no\_logfile = true;

default\_font\_size = 24;

# example of making presentation use visual angle in degrees

screen\_width = 1280;

screen\_height = 1024;

screen\_bit\_depth = 32;

# set these to screen measurements in inches

#screen\_width\_distance = 35.5;

screen\_width\_distance = 47.5;

screen\_height\_distance = 26.75;

# set screen\_distance to distance between screen and eyes of subject

screen\_distance = 48;

begin;

#picture {} default;

trial {

trial\_duration = 1000;

picture {

text { caption = " "; } text1;

x = 0; y = 0;

} pic1;

} trial1;

trial {

trial\_duration = 2000;

picture {

text { caption = " "; } text2;

x = 0; y = 0;

} pic2;

} blackScreen;

sound { wavefile { filename = "C:\\Documents and Settings\\USER\\Desktop\\Behavioral Control\\pics\_sounds\\angryMonkey\_crop.wav"; preload = true; } s1; } snd;

trial {

#trial\_duration = 15000;

picture {

ellipse\_graphic{

ellipse\_width =0.25; ellipse\_height = 0.25; color = 0, 255, 0;

}ellipse1;

x = 0; y = 0;

} circle\_fig;

duration = next\_picture;

} GreenDot;

trial {

#trial\_duration = 15000;

sound snd;

picture { text { caption = " "; } text3; x = 0; y = 0;} pic3;

duration = 1000;

} SoundTrial;

picture {

default\_code = "a"; # Always use this code for pic1...

default\_port\_code = 1; # Always use this port\_code for pic1...

# unless other values are given

bitmap { filename = "C:\\Documents and Settings\\USER\\Desktop\\Behavioral Control\\pics\_sounds\\angryMonkey\_crop.jpg";} bitmapa;

x = 0; y = 0;

} SeqPic;

picture {

default\_code = "a"; # Always use this code for pic1...

default\_port\_code = 1; # Always use this port\_code for pic1...

# unless other values are given

bitmap { filename = "C:\\Documents and Settings\\USER\\Desktop\\Behavioral Control\\pics\_sounds\\human\_crop.jpg";} bitmapOutLier; x = -100; y = 0;

bitmap { filename = "C:\\Documents and Settings\\USER\\Desktop\\Behavioral Control\\pics\_sounds\\angryMonkey\_crop.jpg";} bitmap2P; x = +100; y = 0;

ellipse\_graphic{ ellipse\_width =0.25; ellipse\_height = 0.25; color = 0, 255, 0; }ellipse2P; x = 0; y = 0;

} GreenDot2Pic;

picture {

default\_code = "a"; # Always use this code for pic1...

default\_port\_code = 1; # Always use this port\_code for pic1...

# unless other values are given

bitmap { filename = "C:\\Documents and Settings\\USER\\Desktop\\Behavioral Control\\pics\_sounds\\human\_crop.jpg";} bitmap2D7; x = +6; y = -6;

bitmap { filename = "C:\\Documents and Settings\\USER\\Desktop\\Behavioral Control\\pics\_sounds\\angryMonkey\_crop.jpg";} bitmap2D8; x = 0; y = -6;

ellipse\_graphic ellipse2P; x = 0; y = 0;

bitmap { filename = "C:\\Documents and Settings\\USER\\Desktop\\Behavioral Control\\pics\_sounds\\Black\_Spider\_Monkey\_crop.jpg";} bitmap2D1; x = -6; y = -6;

bitmap { filename = "C:\\Documents and Settings\\USER\\Desktop\\Behavioral Control\\pics\_sounds\\Elephant\_crop.jpg";} bitmap2D2; x = -6; y = 6;

bitmap { filename = "C:\\Documents and Settings\\USER\\Desktop\\Behavioral Control\\pics\_sounds\\Lizard\_crop.jpg";} bitmap2D3; x = 0; y = 6;

bitmap { filename = "C:\\Documents and Settings\\USER\\Desktop\\Behavioral Control\\pics\_sounds\\Peacock\_crop.jpg";} bitmap2D4; x = +6; y = 6;

bitmap { filename = "C:\\Documents and Settings\\USER\\Desktop\\Behavioral Control\\pics\_sounds\\Leopard\_crop.jpg";} bitmap2D5; x = +6; y = -6;

bitmap { filename = "C:\\Documents and Settings\\USER\\Desktop\\Behavioral Control\\pics\_sounds\\Turtle\_crop.jpg";} bitmap2D6; x = 0; y = -6;

