

# AstroVolley:

## Plotting & App Building

### Instructor Guide

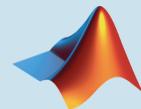


An astrodynamics game where navigators attempt to deliver payloads between star cruisers by volleying them through fantastical gravity fields.

# Course Lesson Plan

- Installing and Playing AstroVolley
  - Plotting Ships (Triangles)
  - Plotting Grav. Bodies (Circles)
  - Gravity and Orbits
  - Launching Volleys
    - Creating a Mission Map Function
    - Plotting Velocity Vectors (quiver)
    - Animating Trajectories (comet)
- App Designer**
- 1) UIAxes, Properties, & Functions
  - 2) Sliders & Value Changing Callback
  - 3) Push Button Callback
  - 4) Generating Random Maps
  - 5) Keeping Score

# Course Lesson Plan



## Installing and Playing AstroVolley

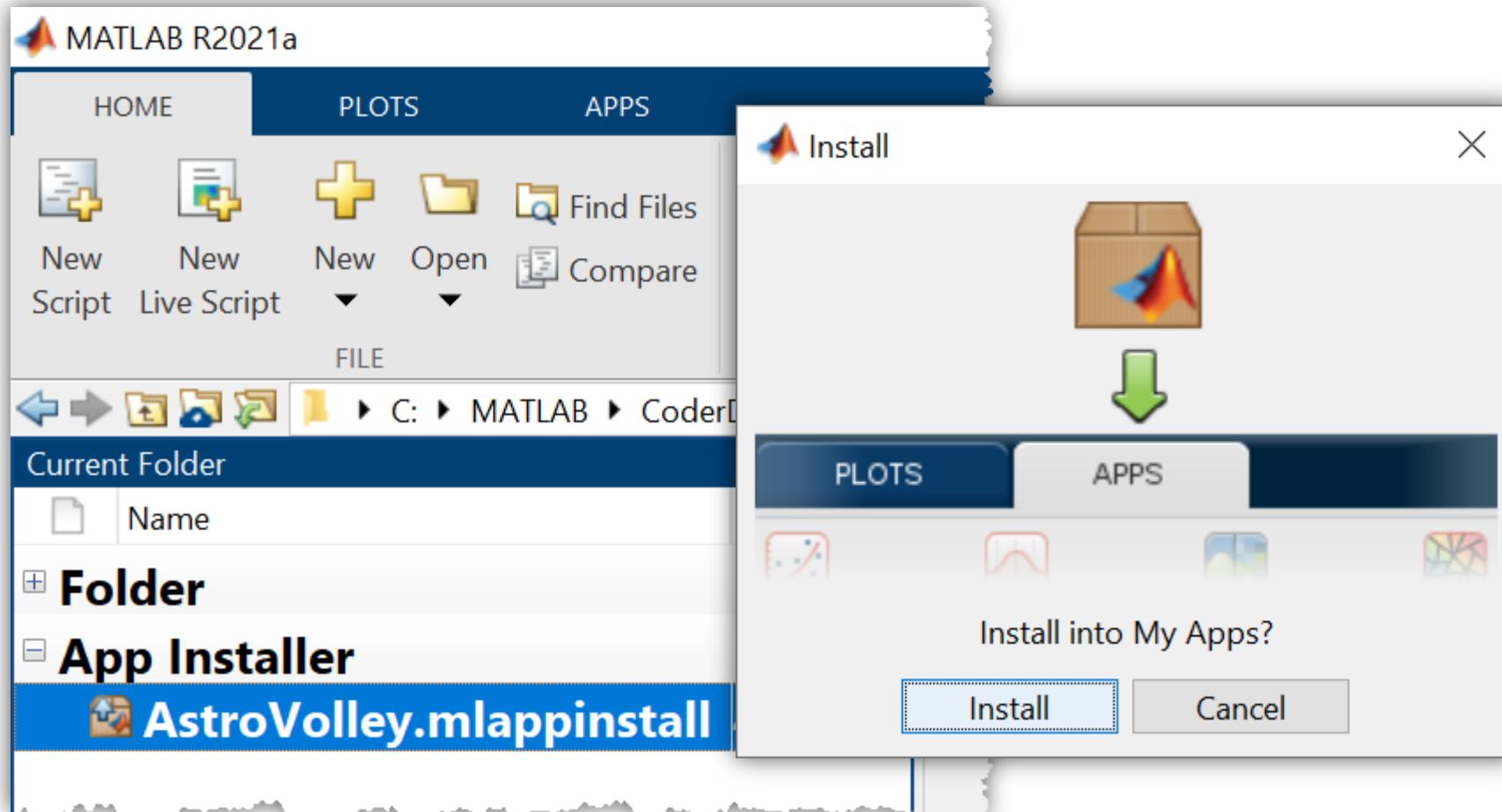
- Plotting Ships (Triangles)
- Plotting Grav. Bodies (Circles)
- Gravity and Orbits
- Launching Volleys
  - Creating a Mission Map Function
  - Plotting Velocity Vectors (quiver)
  - Animating Trajectories (comet)

## App Designer

- 1) UIAxes, Properties, & Functions
- 2) Sliders & Value Changing Callback
- 3) Push Button Callback
- 4) Generating Random Maps
- 5) Keeping Score

