yP_loc - X_Hloc[1])*delta_tau*delta_time^-1 -

```

```

(xP_loc - X_Hloc[0])*vxHI_loc - (yP_loc - X_Hloc[1])*vyHI_loc)*(2*sqrt((xP_loc - X_Hloc[0])^2 + (yP_loc -
X_Hloc[1])^2))^-1;

plane_dist_sum0 = plane_dist_vec[h_h0] + plane_dist_sum0;

if (touch0 != 1)
{
RMS0 = plane_dist_sum0/(h_h0 + 1);
}

plane_dist_av_max = RMS0*1;

k_plane_max = (F_max)/plane_dist_av_max;

h_h0 = h_h0 + 1;

itavec[0] = (xP_loc - xHI_loc)*plane_dist^-1;
itavec[1] = (yP_loc - yHI_loc)*plane_dist^-1;
itavec[2] = 0;

lamvec[1] = 1;
lamvec[0] = -itavec[1]*itavec[0]^-1;
lamvec[2] = 0;

x1_int = xP_loc;
y1_int = yP_loc;

S0_I_mjm = 0;

for (iS = 1; iS<=3; iS+=1)
//20
{
tau0_int = root0 + (tau0 - root0)/3;

x2_int = mjm_a5*tau0_int^5 + mjm_a4*tau0_int^4 + mjm_a3*tau0_int^3 + mjm_a2*tau0_int^2 + mjm_a1*tau0_int +
mjm_a0;
y2_int = mjm_b5*tau0_int^5 + mjm_b4*tau0_int^4 + mjm_b3*tau0_int^3 + mjm_b2*tau0_int^2 + mjm_b1*tau0_int +
mjm_b0;

S0_I_mjm = S0_I_mjm + sqrt((x2_int - x1_int)^2+(y2_int - y1_int)^2);

S0_I_mjm_vec[h_h1] = abs(S0_I_mjm);

S0_I_mjm_sum = S0_I_mjm_vec[h_h1] + S0_I_mjm_sum;

if (touch0 != 1)
{
RMS1 = S0_I_mjm_sum/(h_h1 + 1);
}

```

```

S0_I_mjm_av_max = RMS1*1;

k_time_max = (F_max)/S0_I_mjm_av_max;

h_h1 = h_h1 + 1;

x1_int = x2_int;
y1_int = y2_int;

}

//20

S0_I_mjm = S0_I_mjm * (tau0 - root0)/abs(tau0 - root0);

D_dist_mjm = ((S0_I_mjm - S0_I_mjm_prev)*delta_time^-1)*(delta_tau*delta_time)^-1;

// outputLN(RMS0,"",RMS1,"",next_hole);
prev_time = GetTime();
S0_I_mjm_prev = S0_I_mjm;

// k_plane0 = 0;//k_plane;//2000*10^-3;//stif_peg;
// k_plane3 = 0;//0.001;
// k_time = 0;//2000*10^-3;//stif_peg;

k_norm = 2000*10^-3;
b_norm = 3*(k_norm*10^-3);
b_plane = 3*(k_plane0*10^-3);
b_mjm = 3*(k_time*10^-3);

// if (tau0 < root0 && tau0 >= 0.2)

// {
// k_time = 0;
// }

//21

//21

F_time_x_d = - 0;//(delta_tau*delta_time^-1)*b_mjm*D_dist_mjm*((yH_loc - yH_loc_prev)*abs(yH_loc -
yH_loc_prev)^-1)*lamvec[0]*sqrt(lamvec[0]^2 + lamvec[1]^2 + lamvec[2]^2)^-1;
F_time_y_d = 0;//- (delta_tau*delta_time^-1)*b_mjm*D_dist_mjm*((yH_loc - yH_loc_prev)*abs(yH_loc -
yH_loc_prev)^-1)*lamvec[1]*sqrt(lamvec[0]^2 + lamvec[1]^2 + lamvec[2]^2)^-1;
F_time_z_d = 0;

```

```

F_time_x_d = 0; //(delta_tau*delta_time^-1)*b_mjm*D_dist_mjm*lamvec[0]*sqrt(lamvec[0]^2 + lamvec[1]^2 +
lamvec[2]^2)^-1;
F_time_y_d = 0; //(delta_tau*delta_time^-1)*b_mjm*D_dist_mjm*lamvec[1]*sqrt(lamvec[0]^2 + lamvec[1]^2 +
lamvec[2]^2)^-1;
F_time_z_d = 0;

if (yH_loc_prev == yH_loc)
//22
{
F_time_x_d = 0;
F_time_y_d = 0;
F_time_z_d = 0;
}
//22

F_plane_x_d = 0; //b_plane*D_plane_dist*itavec[0];
F_plane_y_d = 0; //b_plane*D_plane_dist*itavec[1];
F_plane_z_d = 0;

F_norm_x_d = 0;
F_norm_y_d = 0;
F_norm_z_d = - b_norm*D_norm_dist*(zH_loc*abs(zH_loc)^-1);

F_time_x_k = 0; //(k_time)*(S0_I_mjm)*lamvec[0]/sqrt(lamvec[0]^2 + lamvec[1]^2 + lamvec[2]^2);
F_time_y_k = 0; //(k_time)*(S0_I_mjm)*lamvec[1]/sqrt(lamvec[0]^2 + lamvec[1]^2 + lamvec[2]^2);
F_time_z_k = 0; //(k_time)*(S0_I_mjm)*lamvec[2]/sqrt(lamvec[0]^2 + lamvec[1]^2 + lamvec[2]^2);

F_plane_x_k = 0; //0 + (k_plane0*plane_dist + k_plane3*plane_dist^3 + 0)*itavec[0];
F_plane_y_k = 0; //0 + (k_plane0*plane_dist + k_plane3*plane_dist^3 + 0)*itavec[1];
F_plane_z_k = 0; //0 + (k_plane0*plane_dist + k_plane3*plane_dist^3 + 0)*itavec[2];

F_norm_x_k = 0;
F_norm_y_k = 0;
F_norm_z_k = k_norm*-zH_loc;

Fvec_loc[0] = F_time_x_d + F_time_x_k + F_plane_x_d + F_plane_x_k + F_norm_x_d + F_norm_x_k;
Fvec_loc[1] = F_time_y_d + F_time_y_k + F_plane_y_d + F_plane_y_k + F_norm_y_d + F_norm_y_k;
Fvec_loc[2] = F_time_z_d + F_time_z_k + F_plane_z_d + F_plane_z_k + F_norm_z_d + F_norm_z_k;

yH_loc_prev = yH_loc;

Fvec[0] = b1x*(Fvec_loc[0]) + b2x*(Fvec_loc[1]) + b3x*(Fvec_loc[2]);
Fvec[1] = b1y*(Fvec_loc[0]) + b2y*(Fvec_loc[1]) + b3y*(Fvec_loc[2]);
Fvec[2] = b1z*(Fvec_loc[0]) + b2z*(Fvec_loc[1]) + b3z*(Fvec_loc[2]);

F_help_x = Fvec[0];
F_help_y = Fvec[1];
F_help_z = Fvec[2];

F_help_mag = sqrt(F_help_x^2 + F_help_y^2 + F_help_z^2);

```

```

R_H = sqrt((position_x[h] - x_2)^2 + (position_z[h] -z_2)^2);

if (R_H <= R_safe && (position_y[h]-38) <= y_2)
    //23
{
touch0 = 1;
}
//23

if (touch0 == 1)
    //24
{
F_help_x = 0;
F_help_y = 0;
F_help_z = 0;
}
//24

test117 = test117 + 1;
}
//8
}
//1

tool1.updatePose();
tool1.computeForces();

// if((abs(t_position_x[h] - position_x[h]) =< 3.2) && (abs(t_position_y[h] - position_y[h]) =< 3) &&
(abs(t_position_z[h] - position_z[h]) =< 3.2) && (h_toolView.tool.buttonPressed) //&& peg.GetPosition[1] !=
cubel.GetPosition[1])

//if(h_t==0)
// {
// tool1.lastForce = [F_Help_x, F_Help_y, F_Help_z];
// }

if(a==1 && h_t == 1 && (h_0out != 1 || h_1out != 1 || h_2out != 1 || h_3out != 1 || h_4out != 1 || h_5out != 1 ||
h_6out != 1 || h_7out != 1 || h_8out != 1)) {
// tool1.lastForce = [0,-tool1.devicePosition[1]/10,0];
// F_y[h] = -((t_velocity_y[h]-velocity_y[h])*dam_peg + (t_position_y[h]-position_y[h])*stif_peg)*5;

//tool1.lastForce = [-F_m_x*((t_velocity_x[h]-velocity_x[h])*dam_peg +
(tool1.proxyPosition[0]-position_x[h])*stif_peg), -F_m_y*((t_velocity_y[h]-velocity_y[h])*dam_peg +
(tool1.proxyPosition[1]-position_y[h])*stif_peg), -F_m_z*((t_velocity_z[h]-velocity_z[h])*dam_peg +
(tool1.proxyPosition[2]-position_z[h])*stif_peg)];

```

```

    //tooll.lastForce = [-F_m_x*((t_velocity_x[h]-velocity_x[h])*dam_peg + (t_position_x[h]-position_x[h])*stif_peg),
-F_m_y*((t_velocity_y[h]-velocity_y[h])*dam_peg +
(t_position_y[h]-position_y[h])*stif_peg)-(-0.2*0.1*F_m_y*((peg0.GetPosition[1]-velocity_y[h])*dam_peg*2 +
(peg0.GetPosition[1]-position_y[h])*stif_peg*0.1)), -F_m_z*((t_velocity_z[h]-velocity_z[h])*dam_peg +
(t_position_z[h]-position_z[h])*stif_peg)];
    // F_help_x = 0;
    // F_help_y = 0;
    // F_help_z = 0;