# bitmap { filename = "C:\\Documents and Settings\\USER\\Desktop\\Behavioral Control\\pics\_sounds\\angryMonkey\_crop.jpg";} bitmap2D7; x = +900; y = 0;

} GreenDot2MPic;

picture {

default\_code = "a"; # Always use this code for pic1...

default\_port\_code = 1; # Always use this port\_code for pic1...

# unless other values are given

bitmap bitmap2D7; x = +6; y = -6;

bitmap bitmap2D8; x = 0; y = -6;

bitmap bitmap2D1; x = +300; y = 0;

bitmap bitmap2D2; x = +400; y = 0;

bitmap bitmap2D3; x = +500; y = 0;

bitmap bitmap2D4; x = +600; y = 0;

bitmap bitmap2D5; x = +700; y = 0;

bitmap bitmap2D6; x = +800; y = 0;

#bitmap bitmap2D7; x = +900; y = 0;

} SeqMPic;

trial {

picture GreenDot2MPic;

duration = next\_picture;

} GreenDotMPicTrial;

trial {

picture SeqMPic;

duration = next\_picture;

} SeqMPicTrial;

picture {

default\_code = "a"; # Always use this code for pic1...

default\_port\_code = 1; # Always use this port\_code for pic1...

# unless other values are given

bitmap bitmapOutLier; x = -100; y = 0;

bitmap bitmap2P; x = +100; y = 0;

} Seq2Pic;

trial {

picture GreenDot2Pic;

duration = next\_picture;

} GreenDot2PicTrial;

trial {

picture Seq2Pic;

duration = next\_picture;

} Seq2PicTrial;

trial {

#trial\_duration = 15000;

stimulus\_event { sound snd; } events1;

picture circle\_fig;

duration = next\_picture;

} GreenDotSound;

trial {

# wavefile s1;

stimulus\_event {

sound snd;

} events2;

picture SeqPic;

duration = next\_picture;

} PicnSound;

trial {

#trial\_duration = 15000;

picture {

ellipse\_graphic{

ellipse\_width =0.25; ellipse\_height =0.25; color = 0, 255, 0;

}ellipse6;

x = 0; y = 0;

ellipse\_graphic {

ellipse\_width =0.25; ellipse\_height =0.25; color = 0, 255, 0;

}ellipse2; x = -266; y = +100;

ellipse\_graphic {

ellipse\_width =0.25; ellipse\_height =0.25; color = 0, 255, 0;

}ellipse3; x = -266; y = +80;

ellipse\_graphic {

ellipse\_width =0.25; ellipse\_height = 0.25; color = 0, 255, 0;

}ellipse4; x = -266; y = -80;

ellipse\_graphic {

ellipse\_width =0.25; ellipse\_height = 0.25; color = 0, 255, 0;

}ellipse5; x = -266; y = -100;

} circle\_fig2;

duration = 1000;

} Green5Dot;

picture {

default\_code = "a"; # Always use this code for pic1...

default\_port\_code = 1; # Always use this port\_code for pic1...