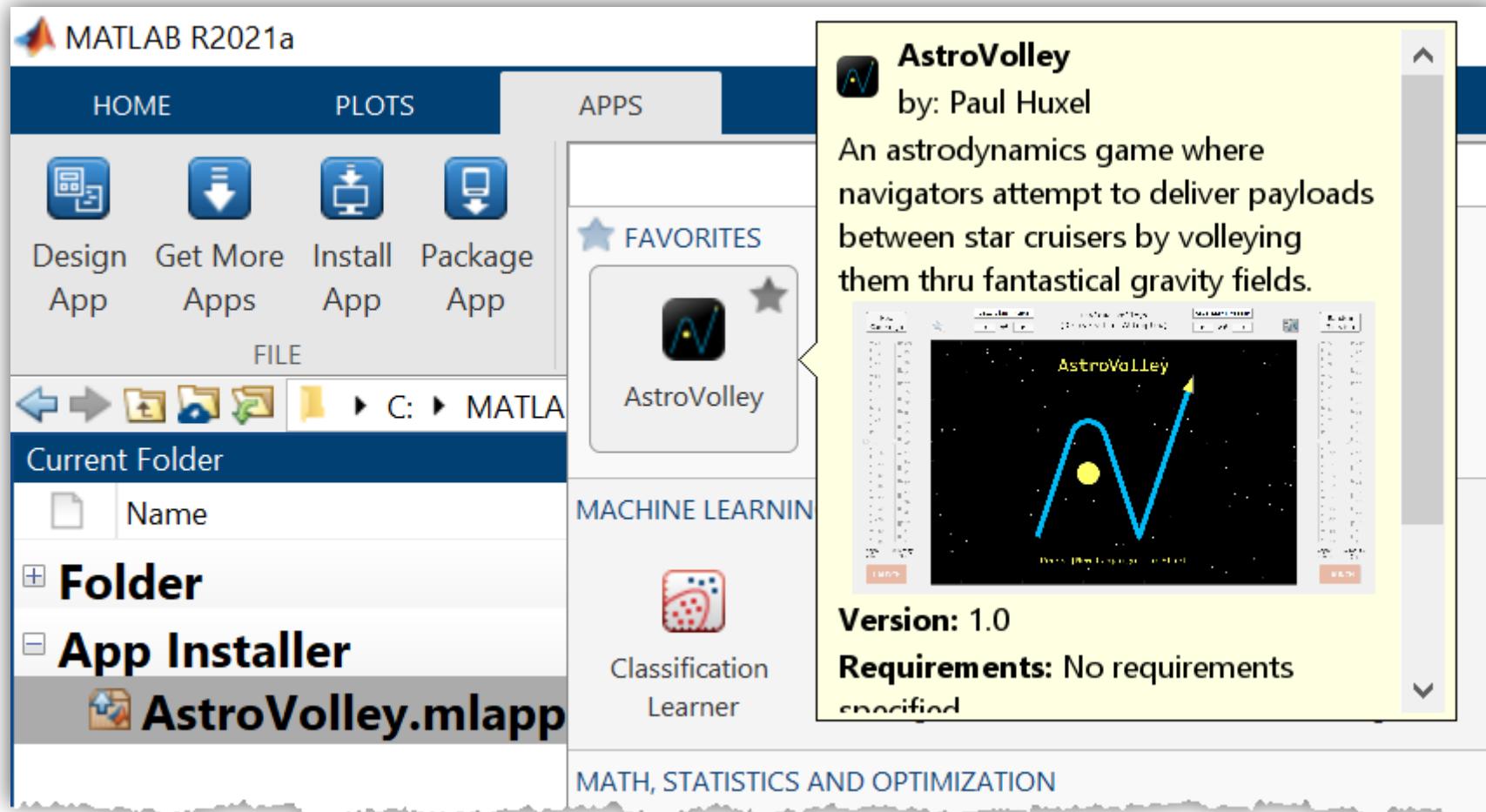
# Installing AstroVolley

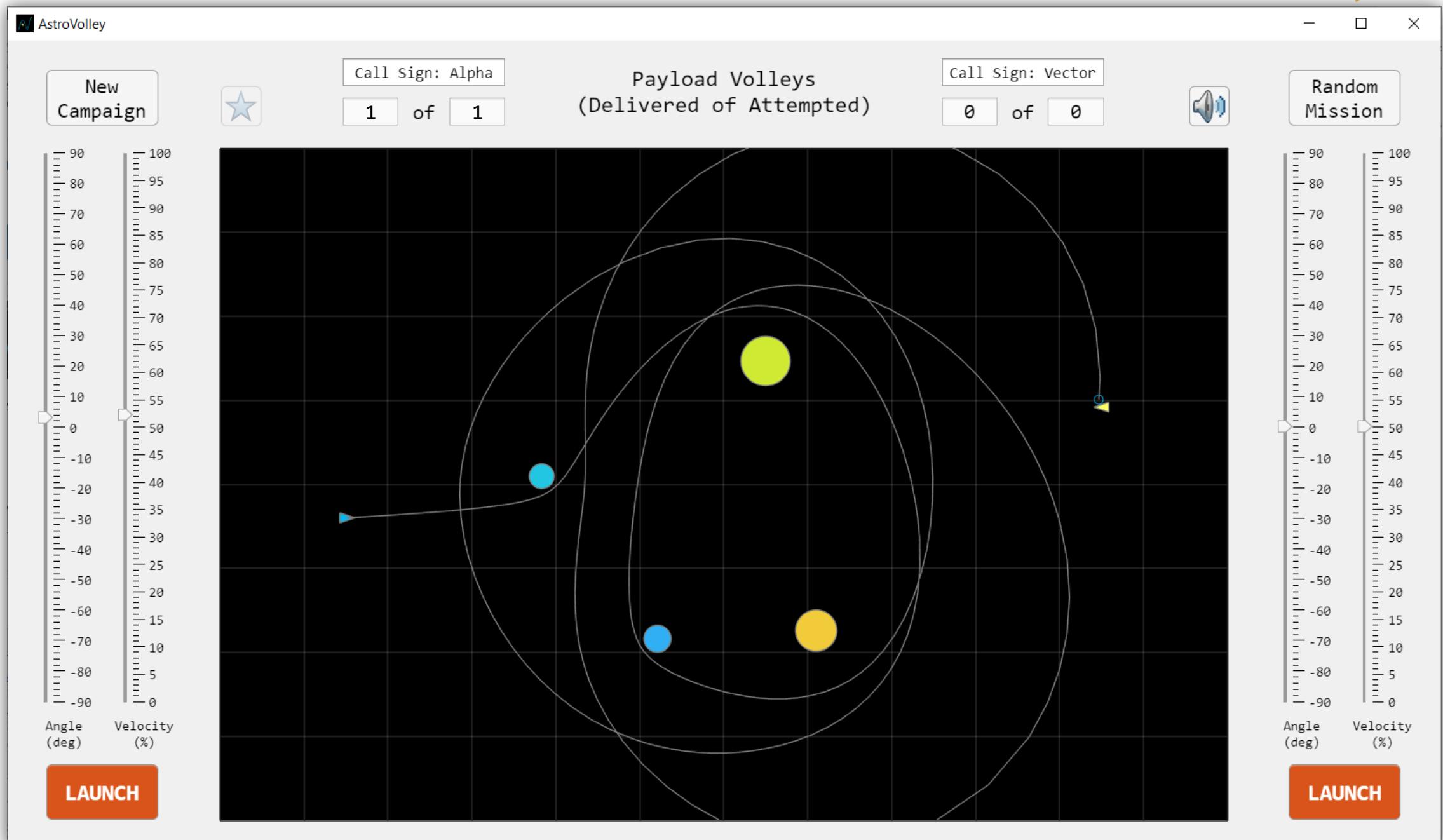
Navigate to and double-click on "AstroVolley.mlappinstall"



# Playing AstroVolley

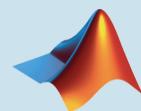
Find and select AstroVolley from the APPS tab





# Course Lesson Plan

- Installing and Playing AstroVolley



## Plotting Ships (Triangles)

- Plotting Grav. Bodies (Circles)
- Gravity and Orbits
- Launching Volleys
  - Creating a Mission Map Function
  - Plotting Velocity Vectors (quiver)
  - Animating Trajectories (comet)

## App Designer

- 1) UIAxes, Properties, & Functions
- 2) Sliders & Value Changing Callback
- 3) Push Button Callback
- 4) Generating Random Maps
- 5) Keeping Score

