```
%% Function graphs the animation of trajectory
function S = Trajectory(v,A,tf,dismax,hf,hmax) %defines the funcition and inputs

figure %creats a seperate plot

hold on %holds the image throughout the animation
img=imread('trees.jpeg'); %reads the image
min_x=0;      %Defines the boundary for image
max_x=dismax; %Defines the boundary for image
min_y=hf;     %Defines the boundary for image
max_y=hmax+1; %Defines the boundary for image
image('CData',flipud(img),'XData',[min_x max_x],'YData',[min_y max_y]) %Sets the boundary for
image
hold on %holds the image throughout the animation

for t4=0:.05:tf %creates a loop with time interval from 0 to Final time
    g=9.81; %gravitational Constant
    xlim([0 dismax]) %Sets x-axis limit from 0 to Final Time
    if hf<0 %if landing height is less than 0
        ylim([hf hmax+1]) %Sets y-axis limit from landing Height to max height+1
    else %if landing height is not less than 0
        ylim([0 hmax+1]) %Sets y-axis limit from 0 to max height+1
    end
    distance4=v*t4.*cosd(A); %distance formula
    h4=v*t4.*sind(A)-((g)*t4.^2)/2; %height formula
    plot(distance4,h4,'.','MarkerSize',10) %plots a dot
    title('Trejectory') %add title to the grah
    xlabel('Horizontal Distance (m)'); %lables x-axis
    ylabel('Vertical Height(m)'); %lables y-axis
    hold on %leaves the old points on
    pause(.002)%pauses the graph for a bit between each go through the loop
               %which allows it to look animated
end

hold off %Allows the figure to be override for next run

end
```