# unless other values are given

ellipse\_graphic{

ellipse\_width =0.25; ellipse\_height = 0.25; color = 0, 255, 0;

}ellipse7; x = 0; y = 0;

ellipse\_graphic {

ellipse\_width =0.25; ellipse\_height = 0.25; color = 0, 255, 0;

}ellipse8; x = -266; y = +70;

ellipse\_graphic {

ellipse\_width =0.25; ellipse\_height = 0.25; color = 0, 255, 0;

}ellipse9; x = -266; y = +80;

ellipse\_graphic {

ellipse\_width =0.25; ellipse\_height = 0.25; color = 0, 255, 0;

}ellipse10; x = -266; y = -70;

ellipse\_graphic {

ellipse\_width =0.25; ellipse\_height = 0.25; color = 0, 255, 0;

}ellipse11; x = -266; y = -100;

bitmap { filename = "C:\\Documents and Settings\\USER\\Desktop\\Behavioral Control\\pics\_sounds\\angryMonkey\_crop.jpg";} bitmapd;

x = -266; y = 0;

} SeqPic2;

begin\_pcl;

int selN = 3;

int comError = 0;

array <int> selIDs[3] = { 3, 4 , 8 };

array <string> targetIDs[1];

array <string> soundName[1];

array <int> soundID[1];

array <int> randID[1];

array <string> picsName[1];

array <int> picsID[1];

int curI = 1; int history = 0;

array <double> CoorX[8] = { -7.5, 0.0, 7.5, -7.5, 7.5, -7.5, 0.0, 7.5 };

array <double> CoorY[8] = { 5.0, 5.0, 5.0, 0.0, 0.0, -5.0, -5.0, -5.0 };

/\*Display message on black screen\*/

sub

show\_text( string message )

begin

text1.set\_caption( message );

text1.redraw();

trial1.present()

end;

/\*read name of stimuli to present from file on disc\*/

sub

read\_Files( string filePath,array <string,1> & tempdata2,array <int,1> & tempIDdata2)

begin

input\_file in = new input\_file;

in.open(filePath); in.set\_delimiter( '\n' ); # for get\_line()

string d\_size=in.get\_string();

int dataSize = int(d\_size);

string junkVal=in.get\_line();

loop int count = 1; until in.end\_of\_file() || count > dataSize || !in.last\_succeeded()

begin

tempdata2.resize(count); tempIDdata2.resize(count);

tempdata2[count] = in.get\_line();

tempIDdata2[count] = count; # setting picture ID

#term.print(tempdata2[count] + string(count)+"of total : "+d\_size+"\n");

count = count + 1;

end;

end;

/\*read \n terminated string$\*/

sub

string read\_anyword(input\_port portin)

begin

#listen to arrival of data

string codes\_string=""; int code\_byte=0; int i=0; int limIndex=portin.total\_count();

#term.print("visiting read\_anyword");

#show\_text("visiting read\_anyword"+string(limIndex));

loop until code\_byte==10 || i>=limIndex begin

i = i+1;

codes\_string.resize(i);

code\_byte = portin.codes(i);

codes\_string.set\_char(i,code\_byte);

end;

if code\_byte == 10 then

portin.clear(1,i);

return codes\_string;

end;

#show\_text("received string:"+codes\_string+"& length is"+string(codes\_string.count()));

return "";

end;

/\*read data\*/

sub

int read\_word(input\_port portin, string awaitS)

begin

string codes\_string="";

loop until codes\_string == awaitS

begin

codes\_string=read\_anyword(portin)

end;

portin.clear();

return 1;

end;

sub

int PresentStimulus(string fname,string sName)

begin

bitmapa.unload();

bitmapa.set\_filename(fname);

#bitmapa.set\_load\_size(0.0,set\_load\_size(0.0,0.0,0.75);

bitmapa.load();

s1.unload();

s1.set\_filename(sName);

s1.load();

#term.print("Trial Start ");

#term.print(sName+":");

#term.print(fname+"\n");

PicnSound.present();

return 1;

end;

sub

int PresentStimulusLR(string fname,string sName)

begin

bitmapa.unload();

bitmapa.set\_filename(fname);

#bitmapa.set\_load\_size(0.0,0.0,0.75);

bitmapa.load();

#s1.unload();

#s1.set\_filename(sName);

#s1.load();

#term.print("Trial Start ");

#term.print(sName+":");

#term.print(fname+"\n");

