% initialize stuff

% create main figure

% Modify this to include your graphic...

%%

fig1 = figure(1);

ax = axes(fig1);

xlabel('Academy Raft Wars')

ax.XLim = [0 9000];

ax.YLim = [0 5000];

grid off

%axis off

Background = figure(2);

BR=imread('beach2.jpeg');

hbg = imshow(BR);

hbg.Parent = ax;

set(hbg, 'XData',[0, 9000]);

set(hbg,'YData',[5000, 0]);

close(Background);

GoodGuy = figure(3);

GG = imread('raft1.png');

h= imshow(GG);

h.Parent = ax;

set(h, 'XData', [0, 1000]);

set(h, 'YData', [1000, 0]);

close(GoodGuy);

BadGuy = figure(4);

BG = imread('enemy1.png');

h2= imshow(BG);

h2.Parent = ax;

set(h2, 'XData', [9000, 8000])

set(h2, 'YData', [1000, 0])

close(BadGuy);

%%

% setup joystick or keyboard using Prof Donnal's code

joystick = KeyboardEmulator(fig1);

% create cannon

% Modify this to change how the cannon looks

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

cannon.Parent = ax;

cannon.UserData.bx = 4500; % base x position

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

cannon.UserData.l = 3000; % length

cannon.UserData.angle = 45;

cannon.set(...

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

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

'Color','k',...

'LineWidth',10); % draw as a fat wide line

% create targets

planets = gobjects(1,1);

for i = 1:size(planets,1)

for j = 1:size(planets,1)

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

p.Parent = ax;

p.Color = 'r';

p.MarkerFaceColor = p.Color;

p.MarkerSize = 100;

p.XData = 8000+(j-1);

p.YData = 2000+(i-1);

p.Visible = 'on';

planets(i,j) = p;

end

end

% create bullet

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

bullet.Parent = ax;

bullet.MarkerFaceColor = 'r'; %rand(1,3); %changed to blue

bullet.MarkerSize = 25;

bullet.Visible = 'off';

bullet.UserData.v = 5000; % 1400px over five seconds

bullet.UserData.vx = 0;

bullet.UserData.vy = 0;

bullet.UserData.exploded = 0;

% 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');

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

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

bullet.UserData.vy = bullet.UserData.vy - 1000\*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,1)

%dist = sqrt((bullet.XData-planets(i,j).XData)^2+(bullet.YData-planets(i,j).YData)^2);

%disp(dist);

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

(sqrt((bullet.XData-planets(i,j).XData)^2+(bullet.YData-planets(i,j).YData)^2)<(1000)))

%if (bullet.Xdata == 8000 && bullet.YData == 1000)

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.Visible = 'off';

end