    // apply forces 1
    tooll.lastForce = [-F_m_x*((t_velocity_x[h]-velocity_x[h])*dam_peg + (t_position_x[h]-position_x[h])*stif_peg) +
F_Help_x, -F_m_y*((t_velocity_y[h]-velocity_y[h])*dam_peg + (t_position_y[h]-position_y[h])*stif_peg) + F_Help_y,
-F_m_z*((t_velocity_z[h]-velocity_z[h])*dam_peg + (t_position_z[h]-position_z[h])*stif_peg)+F_Help_z];

    // tooll.lastForce = [F_Help_x, F_Help_y, F_Help_z];
    // tooll.lastForce = [5, 0, 0];
    if (h_all != 1)
    {

        if (in_hole == 0 && h_0 != 1)
        {

            if (ro_0 == 5)
            {
                tooll.lastForce = [-F_m_x*((t_velocity_x[h]-velocity_x[h])*dam_peg +
(t_position_x[h]-position_x[h])*stif_peg)+F_help_x, -0.5*F_m_y*((t_velocity_y[h]-velocity_y[h])*dam_peg +
(t_position_y[h]-position_y[h])*stif_peg)+F_help_y, -F_m_z*((t_velocity_z[h]-velocity_z[h])*dam_peg +
(t_position_z[h]-position_z[h])*stif_peg)+F_help_z];
            }

            if (ro_0 == 4)
            {
                tooll.lastForce = [-F_m_x*((t_velocity_x[h]-velocity_x[h])*dam_peg +
(t_position_x[h]-position_x[h])*stif_peg)+F_help_x, -0.55*F_m_y*((t_velocity_y[h]-velocity_y[h])*dam_peg +
(t_position_y[h]-position_y[h])*stif_peg)+F_help_y, -F_m_z*((t_velocity_z[h]-velocity_z[h])*dam_peg +
(t_position_z[h]-position_z[h])*stif_peg)+F_help_z];
            }

            if (ro_0 == 3)
            {
                tooll.lastForce = [-F_m_x*((t_velocity_x[h]-velocity_x[h])*dam_peg +
(t_position_x[h]-position_x[h])*stif_peg)+F_help_x, -0.6*F_m_y*((t_velocity_y[h]-velocity_y[h])*dam_peg +
(t_position_y[h]-position_y[h])*stif_peg)+F_help_y, -F_m_z*((t_velocity_z[h]-velocity_z[h])*dam_peg +
(t_position_z[h]-position_z[h])*stif_peg)+F_help_z];
            }

            if (ro_0 == 2)
            {
                tooll.lastForce = [-F_m_x*((t_velocity_x[h]-velocity_x[h])*dam_peg +
(t_position_x[h]-position_x[h])*stif_peg)+F_help_x, -0.9*F_m_y*((t_velocity_y[h]-velocity_y[h])*dam_peg +
(t_position_y[h]-position_y[h])*stif_peg)+F_help_y, -F_m_z*((t_velocity_z[h]-velocity_z[h])*dam_peg +
(t_position_z[h]-position_z[h])*stif_peg)+F_help_z];
            }
        }
    }

```

```

    if (ro_0 == 1)
    {
        tooll.lastForce = [-F_m_x*((t_velocity_x[h]-velocity_x[h])*dam_peg +
(t_position_x[h]-position_x[h])*stif_peg)+F_help_x, -1*F_m_y*((t_velocity_y[h]-velocity_y[h])*dam_peg +
(t_position_y[h]-position_y[h])*stif_peg)+F_help_y, -F_m_z*((t_velocity_z[h]-velocity_z[h])*dam_peg +
(t_position_z[h]-position_z[h])*stif_peg)+F_help_z];
    }

}

    if (in_hole == 1 && h_1 != 1)
    {
        if (ro_1 == 5)
        {
            tooll.lastForce = [-F_m_x*((t_velocity_x[h]-velocity_x[h])*dam_peg +
(t_position_x[h]-position_x[h])*stif_peg)+F_help_x, -0.5*F_m_y*((t_velocity_y[h]-velocity_y[h])*dam_peg +
(t_position_y[h]-position_y[h])*stif_peg)+F_help_y, -F_m_z*((t_velocity_z[h]-velocity_z[h])*dam_peg +
(t_position_z[h]-position_z[h])*stif_peg)+F_help_z];
        }
        if (ro_1 == 4)
        {
            tooll.lastForce = [-F_m_x*((t_velocity_x[h]-velocity_x[h])*dam_peg +
(t_position_x[h]-position_x[h])*stif_peg)+F_help_x, -0.55*F_m_y*((t_velocity_y[h]-velocity_y[h])*dam_peg +
(t_position_y[h]-position_y[h])*stif_peg)+F_help_y, -F_m_z*((t_velocity_z[h]-velocity_z[h])*dam_peg +
(t_position_z[h]-position_z[h])*stif_peg)+F_help_z];
        }
        if (ro_1 == 3)
        {
            tooll.lastForce = [-F_m_x*((t_velocity_x[h]-velocity_x[h])*dam_peg +
(t_position_x[h]-position_x[h])*stif_peg)+F_help_x, -0.6*F_m_y*((t_velocity_y[h]-velocity_y[h])*dam_peg +
(t_position_y[h]-position_y[h])*stif_peg)+F_help_y, -F_m_z*((t_velocity_z[h]-velocity_z[h])*dam_peg +
(t_position_z[h]-position_z[h])*stif_peg)+F_help_z];
        }
        if (ro_1 == 2)
        {
            tooll.lastForce = [-F_m_x*((t_velocity_x[h]-velocity_x[h])*dam_peg +
(t_position_x[h]-position_x[h])*stif_peg)+F_help_x, -0.9*F_m_y*((t_velocity_y[h]-velocity_y[h])*dam_peg +
(t_position_y[h]-position_y[h])*stif_peg)+F_help_y, -F_m_z*((t_velocity_z[h]-velocity_z[h])*dam_peg +
(t_position_z[h]-position_z[h])*stif_peg)+F_help_z];
        }
        if (ro_1 == 1)
        {
            tooll.lastForce = [-F_m_x*((t_velocity_x[h]-velocity_x[h])*dam_peg +
(t_position_x[h]-position_x[h])*stif_peg)+F_help_x, -1*F_m_y*((t_velocity_y[h]-velocity_y[h])*dam_peg +
(t_position_y[h]-position_y[h])*stif_peg)+F_help_y, -F_m_z*((t_velocity_z[h]-velocity_z[h])*dam_peg +
(t_position_z[h]-position_z[h])*stif_peg)+F_help_z];
        }
    }

    if (in_hole == 2 && h_2 != 1)
    {
        if (ro_2 == 5)

```



```

    {
        tool1.lastForce = [-F_m_x*((t_velocity_x[h]-velocity_x[h])*dam_peg +
(t_position_x[h]-position_x[h])*stif_peg)+F_help_x, -0.5*F_m_y*((t_velocity_y[h]-velocity_y[h])*dam_peg +
(t_position_y[h]-position_y[h])*stif_peg)+F_help_y, -F_m_z*((t_velocity_z[h]-velocity_z[h])*dam_peg +
(t_position_z[h]-position_z[h])*stif_peg)+F_help_z];
    }
    if (ro_2 == 4)
    {
        tool1.lastForce = [-F_m_x*((t_velocity_x[h]-velocity_x[h])*dam_peg +
(t_position_x[h]-position_x[h])*stif_peg)+F_help_x, -0.55*F_m_y*((t_velocity_y[h]-velocity_y[h])*dam_peg +
(t_position_y[h]-position_y[h])*stif_peg)+F_help_y, -F_m_z*((t_velocity_z[h]-velocity_z[h])*dam_peg +
(t_position_z[h]-position_z[h])*stif_peg)+F_help_z];
    }
    if (ro_2 == 3)
    {
        tool1.lastForce = [-F_m_x*((t_velocity_x[h]-velocity_x[h])*dam_peg +
(t_position_x[h]-position_x[h])*stif_peg)+F_help_x, -0.6*F_m_y*((t_velocity_y[h]-velocity_y[h])*dam_peg +
(t_position_y[h]-position_y[h])*stif_peg)+F_help_y, -F_m_z*((t_velocity_z[h]-velocity_z[h])*dam_peg +
(t_position_z[h]-position_z[h])*stif_peg)+F_help_z];
    }
    if (ro_2 == 2)
    {
        tool1.lastForce = [-F_m_x*((t_velocity_x[h]-velocity_x[h])*dam_peg +
(t_position_x[h]-position_x[h])*stif_peg)+F_help_x, -0.9*F_m_y*((t_velocity_y[h]-velocity_y[h])*dam_peg +
(t_position_y[h]-position_y[h])*stif_peg)+F_help_y, -F_m_z*((t_velocity_z[h]-velocity_z[h])*dam_peg +
(t_position_z[h]-position_z[h])*stif_peg)+F_help_z];
    }
    if (ro_2 == 1)
    {
        tool1.lastForce = [-F_m_x*((t_velocity_x[h]-velocity_x[h])*dam_peg +
(t_position_x[h]-position_x[h])*stif_peg)+F_help_x, -1*F_m_y*((t_velocity_y[h]-velocity_y[h])*dam_peg +
(t_position_y[h]-position_y[h])*stif_peg)+F_help_y, -F_m_z*((t_velocity_z[h]-velocity_z[h])*dam_peg +
(t_position_z[h]-position_z[h])*stif_peg)+F_help_z];
    }
}

if (in_hole == 3 && h_3 != 1)
{
    if (ro_3 == 5)
    {
        tool1.lastForce = [-F_m_x*((t_velocity_x[h]-velocity_x[h])*dam_peg +
(t_position_x[h]-position_x[h])*stif_peg)+F_help_x, -0.5*F_m_y*((t_velocity_y[h]-velocity_y[h])*dam_peg +
(t_position_y[h]-position_y[h])*stif_peg)+F_help_y, -F_m_z*((t_velocity_z[h]-velocity_z[h])*dam_peg +
(t_position_z[h]-position_z[h])*stif_peg)+F_help_z];
    }
    if (ro_3 == 4)
    {
        tool1.lastForce = [-F_m_x*((t_velocity_x[h]-velocity_x[h])*dam_peg +
(t_position_x[h]-position_x[h])*stif_peg)+F_help_x, -0.55*F_m_y*((t_velocity_y[h]-velocity_y[h])*dam_peg +
(t_position_y[h]-position_y[h])*stif_peg)+F_help_y, -F_m_z*((t_velocity_z[h]-velocity_z[h])*dam_peg +
(t_position_z[h]-position_z[h])*stif_peg)+F_help_z];
    }
    if (ro_3 == 3)

```