SeqPic.present();

return 1;

end;

sub

double readPosReal(input\_port portin, string typeSt)

begin

string respText = "";

loop until respText.count()>0 begin

#show\_text("wait X");

respText = read\_anyword(portin);

#term.print("resp text is"+respText+"\n");

end;

respText.resize(respText.count()-1);

double xval = double(respText);

return xval;

end;

sub

int readPos(input\_port portin, string typeSt)

begin

double xval = readPosReal(portin,typeSt);

double yval = readPosReal(portin,typeSt);

#show\_text("Calibrating x @"+string(xval)+"and y @"+string(yval));

circle\_fig.set\_part\_x(1,xval);

circle\_fig.set\_part\_y(1,yval);

SeqPic.set\_part\_x(1,xval);

SeqPic.set\_part\_y(1,yval);

return 1;

end;

sub

int PresentStimulus2PG(input\_port portin, output\_port port, string fname, string tokenVal)

begin

double bitmap2P\_X=readPosReal(portin,tokenVal);

double bitmap2P\_Y=readPosReal(portin,tokenVal);

GreenDot2Pic.set\_part\_x(1,-bitmap2P\_X);

GreenDot2Pic.set\_part\_y(1,bitmap2P\_Y);

GreenDot2Pic.set\_part\_x(3,0); # GREEN DOT IN CENTER

GreenDot2Pic.set\_part\_y(3,0);

GreenDot2Pic.set\_part\_x(2,bitmap2P\_X); # CORRECT IMAGE

GreenDot2Pic.set\_part\_y(2,bitmap2P\_Y);

port.send\_string(tokenVal+"\_ACK\n");

GreenDot2PicTrial.present();

return 1;

end;

sub ############### change it to set position of pics

int PresentStimulusMPG(input\_port portin, output\_port port, string fname, string tokenVal)

begin

double bitmap2P\_X=readPosReal(portin,tokenVal);

double bitmap2P\_Y=readPosReal(portin,tokenVal);

#OUTLIER FIRST

GreenDot2MPic.set\_part\_x(1,CoorX[soundID[1]]);

GreenDot2MPic.set\_part\_y(1,CoorY[soundID[1]]);

#GreenDot2MPic.set\_part\_x(1,-bitmap2P\_X);

GreenDot2MPic.set\_part\_x(3,0); # GREEN DOT IN CENTER

GreenDot2MPic.set\_part\_y(3,0);

GreenDot2MPic.set\_part\_x(2,CoorX[soundID[2]]); # CORRECT IMAGE

GreenDot2MPic.set\_part\_y(2,CoorY[soundID[2]]);

#distractors setting

GreenDot2MPic.set\_part\_x(4,CoorX[soundID[3]]);

#GreenDot2MPic.set\_part\_x(5,0);

GreenDot2MPic.set\_part\_x(5,CoorX[soundID[4]]);

GreenDot2MPic.set\_part\_x(6,CoorX[soundID[5]]);

#GreenDot2MPic.set\_part\_x(8,0);

GreenDot2MPic.set\_part\_x(7,CoorX[soundID[6]]);

GreenDot2MPic.set\_part\_x(8,CoorX[soundID[7]]);

GreenDot2MPic.set\_part\_x(9,CoorX[soundID[8]]);

GreenDot2MPic.set\_part\_y(4,CoorY[soundID[3]]);

GreenDot2MPic.set\_part\_y(5,CoorY[soundID[4]]);

GreenDot2MPic.set\_part\_y(6,CoorY[soundID[5]]);

GreenDot2MPic.set\_part\_y(7,CoorY[soundID[6]]);

GreenDot2MPic.set\_part\_y(8,CoorY[soundID[7]]);

GreenDot2MPic.set\_part\_y(9,CoorY[soundID[8]]);

#GreenDot2MPic.set\_part\_y(8,-distractor\_X);

#GreenDot2MPic.set\_part\_y(9,-distractor\_X);