# Represent game pieces using basic shapes...



Image Credit: NASA  
GSFC\_20171208  
Archive\_e002131

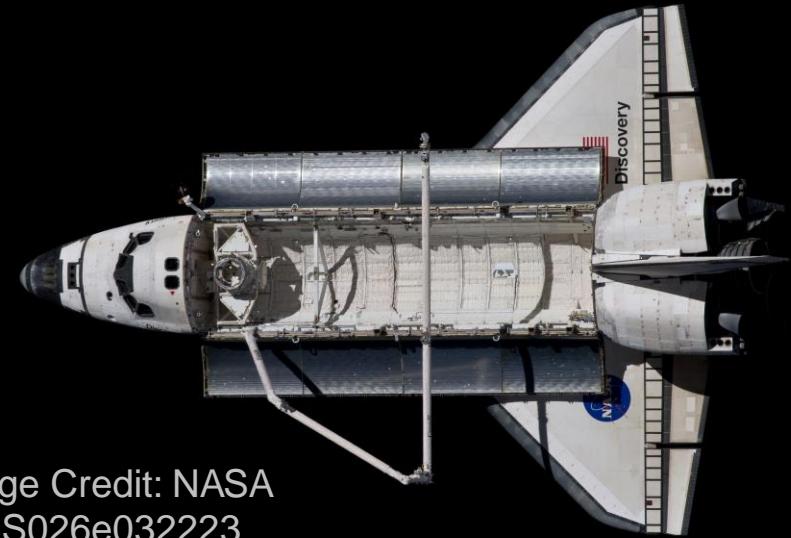


Image Credit: NASA  
ISS026e032223

# Represent game pieces using basic shapes...

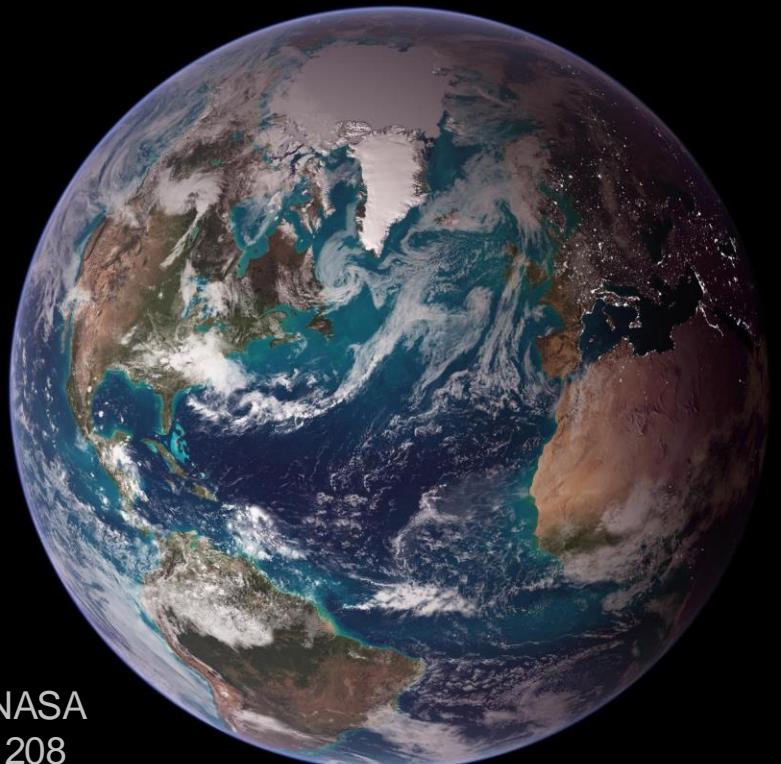


Image Credit: NASA  
GSFC\_20171208  
Archive\_e002131

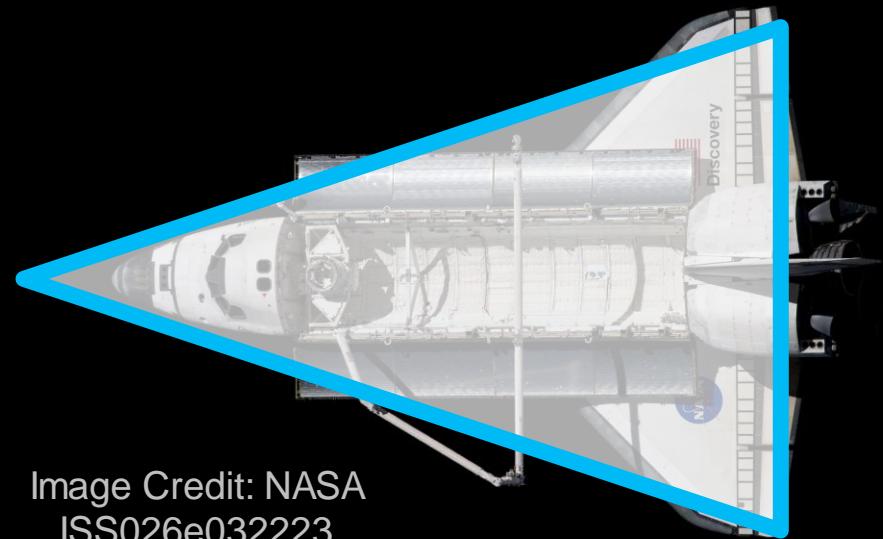
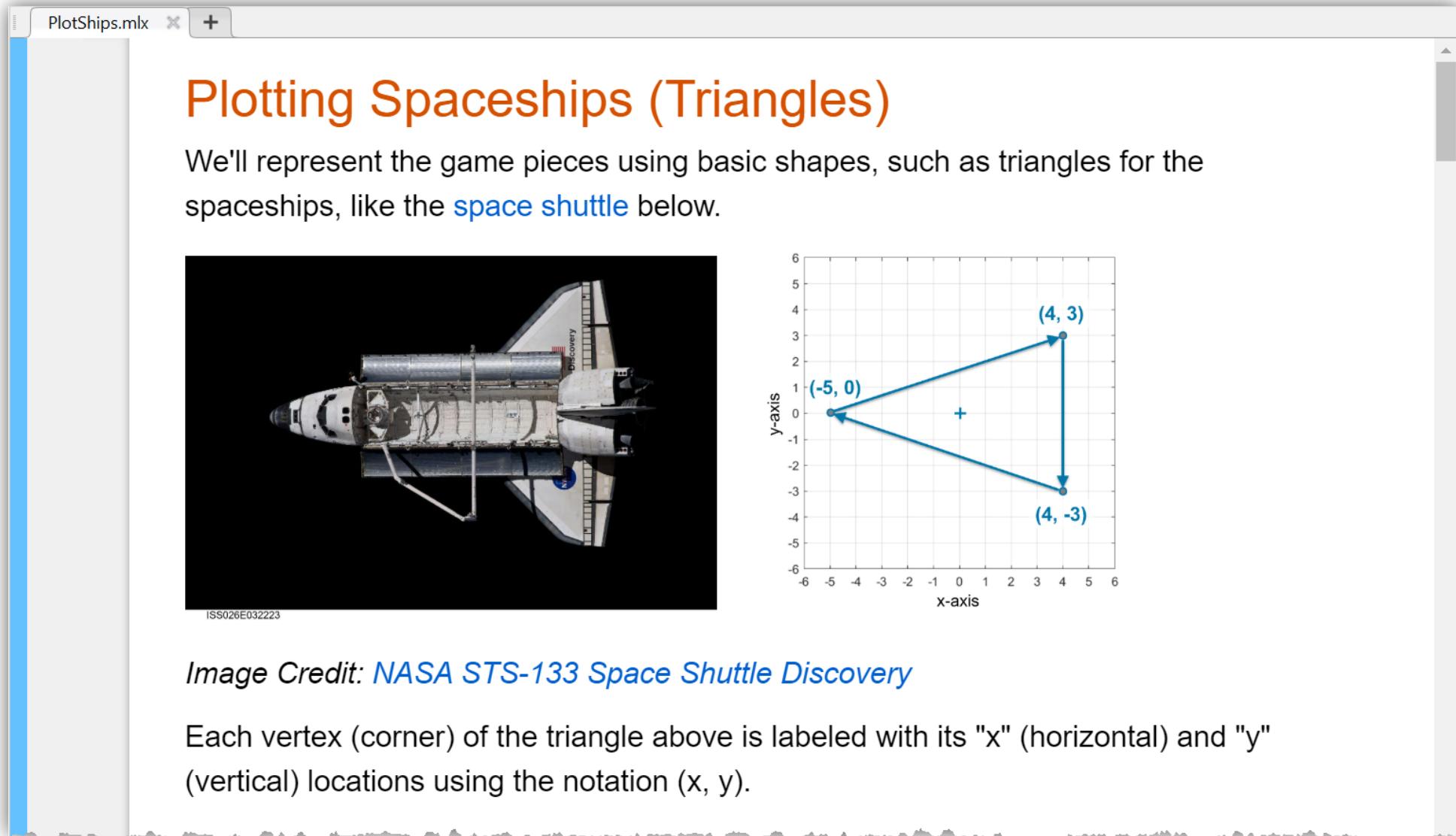