```

    {
        tool1.lastForce = [-F_m_x*((t_velocity_x[h]-velocity_x[h])*dam_peg +
(t_position_x[h]-position_x[h])*stif_peg)+F_help_x, -0.6*F_m_y*((t_velocity_y[h]-velocity_y[h])*dam_peg +
(t_position_y[h]-position_y[h])*stif_peg)+F_help_y, -F_m_z*((t_velocity_z[h]-velocity_z[h])*dam_peg +
(t_position_z[h]-position_z[h])*stif_peg)+F_help_z];
    }
    if (ro_3 == 2)
    {
        tool1.lastForce = [-F_m_x*((t_velocity_x[h]-velocity_x[h])*dam_peg +
(t_position_x[h]-position_x[h])*stif_peg)+F_help_x, -0.9*F_m_y*((t_velocity_y[h]-velocity_y[h])*dam_peg +
(t_position_y[h]-position_y[h])*stif_peg)+F_help_y, -F_m_z*((t_velocity_z[h]-velocity_z[h])*dam_peg +
(t_position_z[h]-position_z[h])*stif_peg)+F_help_z];
    }
    if (ro_3 == 1)
    {
        tool1.lastForce = [-F_m_x*((t_velocity_x[h]-velocity_x[h])*dam_peg +
(t_position_x[h]-position_x[h])*stif_peg)+F_help_x, -1*F_m_y*((t_velocity_y[h]-velocity_y[h])*dam_peg +
(t_position_y[h]-position_y[h])*stif_peg)+F_help_y, -F_m_z*((t_velocity_z[h]-velocity_z[h])*dam_peg +
(t_position_z[h]-position_z[h])*stif_peg)+F_help_z];
    }
}

    if (in_hole == 4 && h_4 != 1)
    {
        if (ro_4 == 5)
        {
            tool1.lastForce = [-F_m_x*((t_velocity_x[h]-velocity_x[h])*dam_peg +
(t_position_x[h]-position_x[h])*stif_peg)+F_help_x, -0.5*F_m_y*((t_velocity_y[h]-velocity_y[h])*dam_peg +
(t_position_y[h]-position_y[h])*stif_peg)+F_help_y, -F_m_z*((t_velocity_z[h]-velocity_z[h])*dam_peg +
(t_position_z[h]-position_z[h])*stif_peg)+F_help_z];
        }
        if (ro_4 == 4)
        {
            tool1.lastForce = [-F_m_x*((t_velocity_x[h]-velocity_x[h])*dam_peg +
(t_position_x[h]-position_x[h])*stif_peg)+F_help_x, -0.55*F_m_y*((t_velocity_y[h]-velocity_y[h])*dam_peg +
(t_position_y[h]-position_y[h])*stif_peg)+F_help_y, -F_m_z*((t_velocity_z[h]-velocity_z[h])*dam_peg +
(t_position_z[h]-position_z[h])*stif_peg)+F_help_z];
        }
        if (ro_4 == 3)
        {
            tool1.lastForce = [-F_m_x*((t_velocity_x[h]-velocity_x[h])*dam_peg +
(t_position_x[h]-position_x[h])*stif_peg)+F_help_x, -0.6*F_m_y*((t_velocity_y[h]-velocity_y[h])*dam_peg +
(t_position_y[h]-position_y[h])*stif_peg)+F_help_y, -F_m_z*((t_velocity_z[h]-velocity_z[h])*dam_peg +
(t_position_z[h]-position_z[h])*stif_peg)+F_help_z];
        }
        if (ro_4 == 2)
        {
            tool1.lastForce = [-F_m_x*((t_velocity_x[h]-velocity_x[h])*dam_peg +
(t_position_x[h]-position_x[h])*stif_peg)+F_help_x, -0.9*F_m_y*((t_velocity_y[h]-velocity_y[h])*dam_peg +
(t_position_y[h]-position_y[h])*stif_peg)+F_help_y, -F_m_z*((t_velocity_z[h]-velocity_z[h])*dam_peg +
(t_position_z[h]-position_z[h])*stif_peg)+F_help_z];
        }
        if (ro_4 == 1)

```

```

    {
        tooll.lastForce = [-F_m_x*((t_velocity_x[h]-velocity_x[h])*dam_peg +
(t_position_x[h]-position_x[h])*stif_peg)+F_help_x, -1*F_m_y*((t_velocity_y[h]-velocity_y[h])*dam_peg +
(t_position_y[h]-position_y[h])*stif_peg)+F_help_y, -F_m_z*((t_velocity_z[h]-velocity_z[h])*dam_peg +
(t_position_z[h]-position_z[h])*stif_peg)+F_help_z];
    }
}

    if (in_hole == 5 && h_5 != 1)
    {
        if (ro_5 == 5)
        {
            tooll.lastForce = [-F_m_x*((t_velocity_x[h]-velocity_x[h])*dam_peg +
(t_position_x[h]-position_x[h])*stif_peg)+F_help_x, -0.5*F_m_y*((t_velocity_y[h]-velocity_y[h])*dam_peg +
(t_position_y[h]-position_y[h])*stif_peg)+F_help_y, -F_m_z*((t_velocity_z[h]-velocity_z[h])*dam_peg +
(t_position_z[h]-position_z[h])*stif_peg)+F_help_z];
        }
        if (ro_5 == 4)
        {
            tooll.lastForce = [-F_m_x*((t_velocity_x[h]-velocity_x[h])*dam_peg +
(t_position_x[h]-position_x[h])*stif_peg)+F_help_x, -0.55*F_m_y*((t_velocity_y[h]-velocity_y[h])*dam_peg +
(t_position_y[h]-position_y[h])*stif_peg)+F_help_y, -F_m_z*((t_velocity_z[h]-velocity_z[h])*dam_peg +
(t_position_z[h]-position_z[h])*stif_peg)+F_help_z];
        }
        if (ro_5 == 3)
        {
            tooll.lastForce = [-F_m_x*((t_velocity_x[h]-velocity_x[h])*dam_peg +
(t_position_x[h]-position_x[h])*stif_peg)+F_help_x, -0.6*F_m_y*((t_velocity_y[h]-velocity_y[h])*dam_peg +
(t_position_y[h]-position_y[h])*stif_peg)+F_help_y, -F_m_z*((t_velocity_z[h]-velocity_z[h])*dam_peg +
(t_position_z[h]-position_z[h])*stif_peg)+F_help_z];
        }
        if (ro_5 == 2)
        {
            tooll.lastForce = [-F_m_x*((t_velocity_x[h]-velocity_x[h])*dam_peg +
(t_position_x[h]-position_x[h])*stif_peg)+F_help_x, -0.9*F_m_y*((t_velocity_y[h]-velocity_y[h])*dam_peg +
(t_position_y[h]-position_y[h])*stif_peg)+F_help_y, -F_m_z*((t_velocity_z[h]-velocity_z[h])*dam_peg +
(t_position_z[h]-position_z[h])*stif_peg)+F_help_z];
        }
        if (ro_5 == 1)
        {
            tooll.lastForce = [-F_m_x*((t_velocity_x[h]-velocity_x[h])*dam_peg +
(t_position_x[h]-position_x[h])*stif_peg)+F_help_x, -1*F_m_y*((t_velocity_y[h]-velocity_y[h])*dam_peg +
(t_position_y[h]-position_y[h])*stif_peg)+F_help_y, -F_m_z*((t_velocity_z[h]-velocity_z[h])*dam_peg +
(t_position_z[h]-position_z[h])*stif_peg)+F_help_z];
        }
    }
}

    if (in_hole == 6 && h_6 != 1)
    {
        if (ro_6 == 5)
        {

```

```

    tooll.lastForce = [-F_m_x*((t_velocity_x[h]-velocity_x[h])*dam_peg +
(t_position_x[h]-position_x[h])*stif_peg)+F_help_x, -0.5*F_m_y*((t_velocity_y[h]-velocity_y[h])*dam_peg +
(t_position_y[h]-position_y[h])*stif_peg)+F_help_y, -F_m_z*((t_velocity_z[h]-velocity_z[h])*dam_peg +
(t_position_z[h]-position_z[h])*stif_peg)+F_help_z];
}
if (ro_6 == 4)
{
    tooll.lastForce = [-F_m_x*((t_velocity_x[h]-velocity_x[h])*dam_peg +
(t_position_x[h]-position_x[h])*stif_peg)+F_help_x, -0.55*F_m_y*((t_velocity_y[h]-velocity_y[h])*dam_peg +
(t_position_y[h]-position_y[h])*stif_peg)+F_help_y, -F_m_z*((t_velocity_z[h]-velocity_z[h])*dam_peg +
(t_position_z[h]-position_z[h])*stif_peg)+F_help_z];
}
if (ro_6 == 3)
{
    tooll.lastForce = [-F_m_x*((t_velocity_x[h]-velocity_x[h])*dam_peg +
(t_position_x[h]-position_x[h])*stif_peg)+F_help_x, -0.6*F_m_y*((t_velocity_y[h]-velocity_y[h])*dam_peg +
(t_position_y[h]-position_y[h])*stif_peg)+F_help_y, -F_m_z*((t_velocity_z[h]-velocity_z[h])*dam_peg +
(t_position_z[h]-position_z[h])*stif_peg)+F_help_z];
}
if (ro_6 == 2)
{
    tooll.lastForce = [-F_m_x*((t_velocity_x[h]-velocity_x[h])*dam_peg +
(t_position_x[h]-position_x[h])*stif_peg)+F_help_x, -0.9*F_m_y*((t_velocity_y[h]-velocity_y[h])*dam_peg +
(t_position_y[h]-position_y[h])*stif_peg)+F_help_y, -F_m_z*((t_velocity_z[h]-velocity_z[h])*dam_peg +
(t_position_z[h]-position_z[h])*stif_peg)+F_help_z];
}
if (ro_6 == 1)
{
    tooll.lastForce = [-F_m_x*((t_velocity_x[h]-velocity_x[h])*dam_peg +
(t_position_x[h]-position_x[h])*stif_peg)+F_help_x, -1*F_m_y*((t_velocity_y[h]-velocity_y[h])*dam_peg +
(t_position_y[h]-position_y[h])*stif_peg)+F_help_y, -F_m_z*((t_velocity_z[h]-velocity_z[h])*dam_peg +
(t_position_z[h]-position_z[h])*stif_peg)+F_help_z];
}
}