GreenDot2MPic.present();

port.send\_string(tokenVal+"\_ACK\n"+string(CoorX[selIDs[curI]])+"\n"+string(CoorY[selIDs[curI]])+"\n");

return 1;

end;

sub ############### change it to set position of pics

int PresentStimulusMP(input\_port portin,output\_port port,string tokenVal)

begin

double bitmap2P\_X=readPosReal(portin,tokenVal);

double bitmap2P\_Y=readPosReal(portin,tokenVal);

SeqMPic.set\_part\_x(1,CoorX[soundID[1]]);

SeqMPic.set\_part\_y(1,CoorY[soundID[1]]);

SeqMPic.set\_part\_x(2,CoorX[soundID[2]]);

SeqMPic.set\_part\_y(2,CoorY[soundID[2]]);

SeqMPic.set\_part\_x(3,CoorX[soundID[3]]);

#SeqMPic.set\_part\_x(5,0);

SeqMPic.set\_part\_x(4,CoorX[soundID[4]]);

SeqMPic.set\_part\_x(5,CoorX[soundID[5]]);

#SeqMPic.set\_part\_x(8,0);

SeqMPic.set\_part\_x(6,CoorX[soundID[6]]);

SeqMPic.set\_part\_x(7,CoorX[soundID[7]]);

SeqMPic.set\_part\_x(8,CoorX[soundID[8]]);

SeqMPic.set\_part\_y(3,CoorY[soundID[3]]);

SeqMPic.set\_part\_y(4,CoorY[soundID[4]]);

SeqMPic.set\_part\_y(5,CoorY[soundID[5]]);

SeqMPic.set\_part\_y(6,CoorY[soundID[6]]);

#SeqMPic.set\_part\_y(8,-distractor\_Y);

#SeqMPic.set\_part\_y(9,-distractor\_Y);

SeqMPic.set\_part\_y(7,CoorY[soundID[7]]);

SeqMPic.set\_part\_y(8,CoorY[soundID[8]]);

port.send\_string(tokenVal+"\_ACK\n");

SeqMPicTrial.present();

return 1;

end;

sub

int PresentStimulus2P(input\_port portin,output\_port port, string fname,string tokenVal)

begin

double bitmap2P\_X=readPosReal(portin,tokenVal);

double bitmap2P\_Y=readPosReal(portin,tokenVal);

#setting image positions

Seq2Pic.set\_part\_x(1,-bitmap2P\_X);

Seq2Pic.set\_part\_y(1,bitmap2P\_Y);

Seq2Pic.set\_part\_x(2,bitmap2P\_X);

Seq2Pic.set\_part\_y(2,bitmap2P\_Y);

port.send\_string(tokenVal+"\_ACK\n");

Seq2PicTrial.present();

return 1;

end;

sub int initialization

begin

# read corresponding wave files

# read corresponding pic files

read\_Files("C:\\Documents and Settings\\USER\\Desktop\\Behavioral Control\\object\_lists\\picture\_list.txt",picsName,picsID);

read\_Files("C:\\Documents and Settings\\USER\\Desktop\\Behavioral Control\\object\_lists\\sound\_list.txt",soundName,soundID);

# if mode is shuffled presentation, Shuffle order & output new order of pics

randID.resize(picsID.count());

loop int ik=1; until ik > picsName.count() begin

term.print(string(ik)+"th is No."+string(picsID[ik])+": "+picsName[soundID[ik]]+"\n");

randID[ik]=ik;

ik=ik+1;

end;

#picsID.shuffle();

soundID = picsID;

randID = soundID;