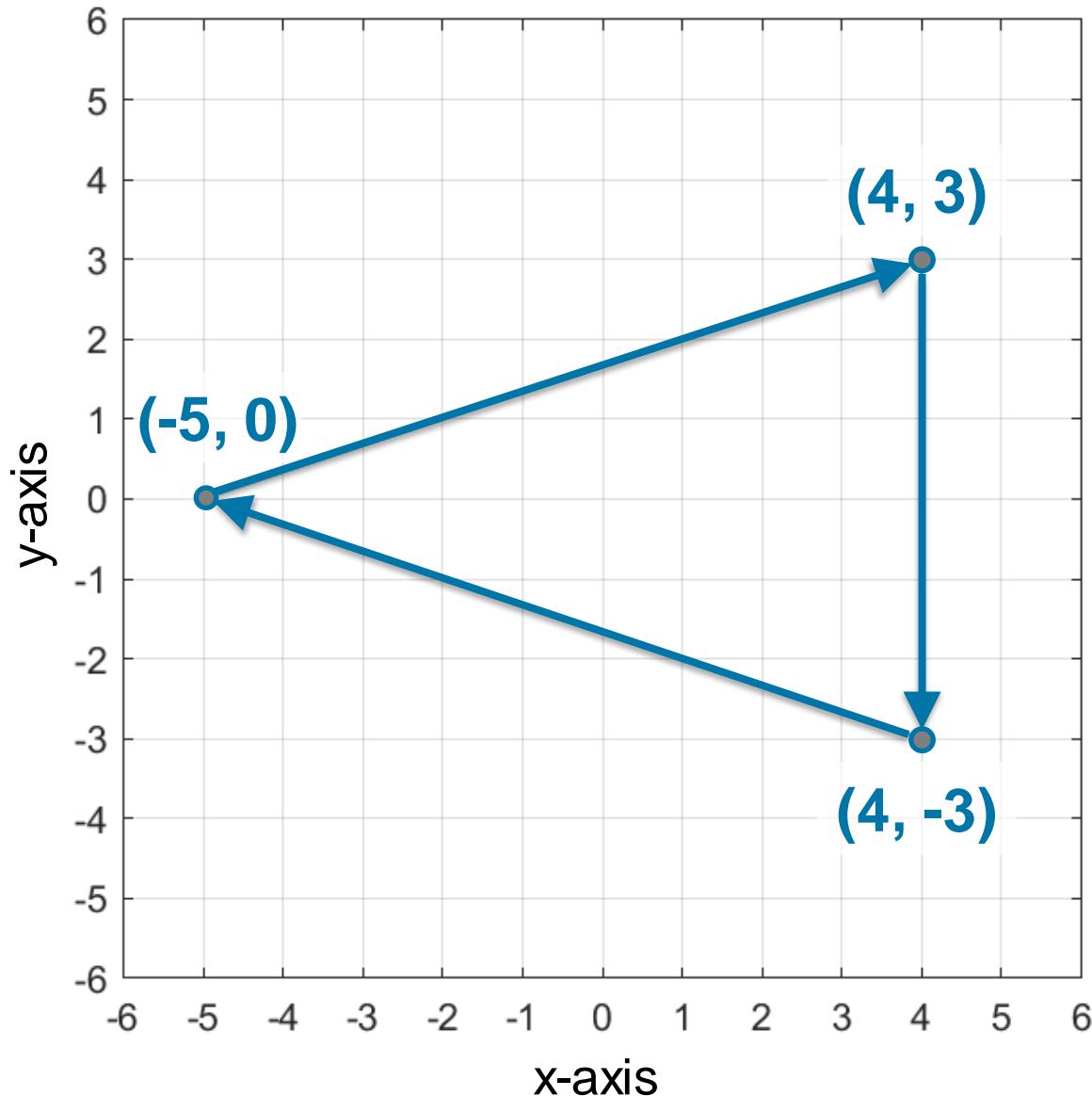
Image Credit: NASA  
ISS026e032223

Triangles for the  
Spaceships  
(Space Shuttle)

```
>> open PlotShips mlx
```

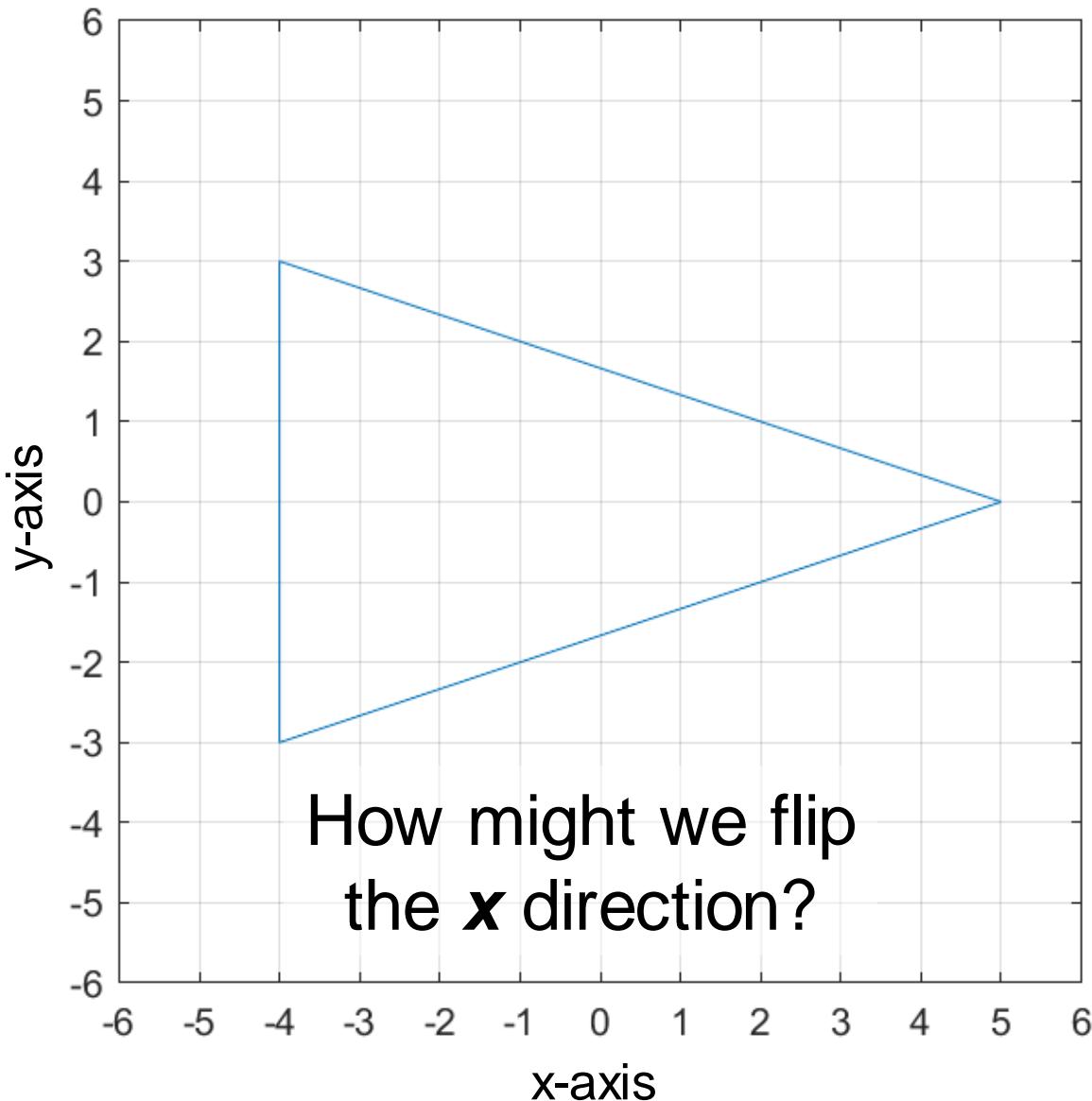


# Plotting Spaceships (Triangles)



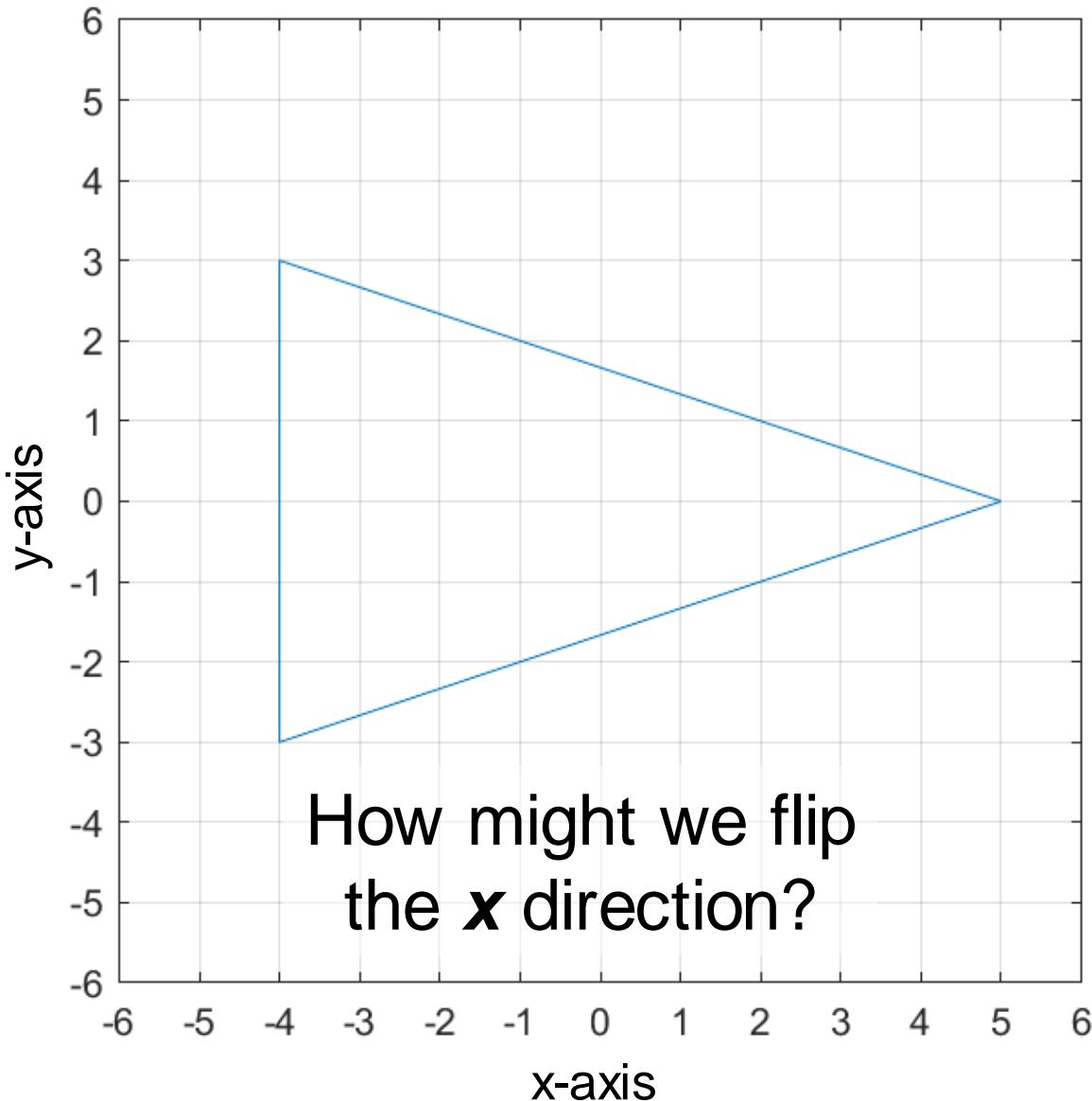
```
>> xt = [-5 4 4 -5]; % tri x-pts  
  
>> yt = [ 0 3 -3 0]; % tri y-pts  
  
>> ax = gca; % get current axes  
  
>> plot(ax,xt,yt)
```

# Plotting Spaceships (Triangles)



```
>> xt = [-5 4 4 -5]; % tri x-pts  
  
>> yt = [ 0 3 -3 0]; % tri y-pts  
  
>> ax = gca; % get current axes  
  
>> plot(ax, xt,yt) % point < left
```