if (in_hole == 7 && h_7 != 1)
{
    if (ro_7 == 5)
    {
        tooll.lastForce = [-F_m_x*((t_velocity_x[h]-velocity_x[h])*dam_peg +
(t_position_x[h]-position_x[h])*stif_peg)+F_help_x, -0.5*F_m_y*((t_velocity_y[h]-velocity_y[h])*dam_peg +
(t_position_y[h]-position_y[h])*stif_peg)+F_help_y, -F_m_z*((t_velocity_z[h]-velocity_z[h])*dam_peg +
(t_position_z[h]-position_z[h])*stif_peg)+F_help_z];
    }
    if (ro_7 == 4)
    {
        tooll.lastForce = [-F_m_x*((t_velocity_x[h]-velocity_x[h])*dam_peg +
(t_position_x[h]-position_x[h])*stif_peg)+F_help_x, -0.55*F_m_y*((t_velocity_y[h]-velocity_y[h])*dam_peg +
(t_position_y[h]-position_y[h])*stif_peg)+F_help_y, -F_m_z*((t_velocity_z[h]-velocity_z[h])*dam_peg +
(t_position_z[h]-position_z[h])*stif_peg)+F_help_z];
    }
    if (ro_7 == 3)
    {

```

```

    tooll.lastForce = [-F_m_x*((t_velocity_x[h]-velocity_x[h])*dam_peg +
(t_position_x[h]-position_x[h])*stif_peg)+F_help_x, -0.6*F_m_y*((t_velocity_y[h]-velocity_y[h])*dam_peg +
(t_position_y[h]-position_y[h])*stif_peg)+F_help_y, -F_m_z*((t_velocity_z[h]-velocity_z[h])*dam_peg +
(t_position_z[h]-position_z[h])*stif_peg)+F_help_z];
}
if (ro_7 == 2)
{
    tooll.lastForce = [-F_m_x*((t_velocity_x[h]-velocity_x[h])*dam_peg +
(t_position_x[h]-position_x[h])*stif_peg)+F_help_x, -0.9*F_m_y*((t_velocity_y[h]-velocity_y[h])*dam_peg +
(t_position_y[h]-position_y[h])*stif_peg)+F_help_y, -F_m_z*((t_velocity_z[h]-velocity_z[h])*dam_peg +
(t_position_z[h]-position_z[h])*stif_peg)+F_help_z];
}
if (ro_7 == 1)
{
    tooll.lastForce = [-F_m_x*((t_velocity_x[h]-velocity_x[h])*dam_peg +
(t_position_x[h]-position_x[h])*stif_peg)+F_help_x, -1*F_m_y*((t_velocity_y[h]-velocity_y[h])*dam_peg +
(t_position_y[h]-position_y[h])*stif_peg)+F_help_y, -F_m_z*((t_velocity_z[h]-velocity_z[h])*dam_peg +
(t_position_z[h]-position_z[h])*stif_peg)+F_help_z];
}
}

if (in_hole == 8 && h_8 != 1)
{
    if (ro_8 == 5)
    {
        tooll.lastForce = [-F_m_x*((t_velocity_x[h]-velocity_x[h])*dam_peg + (t_position_x[h]-position_x[h])*stif_peg)+
F_help_x, -0.5*F_m_y*((t_velocity_y[h]-velocity_y[h])*dam_peg + (t_position_y[h]-position_y[h])*stif_peg)+F_help_y,
-F_m_z*((t_velocity_z[h]-velocity_z[h])*dam_peg + (t_position_z[h]-position_z[h])*stif_peg)+F_help_z];
    }
    if (ro_8 == 4)
    {
        tooll.lastForce = [-F_m_x*((t_velocity_x[h]-velocity_x[h])*dam_peg +
(t_position_x[h]-position_x[h])*stif_peg)+F_help_x, -0.55*F_m_y*((t_velocity_y[h]-velocity_y[h])*dam_peg +
(t_position_y[h]-position_y[h])*stif_peg)+F_help_y, -F_m_z*((t_velocity_z[h]-velocity_z[h])*dam_peg +
(t_position_z[h]-position_z[h])*stif_peg)+F_help_z];
    }
    if (ro_8 == 3)
    {
        tooll.lastForce = [-F_m_x*((t_velocity_x[h]-velocity_x[h])*dam_peg +
(t_position_x[h]-position_x[h])*stif_peg)+F_help_x, -0.6*F_m_y*((t_velocity_y[h]-velocity_y[h])*dam_peg +
(t_position_y[h]-position_y[h])*stif_peg)+F_help_y, -F_m_z*((t_velocity_z[h]-velocity_z[h])*dam_peg +
(t_position_z[h]-position_z[h])*stif_peg)+F_help_z];
    }
    if (ro_8 == 2)
    {
        tooll.lastForce = [-F_m_x*((t_velocity_x[h]-velocity_x[h])*dam_peg +
(t_position_x[h]-position_x[h])*stif_peg)+F_help_x, -0.9*F_m_y*((t_velocity_y[h]-velocity_y[h])*dam_peg +
(t_position_y[h]-position_y[h])*stif_peg)+F_help_y, -F_m_z*((t_velocity_z[h]-velocity_z[h])*dam_peg +
(t_position_z[h]-position_z[h])*stif_peg)+F_help_z];
    }
    if (ro_8 == 1)
    {

```

```

        tool1.lastForce = [-F_m_x*((t_velocity_x[h]-velocity_x[h])*dam_peg +
(t_position_x[h]-position_x[h])*stif_peg)+F_help_x, -1*F_m_y*((t_velocity_y[h]-velocity_y[h])*dam_peg +
(t_position_y[h]-position_y[h])*stif_peg)+F_help_y, -F_m_z*((t_velocity_z[h]-velocity_z[h])*dam_peg +
(t_position_z[h]-position_z[h])*stif_peg)+F_help_z];
    }
}

}

if(h_all == 1 && ((h_0out == 1 && h_0outout != 1) || (h_1out == 1 && h_1outout != 1) || (h_2out == 1 && h_2outout
!= 1) || (h_3out == 1 && h_3outout != 1) || (h_4out == 1 && h_4outout != 1) || (h_5out == 1 && h_5outout != 1) || (h_6out
== 1 && h_6outout != 1) || (h_7out == 1 && h_7outout != 1) || (h_8out == 1 && h_8outout != 1)) && (h_0drop != 1 || h_1drop
!= 1 || h_2drop != 1 || h_3drop != 1 || h_4drop != 1 || h_5drop != 1 || h_6drop != 1 || h_7drop != 1 || h_8drop != 1))

{

    tool1.lastForce = [(-0.2*F_m_x*((t_velocity_x[h]-velocity_x[h])*dam_peg*2 +
(t_position_x[h]-position_x[h])*stif_peg)), -0.2*F_m_y*((t_velocity_y[h]-velocity_y[h])*dam_peg*2 +
(t_position_y[h]-position_y[h])*stif_peg), -0.2*F_m_z*((t_velocity_z[h]-velocity_z[h])*dam_peg*2 +
(t_position_z[h]-position_z[h])*stif_peg)];

}

if(h_all == 1 && (h_0out == 1 || h_1out == 1 || h_2out == 1 || h_3out == 1 || h_4out == 1 || h_5out == 1 ||
h_6out == 1 || h_7out == 1 || h_8out == 1) && (h_0drop != 1 || h_1drop != 1 || h_2drop != 1 || h_3drop != 1 || h_4drop !=
1 || h_5drop != 1 || h_6drop != 1 || h_7drop != 1 || h_8drop != 1) && (h_0in == 1 || h_1in == 1 || h_2in == 1 || h_3in ==
1 || h_4in == 1 || h_5in == 1 || h_6in == 1 || h_7in == 1 || h_8in == 1))

{

    tool1.lastForce = [-F_m_x*((t_velocity_x[h]-velocity_x[h])*dam_peg + (t_position_x[h]-position_x[h])*stif_peg),
-F_m_y*((t_velocity_y[h]-velocity_y[h])*dam_peg + (t_position_y[h]-position_y[h])*stif_peg),
-F_m_z*((t_velocity_z[h]-velocity_z[h])*dam_peg + (t_position_z[h]-position_z[h])*stif_peg)];

}

if(h_all == 1 && (h_0outout == 1 || h_1outout == 1 || h_2outout == 1 || h_3outout == 1 || h_4outout == 1 ||
h_5outout == 1 || h_6outout == 1 || h_7outout == 1 || h_8outout == 1))

{

    tool1.lastForce = [-F_m_x*((t_velocity_x[h]-velocity_x[h])*dam_peg + (t_position_x[h]-position_x[h])*stif_peg),
-F_m_y*((t_velocity_y[h]-velocity_y[h])*dam_peg + (t_position_y[h]-position_y[h])*stif_peg),
-F_m_z*((t_velocity_z[h]-velocity_z[h])*dam_peg + (t_position_z[h]-position_z[h])*stif_peg)];