# initialize multipic placement according to the text file on desk

bitmap2D7.unload();

bitmap2D7.set\_filename(picsName[soundID[1]]);

bitmap2D7.load();

bitmap2D8.unload();

bitmap2D8.set\_filename(picsName[soundID[2]]);

bitmap2D8.load();

bitmap2D1.unload();

bitmap2D1.set\_filename(picsName[soundID[3]]);

bitmap2D1.load();

bitmap2D2.unload();

bitmap2D2.set\_filename(picsName[soundID[4]]);

bitmap2D2.load();

bitmap2D3.unload();

bitmap2D3.set\_filename(picsName[soundID[5]]);

bitmap2D3.load();

bitmap2D4.unload();

bitmap2D4.set\_filename(picsName[soundID[6]]);

bitmap2D4.load();

bitmap2D5.unload();

bitmap2D5.set\_filename(picsName[soundID[7]]);

bitmap2D5.load();

bitmap2D6.unload();

bitmap2D6.set\_filename(picsName[soundID[8]]);

bitmap2D6.load();

# if mode is N-shuffle presentation, Shuffle order N pics whose ID has been sent & output new order of pics

return 1;

end;

sub int randDist(int targetIDIndex)

begin

int distractorIDIndex=targetIDIndex;

# if mode is shuffled presentation, Shuffle order & output new order of pics

randID.shuffle();

loop int i=1 until soundID[distractorIDIndex] != selIDs[targetIDIndex] begin

distractorIDIndex=randID[i];

i=1+1;

end;

term.print("Distractor ID "+string(soundID[distractorIDIndex])+"\n"+"Target ID "+string(soundID[targetIDIndex])+"\n");

return distractorIDIndex;

end;

sub int randDistLoad(int targetIDIndex)

begin

int distractorIDIndex = randDist(targetIDIndex);

bitmapOutLier.unload();

#bitmap2P.set\_load\_size(0.0,0.0,0.75);

bitmapOutLier.set\_filename(picsName[distractorIDIndex]);

bitmapOutLier.load();

return 1;

end;

# read Nread\n,shuffle\nORNoshuffle, selN, selIDs for assocPic, LRPic, 2PIC, MPIC [MPIC not required but reading only for comaptibility]

sub int readStimuli(input\_port portin,output\_port port)

begin

string sceneType = read\_anyword(portin);

term.print("kahana\n");

comError = 0;

loop until read\_word(portin, "scenetype\n") == 1 begin

term.print("waiting scenetype");

comError = comError+1;

if (comError == 50000) then

port.send\_string("commError\_scenetype\n");

end;

end;

selN = int(readPosReal(portin,""));

selIDs.resize(selN);

term.print("Total "+string(selN)+"\n");

port.send\_string("scenetype\_ACK\n");

comError = 0;

loop int i=1 until i>selN begin

#reading single ID

int paRhliya=0;

loop until paRhliya>0 begin

paRhliya = int(readPosReal(portin,""));

#term.print(string(i)+"\n"+"length SelIDs"+string(selIDs.count()));

selIDs[i] = paRhliya;

term.print("waiting StimulusID\n");

comError = comError+1;

if (comError == 50000) then

port.send\_string("commError\_StimulusID\n");

end;

end;

term.print("Stimuli "+string(i)+" is: "+string(selIDs[i])+"\n");

term.print("Counting data\n");

port.send\_string("numb\_ACK\n");

i=i+1;

end;

#term.print("\nkaahaan ho");

return 1;

end;

if (output\_port\_manager.port\_count() == 0 && input\_port\_manager.port\_count() == 0 ) then

show\_text( "No output ports selected!" );

else

#Port initialization

output\_port port = output\_port\_manager.get\_port( 1 );

input\_port portin = input\_port\_manager.get\_port( 1 );

#port.set\_pulse\_width( 1 );

output\_port portLTP = output\_port\_manager.get\_port( 2 );

portLTP.set\_pulse\_width( 1 );

#Seq2Pic.present();

#GreenDot2Pic.present();

#List of globals

/\*Random presentation with picture ID sending\*/

#port.send\_string(string(picsID[1])+"\n");

/\* Present black screen, then green dot \*/

#GreenDot.present();

#PresentStimulus(picsName[picsID[1]]);

#Present pictures one-by-one

int i=0;

loop until 1==0 begin

#show\_text("Loop No: "+string(i));

string responesText = read\_anyword(portin);