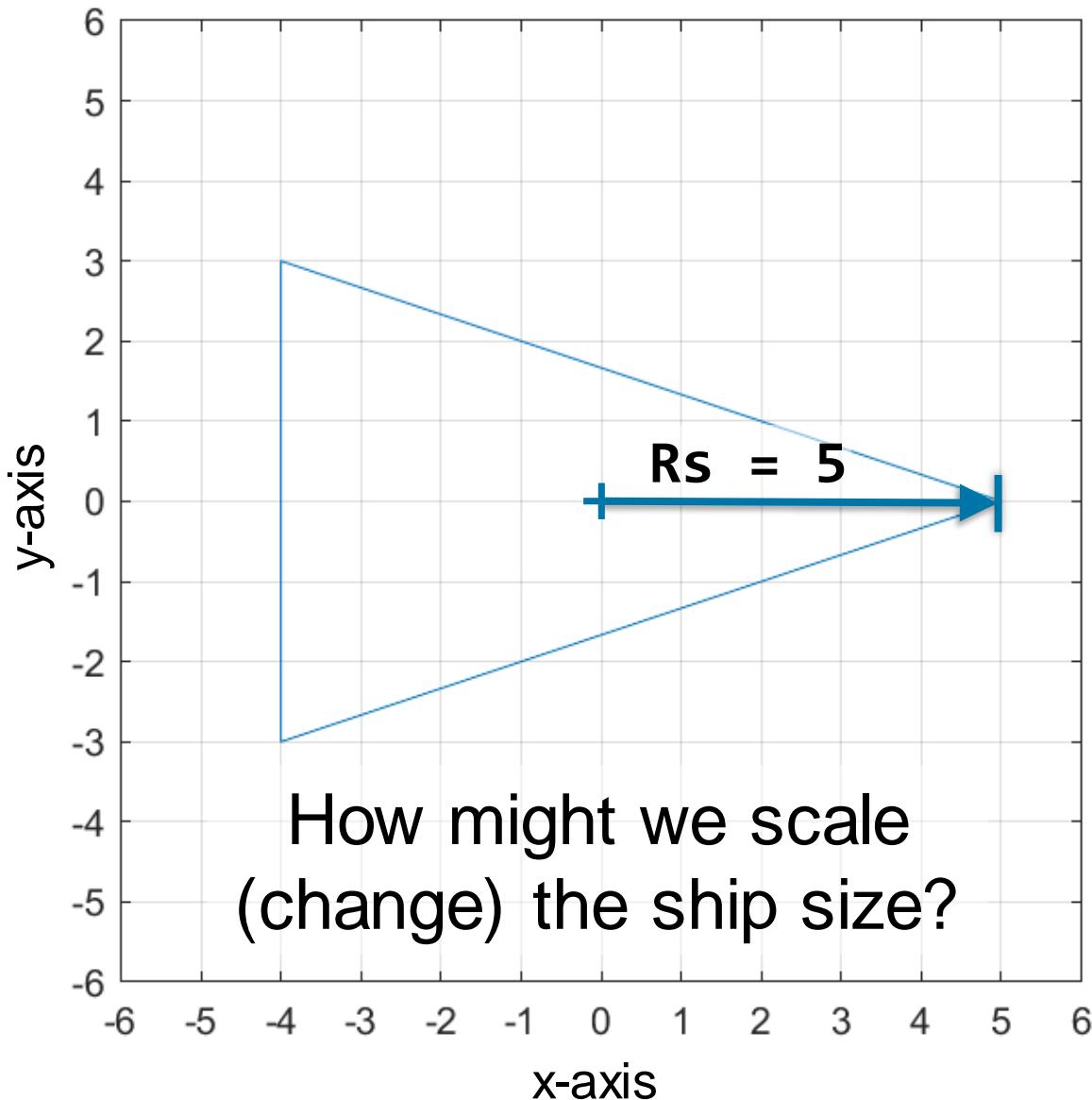
# Plotting Spaceships (Triangles)



```
>> xt = [-5 4 4 -5]; % tri x-pts  
  
>> yt = [ 0 3 -3 0]; % tri y-pts  
  
>> ax = gca; % get current axes  
  
>> plot(ax, xt,yt) % point < left  
  
>> plot(ax, -xt,yt) % point > right
```

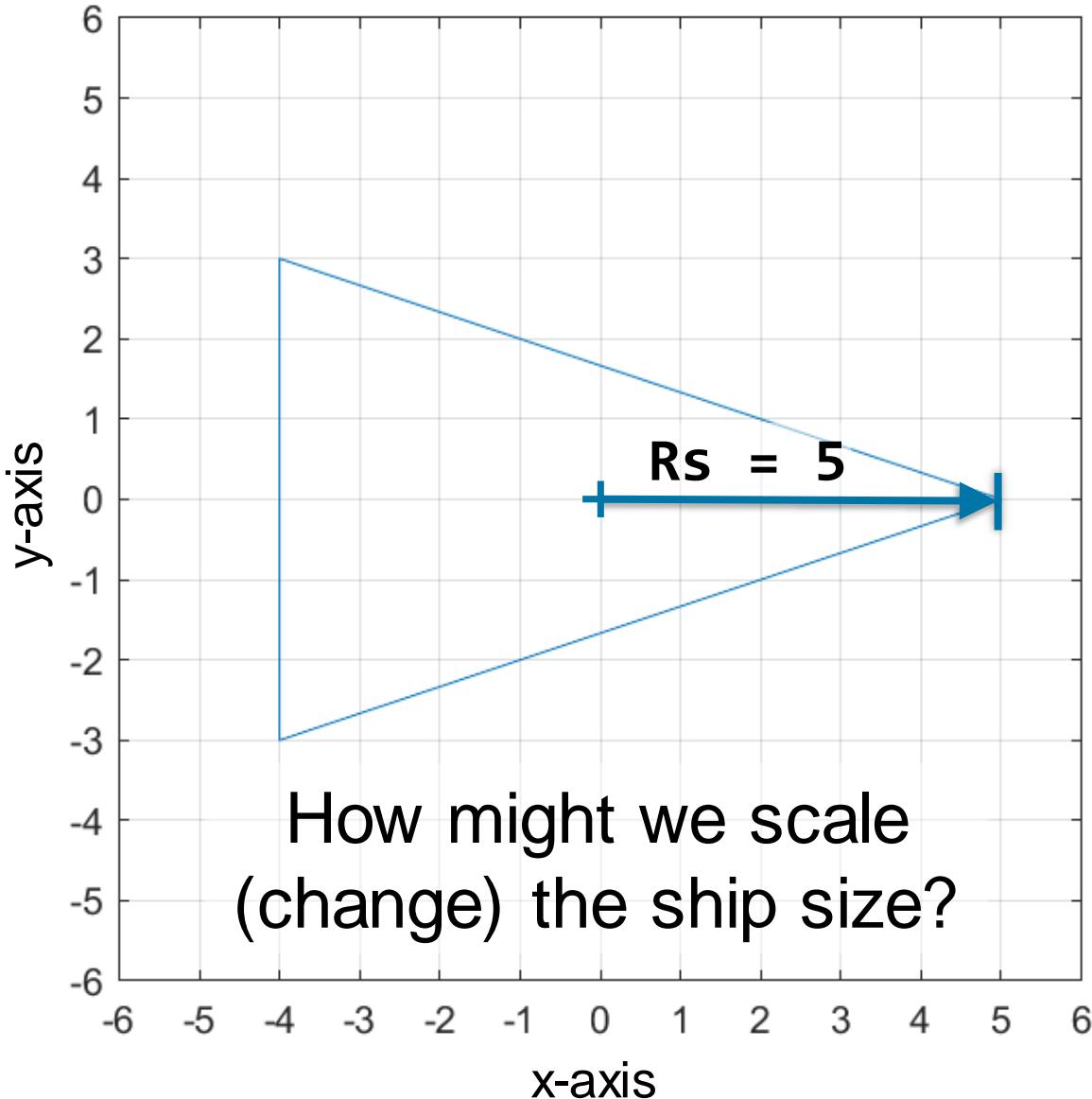


# Plotting Spaceships (Triangles)



```
>> xt = [-5 4 4 -5]; % tri x-pts  
  
>> yt = [ 0 3 -3 0]; % tri y-pts  
  
>> ax = gca; % get current axes  
  
>> plot(ax, xt,yt) % point < left  
  
>> plot(ax,-xt,yt) % point > right
```