}

```

```

//    if(hold==0)
//    {
//    tool1.lastForce = [F_Help_x, F_Help_y, F_Help_z];
//    }
//    // APPLY OR NOT APPLY FORCES!
tool1.applyForces();

    tool1.forceEnabled = true;
}

if (test == 1 || h_t == 1)
{
hold = 1;
}
else
{
hold = 0;
}

//OutputLN(position_x[h],",",position_y[h] -
38,",",position_z[h],",",X_Iloc[0],",",X_Iloc[1],",",X_Iloc[2],",",h,",",x_I,",",y_I,",",z_I,",",xP,",",yP,",",zP,",",xP_loc
//if (hold == 1)
//OutputLN(X_Hloc[0],",",X_Hloc[1],",",X_Hloc[2],",",X_Iloc[0],",",X_Iloc[1],",",X_Iloc[2],",",xP_loc,",",yP_loc,",",zP_loc
//OutputLN(next_hole,",",hold,",",peg_touch1,",",hold2,",",itavec[0],",",itavec[1],",",itavec[2],Fvec_loc[0],",",Fvec_loc[0]
//OutputLN(xP_loc,",",yP_loc,",",tau[hn]));

}

//fn = + (8*mjm_a4^2 + 16*mjm_a3*mjm_a5 + 8*mjm_b4^2 + 16*mjm_b3*mjm_b5)*tau[hn]^7 + (14*mjm_a3*mjm_a4 +
14*mjm_a2*mjm_a5 + 14*mjm_b3*mjm_b4 + 14*mjm_b2*mjm_b5)*tau[hn]^6 + (6*mjm_a3^2 + 12*mjm_a2*mjm_a4 + 12*mjm_a1*mjm_a5 +
6*mjm_b3^2 + 12*mjm_b2*mjm_b4 + 12*mjm_b1*mjm_b5)*tau[hn]^5 + (10*mjm_a2*mjm_a3 + 10*mjm_a1*mjm_a4 + 10*mjm_a0*mjm_a5 +
10*mjm_b2*mjm_b3 + 10*mjm_b1*mjm_b4 + 10*mjm_b0*mjm_b5 - 10*mjm_a5*xHI_loc - 10*mjm_b5*yHI_loc)*tau[hn]^4 + (4*mjm_a2^2 +
8*mjm_a1*mjm_a3 + 8*mjm_a0*mjm_a4 + 4*mjm_b2^2 + 8*mjm_b1*mjm_b3 + 8*mjm_b0*mjm_b4 - 8*mjm_a4*xHI_loc -
8*mjm_b4*yHI_loc)*tau[hn]^3 + (6*mjm_a1*mjm_a2 + 6*mjm_a0*mjm_a3 + 6*mjm_b1*mjm_b2 + 6*mjm_b0*mjm_b3 - 6*mjm_a3*xHI_loc -
6*mjm_b3*yHI_loc)*tau[hn]^2 + (2*mjm_a1^2 + 4*mjm_a0*mjm_a2 + 2*mjm_b1^2 + 4*mjm_b0*mjm_b2 - 4*mjm_a2*xHI_loc -
4*mjm_b2*yHI_loc)*tau[hn] + (2*mjm_a0*mjm_a1 + 2*mjm_b0*mjm_b1 - 2*mjm_a1*xHI_loc - 2*mjm_b1*yHI_loc);

function OnExit()
{
    // TO DO

// for (var i=0; i<h; i+=1)
// {

```

```

// OutputLN(position_x[i],",",position_y[i],",",position_z[i],",",i);
// }
}

// Camera manager (using mouse)
function CameraMoveMouse()
{
    static var InMouseR = false, InMouseL = false;
    static var PrecX = 0, PrecY = 0;
    // Change these values to modify the mouse sensitivity
    var TR_SENSITIVITY = 0.001;
    var ROT_SENSITIVITY = 0.01;
    // Mouse manager
    if(Mouse.ButtonL && !Mouse.ButtonR)
    {
        //==== Left Button: Camera rotation ====//
        if (InMouseL)
        {
            CameraRotate(( Mouse.X-PrecX)*ROT_SENSITIVITY,0,1,0);
            CameraRotateABS((Mouse.Y-PrecY)*ROT_SENSITIVITY,1,0,0);
        }
        else
        {
            PrecX = Mouse.X;
            PrecY = Mouse.Y;
        }
        InMouseL = true;
        InMouseR = false;
    }
    else
    if(Mouse.ButtonR)
    {
        //==== Right Button: Camera translation ====//
        if (InMouseR)
        {
            var CameraMatrix = GetCameraMatrix();
            var CameraPos = GetCameraPosition();
            if (!Mouse.ButtonL)
            //==== Translation on X and Z axis ====//
            CameraPos += GetCameraZAxis() * (Mouse.y-PrecY)*TR_SENSITIVITY + GetCameraXAxis() * (Mouse.X-
PrecX)*TR_SENSITIVITY;
            else
            //==== Right + Left Button: Translation on Y axis ====//
            CameraPos -= GetCameraYAxis() * (Mouse.y-PrecY)*TR_SENSITIVITY;
            SetCameraPosition(CameraPos);
        }
        else
        {
            PrecX = Mouse.X;
            PrecY = Mouse.Y;
        }
        InMouseR = true;
    }
}

```



```

        InMouseL = false;
    }
    else
    {
        InMouseR = false;
        InMouseL = false;
    }
}

// Draws a grid
function DrawGrid()
{
    var i;
    glLineWidth(1);

    glDisable(GL_LIGHTING);
    glColor(0.5,0.5,0.5);

    glBegin(GL_LINES);
    for( i= -100; i<=100; i+=10 )
    {
        glVertex(i, 0, 100 );
        glVertex(i, 0, -100 );

        glVertex( 100, 0, i );
        glVertex(-100, 0, i );
    }
    glEnd();
}

// Draws a grid
function DrawLines()
{
    if (hold!=0 && h_all!=1)
    {
// if ((position_x[h] - peg_0.GetPosition[0] >= 5 || position_y[h] - peg_0.GetPosition[1] >= 5 || position_z[h] -
peg_0.GetPosition[2] >= 5) && h_all != 1)
// {
// drawline = 1;
// }

// if (drawline == 1)
// {
// var i;
// glLineWidth(3);

// glDisable(GL_LIGHTING);
// glColor(0.1,0.1,0.8);

// glBegin(GL_LINES);

```

```

// i = h;
// for( i= i-h; i<=700; i+=10)
// {

//     glVertex(position_x[h-i], position_y[h-i], position_z[h-i]);

// }

// glEnd();
// }

    if(next_hole == 0)
    {

// for( i = 0; i <= 30; i+=1)
// {
// tau2 = i;

//}

        glLineWidth(5);

        glDisable(GL_LIGHTING);
        glColor(0.9,0.7,0.6);

        glBegin(GL_lines);
        {

            glVertex(0 + originX, 0 + originY, 0 + originZ);
            glVertex(0 + originX, 100 + originY, 0 + originZ);

        }
        glEnd();

        glLineWidth(5);

        glDisable(GL_LIGHTING);
        glColor(0.9,0.7,0.6);

        glBegin(GL_lines);
        {

            glVertex(0 + originX,0 + originY,0 + originZ);
            glVertex(0 + originX,0 + originY,100 + originZ);

        }
    }
}

```

```
glEnd();

glLineWidth(5);

glDisable(GL_LIGHTING);
glColor(0.9,0.7,0.6);

glBegin(GL_lines);
{
    glVertex(0 + originX,0 + originY,0 + originZ);
    glVertex(100 + originX,0 + originY,0 + originZ);
}
glEnd();
```

```
glLineWidth(5);

glDisable(GL_LIGHTING);
glColor(0.9,0.0,0.0);

glBegin(GL_TRIANGLES);
{
    glVertex(x_2,y_2,z_2);
    glVertex(x_1,y_1,z_1);
    glVertex(x_3,y_3,z_3);

    glVertex(x_3,y_3,z_3);
    glVertex(x_1,y_1,z_1);
    glVertex(x_2,y_2,z_2);
}
glEnd();
```

```
glLineWidth(5);

glDisable(GL_LIGHTING);
glColor(0.9,0.7,0.6);

glBegin(GL_POINTS);
{
    glVertex(x_2,y_2,z_2);
    glVertex(x_1,y_1,z_1);
    glVertex(x_3,y_3,z_3);
}
```

```

}
glEnd();

glLineWidth(5);

glDisable(GL_LIGHTING);
glColor(0.3,0.9,0.8);

glBegin(GL_TRIANGLES);
{
    glVertex(x_1loc[0] + originX,x_1loc[1] + originY,x_1loc[2] + originZ);
    glVertex(x_2loc[0] + originX,x_2loc[1] + originY,x_2loc[2] + originZ);
    glVertex(x_3loc[0] + originX,x_3loc[1] + originY,x_3loc[2] + originZ);

}
glEnd();

glPointSize(5);

glDisable(GL_LIGHTING);
glColor(0.0,0.0,0.7);

glBegin(GL_POINTS);
{
    glVertex(X_2loc[0] + originX,X_2loc[1] + originY,X_2loc[2] + originZ);
    glVertex(X_1loc[0] + originX,X_1loc[1] + originY,X_1loc[2] + originZ);
    glVertex(X_3loc[0] + originX,X_3loc[1] + originY,X_3loc[2] + originZ);

}
glEnd();

glLineWidth(5);

glDisable(GL_LIGHTING);
glColor(0.9,0.3,0.1);

glBegin(GL_LINES);
{
    glVertex(x_I, y_I, z_I);
    glVertex(xP, yP, zP);

}

```

```
glEnd();
```

```
glLineWidth(5);
```

```
glDisable(GL_LIGHTING);  
glColor(0.9,0.3,0.1);
```

```
glBegin(GL_LINES);
```

```
{
```

```
    glVertex(x_Iloc[0] + originX, x_Iloc[1] + originY, x_Iloc[2] + originZ);  
    glVertex(xP_loc + originX, yP_loc + originY, zP_loc + originZ);
```