#intialization phase

if responesText=="startr\n" then

#show\_text("strating");

picsID.resize(1); picsName.resize(1); i=1;

circle\_fig.set\_part\_x(1,0);

circle\_fig.set\_part\_y(1,0);

if initialization() == 1 then

port.send\_string("start\_ACK\n");

portLTP.send\_code( 255 );

readStimuli(portin,port);

# read Nread\n,shuffle\nORNoshuffle, selN, selIDs for assocPic, LRPic, 2PIC, MPIC [MPIC not required but reading only for comaptibility]

# shuffle or no shuffle... sets shuffleFlag=1, shuffles selIDs array

# send ACK "Nread\_ACK"

blackScreen.present();

#GreenDot.present();

# add utility to send "time of presentation"

#show\_text("Received list of pictures and sent ACK back");

else

show\_text("Error in Initialization");

port.send\_string("Start\_ERR\n"); # add utility to send "time of presentation"

end;

# Present next image

elseif responesText=="next\n" then

#show\_text(responesText);

if i<selIDs.count() then

i=i+1;

else

i=1;

selIDs.shuffle();

# shuffle seldIDs array if shuffleFlag=1

end;

#show\_text("Present Image"); #

readPos(portin,"next");

#port.send\_string(string(picsID[i])+"\n");

port.send\_string("next\_ACK\n");

portLTP.send\_code( 255 );

PresentStimulus(picsName[selIDs[i]],soundName[selIDs[i]]);

#term.print(picsName[picsID[i]]+"\n");

#portin.clear();

# Present same image

elseif responesText=="back\n" then

#break;

#show\_text("Re-start the loop... as Spike2 asks to come back \n bcaz monkey ignores the dot");

#show\_text(responesText);

readPos(portin,"back");

#port.send\_string(string(picsID[i])+"\n");

port.send\_string("back\_ACK\n");

portLTP.send\_code( 255 );

PresentStimulus(picsName[selIDs[i]],soundName[selIDs[i]]);

#portin.clear();

# Present green Dot

elseif responesText=="gren\n" then

#show\_text(responesText);

readPos(portin,"gren");

port.send\_string("gren\_ACK\n");

portLTP.send\_code( 255 );

GreenDot.present();

#port.send\_string(string(100)+"\n");

#show\_text("Pictures Finished!!!! at i value:"+string(i));

#port.send\_string("end\_ACK\n");

# Stop presenting

elseif responesText=="stop\n" then

port.send\_string("stop\_ACK\n");

portLTP.send\_code( 255 );

show\_text("Asked to be stopped");

show\_text(" ");

# Erroneous token presenting

elseif responesText=="blak\n" then

port.send\_string("blak\_ACK\n");

portLTP.send\_code( 255 );

blackScreen.present();

#show\_text("Nothing received");

#present old stuff again

# Erroneous token presenting

elseif responesText=="cali\n" then

readPos(portin,"gren");

#read coordinates for green dot

port.send\_string("cali\_ACK\n");

portLTP.send\_code( 255 );

GreenDot.present();

#show\_text("Calibrating");

elseif responesText=="nextLR\n" then

readPos(portin,"nextLR");

port.send\_string("nextLR\_ACK\n");

portLTP.send\_code( 255 );

PresentStimulusLR(picsName[selIDs[i]],soundName[selIDs[curI]]);

elseif responesText=="backLR\n" then

readPos(portin,"backLR");

port.send\_string("backLR\_ACK\n");

portLTP.send\_code( 255 );

PresentStimulusLR(picsName[selIDs[i]],soundName[selIDs[curI]]);

elseif responesText=="GNSDn\n" then

if i<selIDs.count() then

i=i+1;

else

i=1;

selIDs.shuffle();

end;

curI = i;

#show\_text(responesText);

s1.unload();

s1.set\_filename(soundName[selIDs[curI]]);

s1.load();

#code changed for MPIC

readPos(portin,"GNSDn");

port.send\_string("GNSDn\_ACK\n");

portLTP.send\_code( 255 );

GreenDotSound.present();