# Plotting Spaceships (Triangles)



```
>> Rs = 5;
```

% ship size

```
>> xt = [-5 4 4 -5]/5; % tri x-pts
```

```
>> yt = [ 0 3 -3 0]/5; % tri y-pts
```

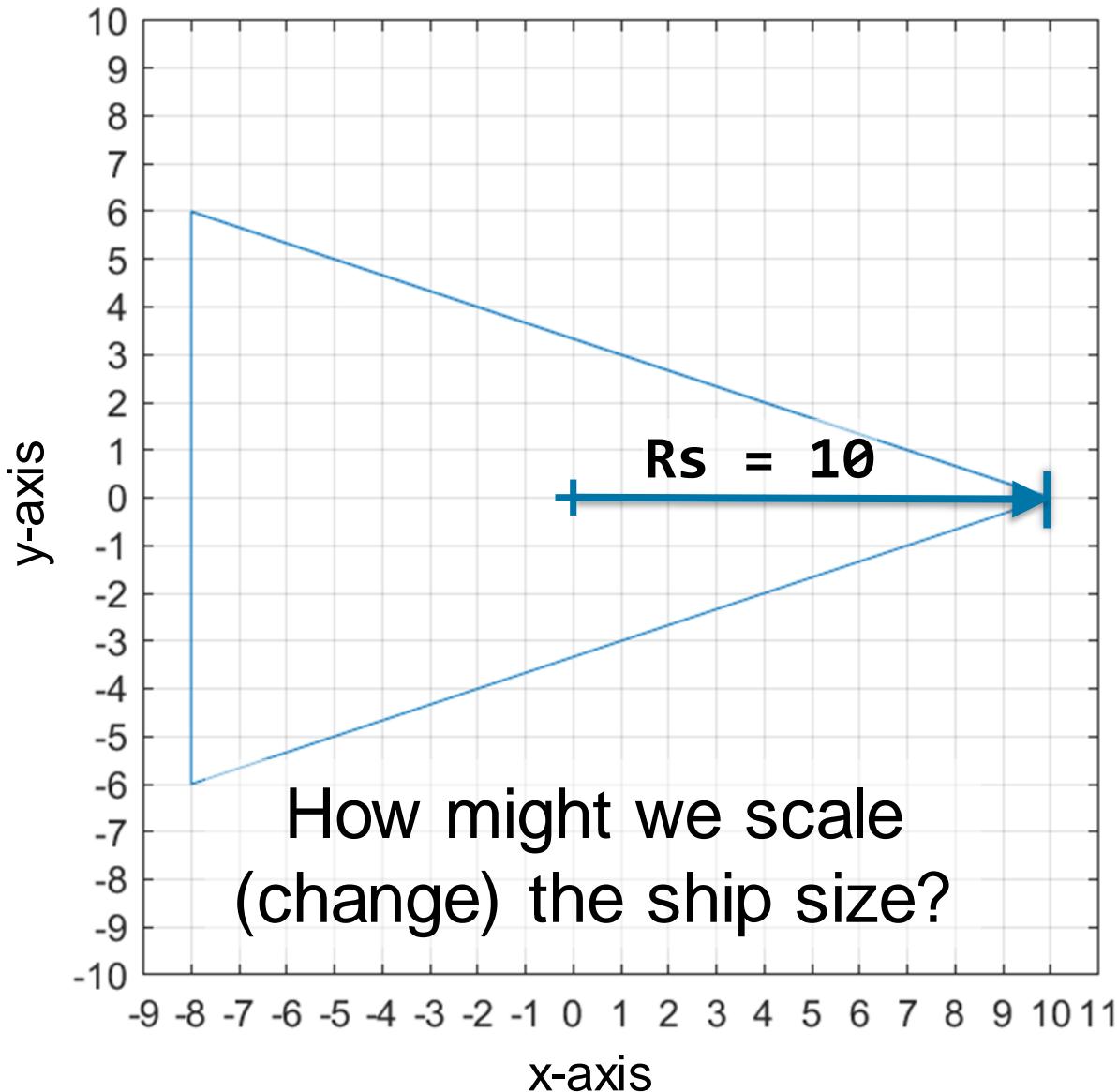
```
>> ax = gca; % get current axes
```

```
>> plot(ax, Rs*xt,Rs*yt) % < left
```

```
>> plot(ax, -Rs*xt,Rs*yt) % > right
```



# Plotting Spaceships (Triangles)



```
>> Rs = 10;
```

% ship size

```
>> xt = [-5 4 4 -5]/5; % tri x-pts
```

```
>> yt = [ 0 3 -3 0]/5; % tri y-pts
```

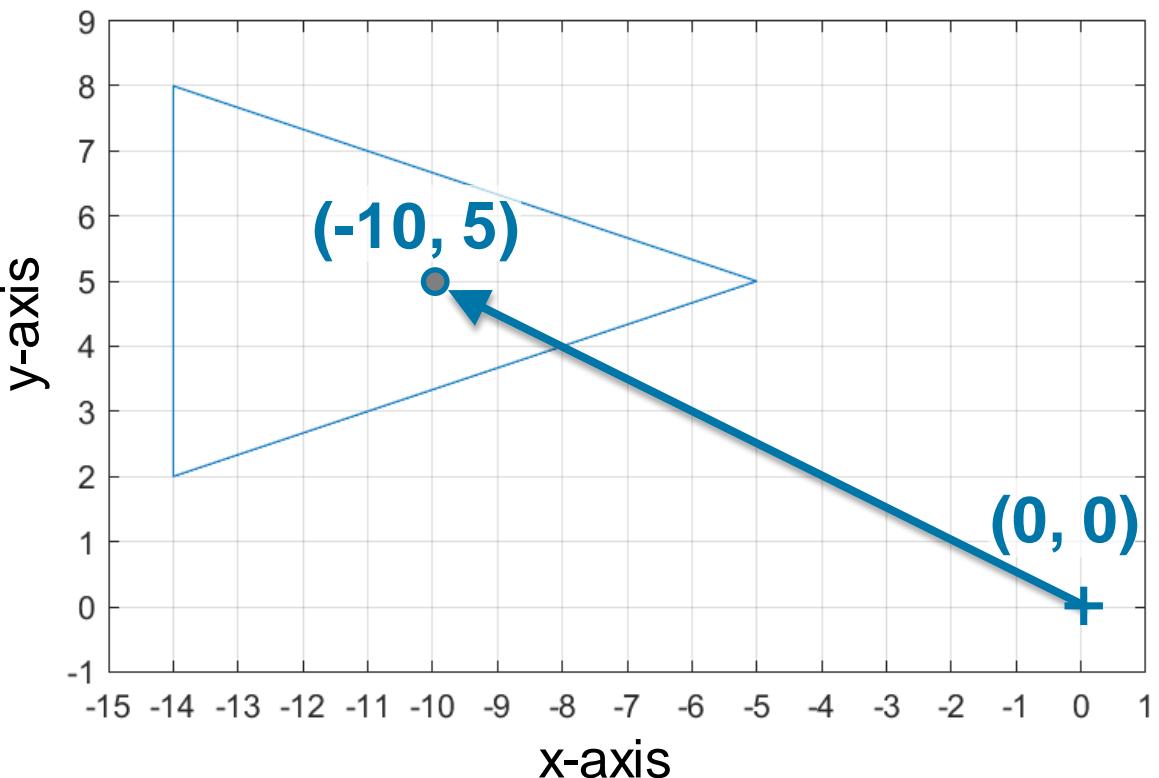
```
>> ax = gca; % get current axes
```

```
>> plot(ax, Rs*xt,Rs*yt) % < left
```

```
>> plot(ax, -Rs*xt,Rs*yt) % > right
```



