% initialize stuff

% create main figure

% Modify this to include your graphic...

close all

fig = figure();

ax = axes(fig);

ax.XLim = [0 9000];

ax.YLim = [0 5000];

ax.Color = [0 0 0];

ax.NextPlot = 'add';

% I=imread('k'); %insert background image

hi = imagesc(I)

% setup joystick or keyboard using Prof Donnal's code

joystick = KeyboardEmulator(fig);

% create cannon

% Modify this to change how the cannon looks

cannon = plot(ax,[0],[0]);

cannon.UserData.bx = 0; % base x position

cannon.UserData.by = 0; % base y position

cannon.UserData.l = 1000; % length

cannon.UserData.angle = 0;

cannon.set(...

'XData',[cannon.UserData.bx cannon.UserData.bx],... % XData

'YData',[cannon.UserData.by cannon.UserData.by+cannon.UserData.l], ... % YData

'Color','w',...

'LineWidth',5); % draw as a fat wide line

mycolors = [0.2 0.6 1; 0 0.9 0; 1 1 1; 1 0.6 0.2; 0.4 0 0.6; 1 0 1];

% create targets

planets = gobjects(1,1);

for i = 1:size(planets,1)

for j = 1:size(planets,2)

p = plot(ax,[0],[0],'o');

p.Color = mycolors(randi([1 6]),:); %rand(1,3); %changed to blue from rand(1,3)

p.MarkerFaceColor = p.Color;

p.MarkerSize = 25;

p.XData = 8000;

p.YData = 2500;

p.Visible = 'on';

planets(i,j) = p;

end

end

% create bullet

bullet = plot(ax,[700],[50],'ko');

bullet.MarkerFaceColor = mycolors(randi([1 6]),:); %rand(1,3); %changed to blue

bullet.MarkerSize = 25;

bullet.Visible = 'off';

bullet.UserData.v = 1400/1; % 1400px over five seconds

bullet.UserData.vx = 0;

bullet.UserData.vy = 0;

bullet.UserData.exploded = 0;

cannon.MarkerFaceColor = bullet.MarkerFaceColor;

% score

score = 0;

scoreboard = text(0,25,['SCORE: ',num2str(score)]);

scoreboard.Color = 'w';

remaining = 60;

timeboard = text(950,25,['TIME LEFT: ',num2str(remaining)]);

timeboard.Color = 'w';

nameboard = text(538, 50, 'SPACE BALLS');

nameboard.Color = 'w';

% main game loop

dt = 1/20;

tic

% upload sounds

[y, Fs] = audioread('pop2.wav');

pop2\_sound = audioplayer(y, Fs);

while(remaining>0)

% State 1: aiming

while ~joystick.btnstate(1) && (remaining>0);

disp('aiming');

% update angle according to joystick or keyboard

cannon.UserData.angle = cannon.UserData.angle+90/20\*joystick.jlx;

% limit cannon motion

if cannon.UserData.angle > 75

cannon.UserData.angle = 75;

elseif cannon.UserData.angle <-75

cannon.UserData.angle = -75;

end

% actually move the cannon

cannon.set(...

'XData',[cannon.UserData.bx cannon.UserData.bx+cannon.UserData.l\*sind(cannon.UserData.angle)],...

'YData',[cannon.UserData.by cannon.UserData.by+cannon.UserData.l\*cosd(cannon.UserData.angle)])

pause(dt);

remaining = 60-toc;

timeboard.String = ['TIME REMAINING: ',num2str(round(remaining))];

if remaining<10

timeboard.Color = 'r';

end

end

% State 2: firing (if you hit space or button 1)

disp('firing');

% update the bullet

bullet.XData = cannon.XData(2);

bullet.YData = cannon.YData(2);

bullet.UserData.vx = bullet.UserData.v\*sind(cannon.UserData.angle);

bullet.UserData.vy = bullet.UserData.v\*cosd(cannon.UserData.angle);

bullet.UserData.exploded = 0;

bullet.Visible = 'on';

pause(dt);

% State 3: bullet is in flight, ends when bullet explodes

while ~bullet.UserData.exploded & (remaining>0)

disp('bullet in flight');

vx = 200; vy0=vx;

x = vx\*t;

y = 1/2\*(-9.81)\*t.^2+vy0\*t+0;

bullet.XData = bullet.XData + bullet.UserData.vx\*dt;

bullet.YData = bullet.YData + bullet.UserData.vy\*dt;

% loop to animate

for i=1:length(t)

% at each step, move the cannonball to the right place

end

% % reflect off right and left walls

%if (max(bullet.XData)>1400) || (min(bullet.XData)<0)

% bullet.UserData.vx = -bullet.UserData.vx;

%end

% explode if you hit the wall

if max((bullet.XData)>9000 | (bullet.YData)>5000)

bullet.UserData.exploded = 1;

end

% check for collision with targets

for i=1:size(planets,1)

for j=1:size(planets,2)

if (strcmp(planets(i,j).Visible,'on') && ...

(sqrt((bullet.XData-planets(i,j).XData)^2+(bullet.YData-planets(i,j).YData)^2)<75))

if all(bullet.MarkerFaceColor == planets(i,j).MarkerFaceColor) %recently added

planets(i,j).Visible = 'off';

bullet.UserData.exploded = 1;

disp('hit!');

score = score + 1;

scoreboard.String = ['SCORE: ',num2str(score)];

break;

end %recently added

end

end

if bullet.UserData.exploded == 1

pop2\_sound.play;

break;

end

end

pause(dt);

remaining = 60-toc;

timeboard.String = ['TIME REMAINING: ',num2str(round(remaining))];

if remaining<10

timeboard.Color = 'r';

end

end

% bullet is done, reset it and go back to aiming

bullet.UserData.exploded = 0;

bullet.MarkerFaceColor = mycolors(randi([1 6]),:);

cannon.Color = bullet.MarkerFaceColor;

bullet.Visible = 'off';

end