```
}
```

```
glEnd();
```

```
glLineWidth(5);
```

```
glDisable(GL_LIGHTING);  
glColor(0.9,0.3,0.1);
```

```
glBegin(GL_LINES);
```

```
{
```

```
    glVertex(xP_loc + lamvec[0]*10 + originX, yP_loc + lamvec[1]*10 + originY, zP_loc + lamvec[2]*10 + originZ);  
    glVertex(xP_loc + originX, yP_loc + originY, zP_loc + originZ);
```

```
}
```

```
glEnd();
```

```
glLineWidth(5);
```

```
glDisable(GL_LIGHTING);  
glColor(0.7,0.3,0.7);
```

```
glBegin(GL_LINES);
```

```
{
```

```
    glVertex(position_x[h], position_y[h]-38, position_z[h]);  
    glVertex(x_I, y_I, z_I);
```

```

}
glEnd();

    if (hold == 1)
    {
glPointSize(5);

glDisable(GL_LIGHTING);
glColor(0.0,0.0,0.7);

glBegin(GL_POINTS);

{

    glVertex(x_mjm,y_mjm,z_mjm);

}
glEnd();

for (i = 1; i <= draw_points; i = i + 1)
{
glPointSize(5);

glDisable(GL_LIGHTING);
glColor(0.9,0.3,0.1);

glBegin(GL_POINTS);

{

    glVertex(x_mjm_draw[i] + originX, y_mjm_draw[i] + originY, 0 + originZ);

}
glEnd();
}

}

if (hold == 1)
{
glPointSize(5);

```

```
glDisable(GL_LIGHTING);
glColor(0.0,0.7,0.7);

glBegin(GL_POINTS);
{

    glVertex(x_mjm_MS,y_mjm_MS,z_mjm_MS);

}
glEnd();
}

}

if(next_hole == 1)
{

glLineWidth(5);

glDisable(GL_LIGHTING);
glColor(0.9,0.7,0.6);

glBegin(GL_lines);
{

    glVertex(0 + originX, 0 + originY, 0 + originZ);
    glVertex(0 + originX, 100 + originY, 0 + originZ);

}
glEnd();

glLineWidth(5);

glDisable(GL_LIGHTING);
glColor(0.9,0.7,0.6);

glBegin(GL_lines);
{
```

```
    glVertex(0 + originX,0 + originY,0 + originZ);
    glVertex(0 + originX,0 + originY,100 + originZ);

}
glEnd();

glLineWidth(5);

glDisable(GL_LIGHTING);
glColor(0.9,0.7,0.6);

glBegin(GL_lines);

{
    glVertex(0 + originX,0 + originY,0 + originZ);
    glVertex(100 + originX,0 + originY,0 + originZ);

}
glEnd();

glLineWidth(5);

glDisable(GL_LIGHTING);
glColor(0.9,0.0,0.0);

glBegin(GL_TRIANGLES);

{
    glVertex(x_2,y_2,z_2);
    glVertex(x_1,y_1,z_1);
    glVertex(x_3,y_3,z_3);

    glVertex(x_3,y_3,z_3);
    glVertex(x_1,y_1,z_1);
    glVertex(x_2,y_2,z_2);

}
glEnd();

glLineWidth(5);

glDisable(GL_LIGHTING);
glColor(0.9,0.7,0.6);
```



```
glBegin(GL_POINTS);
{
    glVertex(x_2,y_2,z_2);
    glVertex(x_1,y_1,z_1);
    glVertex(x_3,y_3,z_3);
}
glEnd();

glLineWidth(5);

glDisable(GL_LIGHTING);
glColor(0.3,0.9,0.8);

glBegin(GL_TRIANGLES);
{
    glVertex(x_1loc[0] + originX,x_1loc[1] + originY,x_1loc[2] + originZ);
    glVertex(x_2loc[0] + originX,x_2loc[1] + originY,x_2loc[2] + originZ);
    glVertex(x_3loc[0] + originX,x_3loc[1] + originY,x_3loc[2] + originZ);
}
glEnd();

glPointSize(5);

glDisable(GL_LIGHTING);
glColor(0.0,0.0,0.7);

glBegin(GL_POINTS);
{
    glVertex(X_2loc[0] + originX,X_2loc[1] + originY,X_2loc[2] + originZ);
    glVertex(X_1loc[0] + originX,X_1loc[1] + originY,X_1loc[2] + originZ);
    glVertex(X_3loc[0] + originX,X_3loc[1] + originY,X_3loc[2] + originZ);
}
glEnd();

glLineWidth(5);
```

```

glDisable(GL_LIGHTING);
glColor(0.9,0.3,0.1);

glBegin(GL_LINES);
{
    glVertex(x_I,y_I,z_I);
    glVertex(xP, yP, zP);

}
glEnd();

glLineWidth(5);

glDisable(GL_LIGHTING);
glColor(0.9,0.3,0.1);

glBegin(GL_LINES);
{
    glVertex(x_Iloc[0] + originX,x_Iloc[1] + originY,x_Iloc[2] + originZ);
    glVertex(xP_loc + originX, yP_loc + originY, zP_loc + originZ);

}
glEnd();

glLineWidth(5);

glDisable(GL_LIGHTING);
glColor(0.7,0.3,0.7);

glBegin(GL_LINES);
{

    glVertex(position_x[h],position_y[h]-38,position_z[h]);
    glVertex(x_I, y_I, z_I);

}
glEnd();
}

```

```
if(next_hole == 2)
{

glLineWidth(5);

glDisable(GL_LIGHTING);
glColor(0.9,0.7,0.6);

glBegin(GL_lines);
{
    glVertex(0,0,0);
    glVertex(0,100,0);

}
glEnd();

glLineWidth(5);

glDisable(GL_LIGHTING);
glColor(0.9,0.7,0.6);

glBegin(GL_lines);
{
    glVertex(0,0,0);
    glVertex(0,0,100);

}
glEnd();

glLineWidth(5);

glDisable(GL_LIGHTING);
glColor(0.9,0.7,0.6);

glBegin(GL_lines);
{
    glVertex(0,0,0);
    glVertex(100,0,0);
```

```
}
glEnd();

glLineWidth(5);

glDisable(GL_LIGHTING);
glColor(0.9,0.0,0.0);

glBegin(GL_TRIANGLES);
{
    glVertex(x_2,y_2,z_2);
    glVertex(x_1,y_1,z_1);
    glVertex(x_3,y_3,z_3);

    glVertex(x_3,y_3,z_3);
    glVertex(x_1,y_1,z_1);
    glVertex(x_2,y_2,z_2);
}
glEnd();

glLineWidth(5);

glDisable(GL_LIGHTING);
glColor(0.9,0.7,0.6);

glBegin(GL_POINTS);
{
    glVertex(x_2,y_2,z_2);
    glVertex(x_1,y_1,z_1);
    glVertex(x_3,y_3,z_3);
}
glEnd();

glLineWidth(5);

glDisable(GL_LIGHTING);
glColor(0.3,0.9,0.8);

glBegin(GL_TRIANGLES);
{
```

```
    glVertex(x_1loc[0],x_1loc[1],x_1loc[2]);
    glVertex(x_2loc[0],x_2loc[1],x_2loc[2]);
    glVertex(x_3loc[0],x_3loc[1],x_3loc[2]);

}
glEnd();

glPointSize(5);

glDisable(GL_LIGHTING);
glColor(0.0,0.0,0.7);

glBegin(GL_POINTS);
{
    glVertex(X_2loc[0],X_2loc[1],X_2loc[2]);
    glVertex(X_1loc[0],X_1loc[1],X_1loc[2]);
    glVertex(X_3loc[0],X_3loc[1],X_3loc[2]);

}
glEnd();

glLineWidth(5);

glDisable(GL_LIGHTING);
glColor(0.9,0.3,0.1);

glBegin(GL_LINES);
{
    glVertex(x_I,y_I,z_I);
    glVertex(xP, yP, zP);

}
glEnd();

glLineWidth(5);

glDisable(GL_LIGHTING);
glColor(0.9,0.3,0.1);

glBegin(GL_LINES);
{
```

```
    glVertex(x_Iloc[0],x_Iloc[1],x_Iloc[2]);
    glVertex(xP_loc, yP_loc, zP_loc);

}
glEnd();
```

```
glLineWidth(5);

glDisable(GL_LIGHTING);
glColor(0.7,0.3,0.7);

glBegin(GL_LINES);

{

    glVertex(position_x[h],position_y[h]-38,position_z[h]);
    glVertex(x_I, y_I, z_I);

}
glEnd();
}
```

```
if(next_hole == 3)
{
```

```
    glLineWidth(5);

    glDisable(GL_LIGHTING);
    glColor(0.9,0.7,0.6);

    glBegin(GL_lines);

    {

        glVertex(0,0,0);
        glVertex(0,100,0);

    }

}
```

```
glEnd();
```

```
glLineWidth(5);

glDisable(GL_LIGHTING);
glColor(0.9,0.7,0.6);

glBegin(GL_lines);

{
    glVertex(0,0,0);
    glVertex(0,0,100);

}
glEnd();

glLineWidth(5);

glDisable(GL_LIGHTING);
glColor(0.9,0.7,0.6);

glBegin(GL_lines);

{
    glVertex(0,0,0);
    glVertex(100,0,0);

}
glEnd();

glLineWidth(5);

glDisable(GL_LIGHTING);
glColor(0.9,0.0,0.0);

glBegin(GL_TRIANGLES);

{
    glVertex(x_2,y_2,z_2);
    glVertex(x_1,y_1,z_1);
    glVertex(x_3,y_3,z_3);

    glVertex(x_3,y_3,z_3);
    glVertex(x_1,y_1,z_1);
    glVertex(x_2,y_2,z_2);
```

```
}
glEnd();

glLineWidth(5);

glDisable(GL_LIGHTING);
glColor(0.9,0.7,0.6);

glBegin(GL_POINTS);
{
    glVertex(x_2,y_2,z_2);
    glVertex(x_1,y_1,z_1);
    glVertex(x_3,y_3,z_3);