#Change target picture for anticipated change

bitmap2P.unload();

bitmap2P.set\_filename(picsName[selIDs[curI]]);

bitmap2P.load();

elseif responesText=="GNSDb\n" then

#show\_text(responesText);

readPos(portin,"GNSDb");

s1.unload();

s1.set\_filename(soundName[selIDs[i]]);

s1.load();

#code changed for MPIC

curI = i;

port.send\_string("GNSDb\_ACK\n");

portLTP.send\_code( 255 );

GreenDotSound.present();

#Change target picture for anticipated change

bitmap2P.unload();

bitmap2P.set\_filename(picsName[selIDs[curI]]);

bitmap2P.load();

elseif responesText=="G2PC\n" then

if history != curI then

randDistLoad(curI);

history =curI;

end;

PresentStimulus2PG(portin,port,picsName[selIDs[curI]],"G2PC");

portLTP.send\_code( 255 );

elseif responesText=="next2P\n" then

PresentStimulus2P(portin,port,picsName[selIDs[curI]],"next2P");

portLTP.send\_code( 255 );

elseif responesText=="back2P\n" then

PresentStimulus2P(portin,port,picsName[selIDs[curI]],"back2P");

portLTP.send\_code( 255 );

elseif responesText=="GMPC\n" then

PresentStimulusMPG(portin,port,picsName[selIDs[curI]],"GMPC");

portLTP.send\_code( 255 );

elseif responesText=="nextMP\n" then

curI = i;

PresentStimulusMP(portin,port,"nextMP");

portLTP.send\_code( 255 );

elseif responesText=="backMP\n" then

curI = i;

PresentStimulusMP(portin,port,"backMP");

portLTP.send\_code( 255 );

elseif responesText=="testing\n" then

double xt=readPosReal(portin,"testing");

double yt=readPosReal(portin,"testing");

double Tt=readPosReal(portin,"testing");

double Bt=readPosReal(portin,"testing");

double Lt=readPosReal(portin,"testing");

double Rt=readPosReal(portin,"testing");

SeqPic2.set\_part\_x(6,xt);

SeqPic2.set\_part\_y(6,yt);

show\_text("Pic x:"+string(xt)+"Pic y:"+string(yt));

SeqPic2.set\_part\_x(2,xt);

SeqPic2.set\_part\_y(2,Tt);

show\_text("Pic topx:"+string(xt)+"Pic topy:"+string(Tt));

SeqPic2.set\_part\_x(3,xt);

SeqPic2.set\_part\_y(3,Bt);

show\_text("Pic Botx:"+string(xt)+"Pic Boty:"+string(Bt));

SeqPic2.set\_part\_x(4,Lt);

SeqPic2.set\_part\_y(4,yt);

show\_text("Pic Leftx:"+string(Lt)+"Pic Lefty:"+string(yt));

SeqPic2.set\_part\_x(5,Rt);

SeqPic2.set\_part\_y(5,yt);

show\_text("Pic Rightx:"+string(Rt)+"Pic Righty:"+string(yt));

SeqPic2.set\_part\_x(1,xt);

SeqPic2.set\_part\_y(1,yt);

show\_text("Pic Centerx:"+string(xt)+"Pic Centery:"+string(yt));

portLTP.send\_code( 255 );

SeqPic2.present();

elseif responesText=="SNDDot\n" then

#show\_text(responesText);

readPos(portin,"SNDDot");

port.send\_string("SNDDot\_ACK\n");

portLTP.send\_code( 255 );

SoundTrial.present();

GreenDot.present();

#port.send\_string(string(100)+"\n");

#show\_text("Pictures Finished!!!! at i value:"+string(i));

#port.send\_string("end\_ACK\n");

# Stop presenting

else #<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<

term.print("Error Token : " + responesText + "\n"); #<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<

#blackScreen.present();

#show\_text("Un-expected token received: "+responesText);

#show\_text(" "+responesText);

#show\_text(" ");

#return 0;

end; #end of if

end;#end of loop

end;#end of main if