}
glEnd();

glLineWidth(5);

glDisable(GL_LIGHTING);
glColor(0.3,0.9,0.8);

glBegin(GL_TRIANGLES);
{
    glVertex(x_1loc[0],x_1loc[1],x_1loc[2]);
    glVertex(x_2loc[0],x_2loc[1],x_2loc[2]);
    glVertex(x_3loc[0],x_3loc[1],x_3loc[2]);

}
glEnd();

glPointSize(5);

glDisable(GL_LIGHTING);
glColor(0.0,0.0,0.7);

glBegin(GL_POINTS);
{
    glVertex(X_2loc[0],X_2loc[1],X_2loc[2]);
```



```
    glVertex(X_1loc[0],X_1loc[1],X_1loc[2]);
    glVertex(X_3loc[0],X_3loc[1],X_3loc[2]);

}
glEnd();

glLineWidth(5);

glDisable(GL_LIGHTING);
glColor(0.9,0.3,0.1);

glBegin(GL_LINES);
{
    glVertex(x_I,y_I,z_I);
    glVertex(xP, yP, zP);

}
glEnd();

glLineWidth(5);

glDisable(GL_LIGHTING);
glColor(0.9,0.3,0.1);

glBegin(GL_LINES);
{
    glVertex(x_Iloc[0],x_Iloc[1],x_Iloc[2]);
    glVertex(xP_loc, yP_loc, zP_loc);

}
glEnd();

glLineWidth(5);

glDisable(GL_LIGHTING);
glColor(0.7,0.3,0.7);

glBegin(GL_LINES);
```

```
{

    glVertex(position_x[h],position_y[h]-38,position_z[h]);
    glVertex(x_I, y_I, z_I);

}
glEnd();
}

if(next_hole == 4)
{

glLineWidth(5);

glDisable(GL_LIGHTING);
glColor(0.9,0.7,0.6);

glBegin(GL_lines);
{

    glVertex(0,0,0);
    glVertex(0,100,0);

}
glEnd();

glLineWidth(5);

glDisable(GL_LIGHTING);
glColor(0.9,0.7,0.6);

glBegin(GL_lines);
{

    glVertex(0,0,0);
    glVertex(0,0,100);

}
glEnd();

glLineWidth(5);

glDisable(GL_LIGHTING);
```

```
glColor(0.9,0.7,0.6);  
glBegin(GL_lines);  
{  
    glVertex(0,0,0);  
    glVertex(100,0,0);  
  
}  
glEnd();
```

```
glLineWidth(5);  
glDisable(GL_LIGHTING);  
glColor(0.9,0.0,0.0);  
glBegin(GL_TRIANGLES);  
{  
    glVertex(x_2,y_2,z_2);  
    glVertex(x_1,y_1,z_1);  
    glVertex(x_3,y_3,z_3);  
  
    glVertex(x_3,y_3,z_3);  
    glVertex(x_1,y_1,z_1);  
    glVertex(x_2,y_2,z_2);  
  
}  
glEnd();
```

```
glLineWidth(5);  
glDisable(GL_LIGHTING);  
glColor(0.9,0.7,0.6);  
glBegin(GL_POINTS);  
{  
    glVertex(x_2,y_2,z_2);  
    glVertex(x_1,y_1,z_1);  
    glVertex(x_3,y_3,z_3);  
  
}  
glEnd();
```

```
glLineWidth(5);

glDisable(GL_LIGHTING);
glColor(0.3,0.9,0.8);

glBegin(GL_TRIANGLES);
{
    glVertex(x_1loc[0],x_1loc[1],x_1loc[2]);
    glVertex(x_2loc[0],x_2loc[1],x_2loc[2]);
    glVertex(x_3loc[0],x_3loc[1],x_3loc[2]);

}
glEnd();

glPointSize(5);

glDisable(GL_LIGHTING);
glColor(0.0,0.0,0.7);

glBegin(GL_POINTS);
{
    glVertex(X_2loc[0],X_2loc[1],X_2loc[2]);
    glVertex(X_1loc[0],X_1loc[1],X_1loc[2]);
    glVertex(X_3loc[0],X_3loc[1],X_3loc[2]);

}
glEnd();

glLineWidth(5);

glDisable(GL_LIGHTING);
glColor(0.9,0.3,0.1);

glBegin(GL_LINES);
{
    glVertex(x_I,y_I,z_I);
    glVertex(xP,yP,zP);

}
}
```

```
glEnd();

glLineWidth(5);

glDisable(GL_LIGHTING);
glColor(0.9,0.3,0.1);

glBegin(GL_LINES);
{
    glVertex(x_Iloc[0],x_Iloc[1],x_Iloc[2]);
    glVertex(xP_loc, yP_loc, zP_loc);
}
glEnd();

glLineWidth(5);

glDisable(GL_LIGHTING);
glColor(0.7,0.3,0.7);

glBegin(GL_LINES);
{
    glVertex(position_x[h],position_y[h]-38,position_z[h]);
    glVertex(x_I, y_I, z_I);
}
glEnd();
}

if(next_hole == 5)
{

glLineWidth(5);

glDisable(GL_LIGHTING);
glColor(0.9,0.7,0.6);

glBegin(GL_lines);
```

```
{
    glVertex(0,0,0);
    glVertex(0,100,0);

}
glEnd();

glLineWidth(5);

glDisable(GL_LIGHTING);
glColor(0.9,0.7,0.6);

glBegin(GL_lines);
{
    glVertex(0,0,0);
    glVertex(0,0,100);

}
glEnd();

glLineWidth(5);

glDisable(GL_LIGHTING);
glColor(0.9,0.7,0.6);

glBegin(GL_lines);
{
    glVertex(0,0,0);
    glVertex(100,0,0);

}
glEnd();

glLineWidth(5);

glDisable(GL_LIGHTING);
glColor(0.9,0.0,0.0);

glBegin(GL_TRIANGLES);
```

```
{
    glVertex(x_2, y_2, z_2);
    glVertex(x_1, y_1, z_1);
    glVertex(x_3, y_3, z_3);

    glVertex(x_3, y_3, z_3);
    glVertex(x_1, y_1, z_1);
    glVertex(x_2, y_2, z_2);

}
glEnd();

glLineWidth(5);

glDisable(GL_LIGHTING);
glColor(0.9, 0.7, 0.6);

glBegin(GL_POINTS);
{
    glVertex(x_2, y_2, z_2);
    glVertex(x_1, y_1, z_1);
    glVertex(x_3, y_3, z_3);

}
glEnd();

glLineWidth(5);

glDisable(GL_LIGHTING);
glColor(0.3, 0.9, 0.8);

glBegin(GL_TRIANGLES);
{
    glVertex(x_1loc[0], x_1loc[1], x_1loc[2]);
    glVertex(x_2loc[0], x_2loc[1], x_2loc[2]);
    glVertex(x_3loc[0], x_3loc[1], x_3loc[2]);

}
glEnd();
```

```
glPointSize(5);

glDisable(GL_LIGHTING);
glColor(0.0,0.0,0.7);

glBegin(GL_POINTS);
{
    glVertex(X_2loc[0],X_2loc[1],X_2loc[2]);
    glVertex(X_1loc[0],X_1loc[1],X_1loc[2]);
    glVertex(X_3loc[0],X_3loc[1],X_3loc[2]);

}
glEnd();

glLineWidth(5);

glDisable(GL_LIGHTING);
glColor(0.9,0.3,0.1);

glBegin(GL_LINES);
{
    glVertex(x_I,y_I,z_I);
    glVertex(xP, yP, zP);

}
glEnd();

glLineWidth(5);

glDisable(GL_LIGHTING);
glColor(0.9,0.3,0.1);

glBegin(GL_LINES);
{
    glVertex(x_Iloc[0],x_Iloc[1],x_Iloc[2]);
    glVertex(xP_loc, yP_loc, zP_loc);

}
glEnd();
```



```
glLineWidth(5);

glDisable(GL_LIGHTING);
glColor(0.7,0.3,0.7);

glBegin(GL_LINES);
{

    glVertex(position_x[h],position_y[h]-38,position_z[h]);
    glVertex(x_I, y_I, z_I);

}
glEnd();
}

if(next_hole == 6)
{

glLineWidth(5);

glDisable(GL_LIGHTING);
glColor(0.9,0.7,0.6);

glBegin(GL_lines);
{

    glVertex(0,0,0);
    glVertex(0,100,0);

}
glEnd();

glLineWidth(5);

glDisable(GL_LIGHTING);
glColor(0.9,0.7,0.6);

glBegin(GL_lines);
{

    glVertex(0,0,0);
    glVertex(0,0,100);
```

```
}
glEnd();

glLineWidth(5);

glDisable(GL_LIGHTING);
glColor(0.9,0.7,0.6);

glBegin(GL_lines);
{
    glVertex(0,0,0);
    glVertex(100,0,0);

}
glEnd();

glLineWidth(5);

glDisable(GL_LIGHTING);
glColor(0.9,0.0,0.0);

glBegin(GL_TRIANGLES);
{
    glVertex(x_2,y_2,z_2);
    glVertex(x_1,y_1,z_1);
    glVertex(x_3,y_3,z_3);

    glVertex(x_3,y_3,z_3);
    glVertex(x_1,y_1,z_1);
    glVertex(x_2,y_2,z_2);

}
glEnd();

glLineWidth(5);

glDisable(GL_LIGHTING);
glColor(0.9,0.7,0.6);

glBegin(GL_POINTS);
```

```
{
    glVertex(x_2, y_2, z_2);
    glVertex(x_1, y_1, z_1);
    glVertex(x_3, y_3, z_3);

}
glEnd();

glLineWidth(5);

glDisable(GL_LIGHTING);
glColor(0.3, 0.9, 0.8);

glBegin(GL_TRIANGLES);
{
    glVertex(x_1loc[0], x_1loc[1], x_1loc[2]);
    glVertex(x_2loc[0], x_2loc[1], x_2loc[2]);
    glVertex(x_3loc[0], x_3loc[1], x_3loc[2]);

}
glEnd();

glPointSize(5);

glDisable(GL_LIGHTING);
glColor(0.0, 0.0, 0.7);

glBegin(GL_POINTS);
{
    glVertex(X_2loc[0], X_2loc[1], X_2loc[2]);
    glVertex(X_1loc[0], X_1loc[1], X_1loc[2]);
    glVertex(X_3loc[0], X_3loc[1], X_3loc[2]);

}
glEnd();

glLineWidth(5);

glDisable(GL_LIGHTING);
glColor(0.9, 0.3, 0.1);
```

```
glBegin(GL_LINES);
{
    glVertex(x_I, y_I, z_I);
    glVertex(xP, yP, zP);

}
glEnd();

glLineWidth(5);

glDisable(GL_LIGHTING);
glColor(0.9, 0.3, 0.1);

glBegin(GL_LINES);
{
    glVertex(x_Iloc[0], x_Iloc[1], x_Iloc[2]);
    glVertex(xP_loc, yP_loc, zP_loc);

}
glEnd();

glLineWidth(5);

glDisable(GL_LIGHTING);
glColor(0.7, 0.3, 0.7);

glBegin(GL_LINES);
{

    glVertex(position_x[h], position_y[h]-38, position_z[h]);
    glVertex(x_I, y_I, z_I);

}
glEnd();
}

if(next_hole == 7)
{
```

```
glLineWidth(5);

glDisable(GL_LIGHTING);
glColor(0.9,0.7,0.6);

glBegin(GL_lines);

{
    glVertex(0,0,0);
    glVertex(0,100,0);

}
glEnd();

glLineWidth(5);

glDisable(GL_LIGHTING);
glColor(0.9,0.7,0.6);

glBegin(GL_lines);

{
    glVertex(0,0,0);
    glVertex(0,0,100);

}
glEnd();

glLineWidth(5);

glDisable(GL_LIGHTING);
glColor(0.9,0.7,0.6);

glBegin(GL_lines);

{
    glVertex(0,0,0);
    glVertex(100,0,0);

}
glEnd();
```

```
glLineWidth(5);

glDisable(GL_LIGHTING);
glColor(0.9,0.0,0.0);

glBegin(GL_TRIANGLES);
{
    glVertex(x_2,y_2,z_2);
    glVertex(x_1,y_1,z_1);
    glVertex(x_3,y_3,z_3);

    glVertex(x_3,y_3,z_3);
    glVertex(x_1,y_1,z_1);
    glVertex(x_2,y_2,z_2);

}
glEnd();

glLineWidth(5);

glDisable(GL_LIGHTING);
glColor(0.9,0.7,0.6);

glBegin(GL_POINTS);
{
    glVertex(x_2,y_2,z_2);
    glVertex(x_1,y_1,z_1);
    glVertex(x_3,y_3,z_3);

}
glEnd();

glLineWidth(5);

glDisable(GL_LIGHTING);
glColor(0.3,0.9,0.8);

glBegin(GL_TRIANGLES);
{
    glVertex(x_1loc[0],x_1loc[1],x_1loc[2]);
    glVertex(x_2loc[0],x_2loc[1],x_2loc[2]);
    glVertex(x_3loc[0],x_3loc[1],x_3loc[2]);
```

```
}  
glEnd();
```

```
glPointSize(5);
```

```
glDisable(GL_LIGHTING);  
glColor(0.0,0.0,0.7);
```

```
glBegin(GL_POINTS);
```

```
{  
  
    glVertex(X_2loc[0],X_2loc[1],X_2loc[2]);  
    glVertex(X_1loc[0],X_1loc[1],X_1loc[2]);  
    glVertex(X_3loc[0],X_3loc[1],X_3loc[2]);
```

```
}  
glEnd();
```

```
glLineWidth(5);
```

```
glDisable(GL_LIGHTING);  
glColor(0.9,0.3,0.1);
```

```
glBegin(GL_LINES);
```

```
{  
  
    glVertex(x_I,y_I,z_I);  
    glVertex(xP, yP, zP);
```

```
}  
glEnd();
```

```
glLineWidth(5);
```

```
glDisable(GL_LIGHTING);  
glColor(0.9,0.3,0.1);
```

```
glBegin(GL_LINES);
```

```
{  
  
    glVertex(x_Iloc[0],x_Iloc[1],x_Iloc[2]);  
    glVertex(xP_loc, yP_loc, zP_loc);
```

```
}  
glEnd();
```

```
glLineWidth(5);
```

```
glDisable(GL_LIGHTING);  
glColor(0.7,0.3,0.7);
```

```
glBegin(GL_LINES);
```

```
{
```

```
    glVertex(position_x[h],position_y[h]-38,position_z[h]);  
    glVertex(x_I, y_I, z_I);
```

```
}  
glEnd();  
}
```

```
if(next_hole == 8)  
{
```

```
glLineWidth(5);
```

```
glDisable(GL_LIGHTING);  
glColor(0.9,0.7,0.6);
```

```
glBegin(GL_lines);
```

```
{
```

```
    glVertex(0,0,0);  
    glVertex(0,100,0);
```

```
}  
glEnd();
```

```
glLineWidth(5);
```

```
glDisable(GL_LIGHTING);  
glColor(0.9,0.7,0.6);
```



```
glBegin(GL_lines);
{
    glVertex(0,0,0);
    glVertex(0,0,100);

}
glEnd();

glLineWidth(5);

glDisable(GL_LIGHTING);
glColor(0.9,0.7,0.6);

glBegin(GL_lines);
{
    glVertex(0,0,0);
    glVertex(100,0,0);

}
glEnd();

glLineWidth(5);

glDisable(GL_LIGHTING);
glColor(0.9,0.0,0.0);

glBegin(GL_TRIANGLES);
{
    glVertex(x_2,y_2,z_2);
    glVertex(x_1,y_1,z_1);
    glVertex(x_3,y_3,z_3);

    glVertex(x_3,y_3,z_3);
    glVertex(x_1,y_1,z_1);
    glVertex(x_2,y_2,z_2);

}
glEnd();
```

```
glLineWidth(5);

glDisable(GL_LIGHTING);
glColor(0.9,0.7,0.6);

glBegin(GL_POINTS);

{
    glVertex(x_2,y_2,z_2);
    glVertex(x_1,y_1,z_1);
    glVertex(x_3,y_3,z_3);

}
glEnd();

glLineWidth(5);

glDisable(GL_LIGHTING);
glColor(0.3,0.9,0.8);

glBegin(GL_TRIANGLES);

{
    glVertex(x_1loc[0],x_1loc[1],x_1loc[2]);
    glVertex(x_2loc[0],x_2loc[1],x_2loc[2]);
    glVertex(x_3loc[0],x_3loc[1],x_3loc[2]);

}
glEnd();

glPointSize(5);

glDisable(GL_LIGHTING);
glColor(0.0,0.0,0.7);

glBegin(GL_POINTS);

{
    glVertex(X_2loc[0],X_2loc[1],X_2loc[2]);
    glVertex(X_1loc[0],X_1loc[1],X_1loc[2]);
    glVertex(X_3loc[0],X_3loc[1],X_3loc[2]);

}

}
```

```
glEnd();

glLineWidth(5);

glDisable(GL_LIGHTING);
glColor(0.9,0.3,0.1);

glBegin(GL_LINES);
{
    glVertex(x_I, y_I, z_I);
    glVertex(xP, yP, zP);

}
glEnd();

glLineWidth(5);

glDisable(GL_LIGHTING);
glColor(0.9,0.3,0.1);

glBegin(GL_LINES);
{
    glVertex(x_Iloc[0],x_Iloc[1],x_Iloc[2]);
    glVertex(xP_loc, yP_loc, zP_loc);

}
glEnd();

glLineWidth(5);

glDisable(GL_LIGHTING);
glColor(0.7,0.3,0.7);

glBegin(GL_LINES);
{

    glVertex(position_x[h],position_y[h]-38,position_z[h]);
    glVertex(x_I, y_I, z_I);
```

```

}
glEnd();

}

glLineWidth(5);

glDisable(GL_LIGHTING);
glColor(0.7,0.3,0.1);

glBegin(GL_LINES);

{

// for(var i=1; i<=h; i+=1)
// {
// glVertex(position_x[h-1],position_y[h-1],position_z[h-1]);
// glVertex(position_x[h],position_y[h],position_z[h]);
// }

}
glEnd();

}

if (hold == 0)
{
drawline = 0;
}
}

function ShowNextHole()
{

if (hold!=0 && h_all!=1)
{

for(var i=0; i<=8; i+=1)
{
if(next_hole == i)
{

//glLineWidth(8);
glPointSize(2);

```

```

glDisable(GL_LIGHTING);
glColor(rand(5)*0.1,0.9*rand(5),0.1*rand(5));

xf_c = x_2 + R_c;
xs_c = x_2 - R_c;
dx_c = (xf_c - xs_c)/ni_draw;

glBegin(GL_POINTS);
{
    glVertex(xs_c,y_2,z_2);

    for (var i_c = 0; i_c<=ni_draw + 1; i_c+=1)
    {
        x_c = xs_c + (i_c+1)*dx_c;

        z_c1 = -sqrt((R_c)^2 - (x_c - x_2)^2) + z_2;

        glVertex(x_c,y_2,z_c1);
//        outputLN(x_c," ",z_c1," ",x_2," ",z_2," ",i_c);

    }

}

glEnd();

xf_c = x_2 + R_c;
xs_c = x_2 - R_c;
dx_c = (xf_c - xs_c)/ni_draw;

glBegin(GL_POINTS);
{
    for (var i_c = 0; i_c<=ni_draw + 1; i_c+=1)
    {
        x_c = xs_c + (i_c+1)*dx_c;

        z_c2 = sqrt((R_c)^2 - (x_c - x_2)^2) + z_2;

        glVertex(x_c,y_2,z_c2);

    }

}

```

```
glEnd();
```

```
}
```

```
}
```

```
}
```

```
}
```

Evagoras Xydias