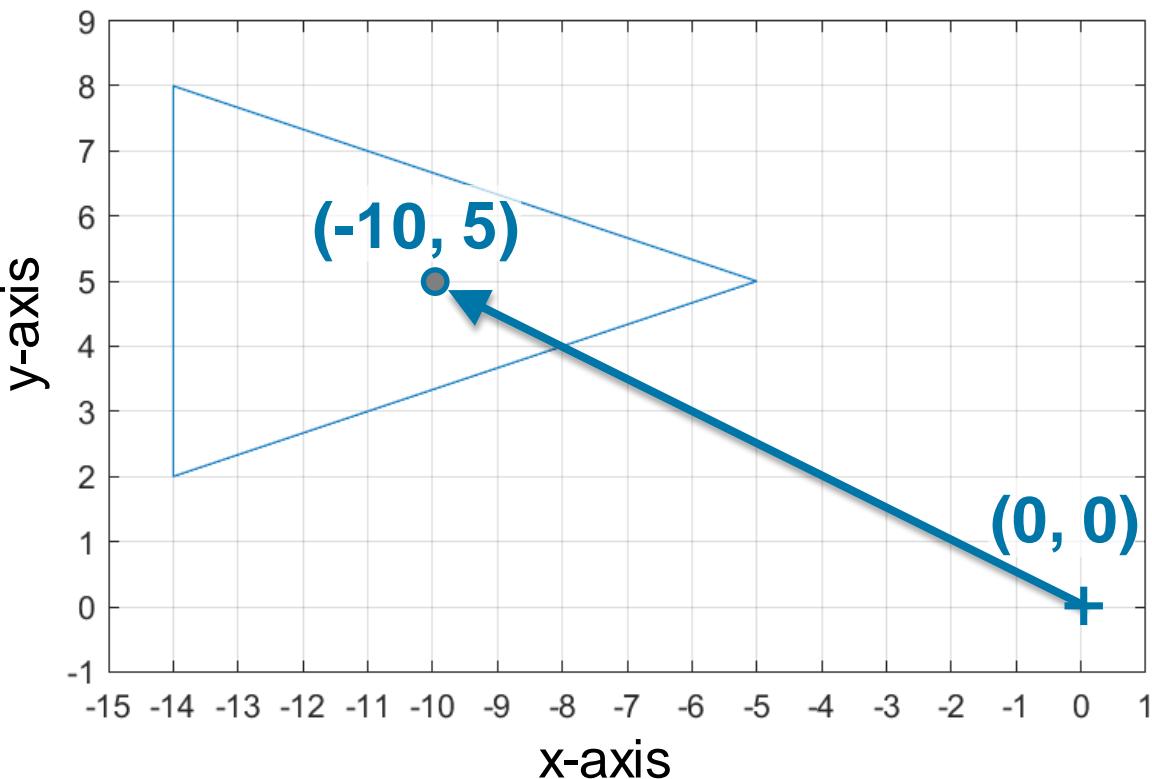
# Plotting Spaceships (Triangles)



How might we reposition  
the spaceship?

```
>> Rs = 5; % ship size  
  
>> xt = [-5 4 4 -5]/5; % tri x-pts  
  
>> yt = [ 0 3 -3 0]/5; % tri y-pts  
  
>> xs = ? % ship x-position  
  
>> ys = ? % ship y-position
```

# Plotting Spaceships (Triangles)

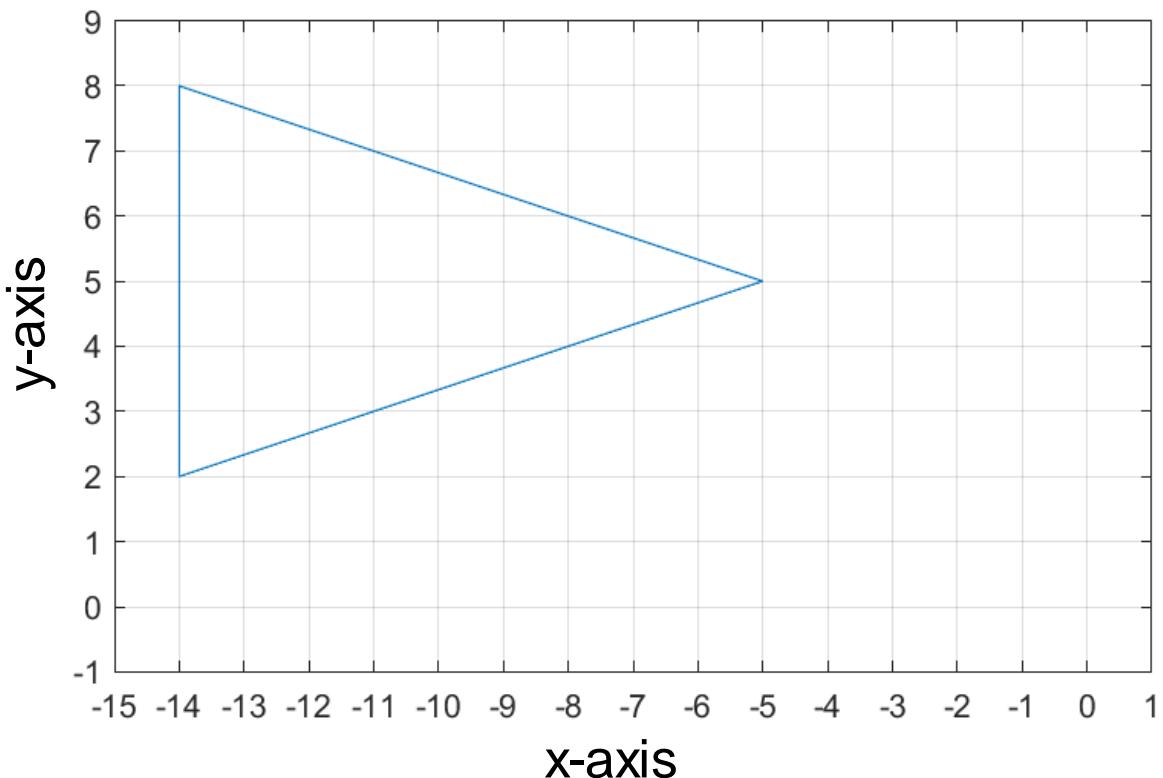


How might we reposition  
the spaceship?

```
>> Rs = 5; % ship size  
  
>> xt = [-5 4 4 -5]/5; % tri x-pts  
  
>> yt = [ 0 3 -3 0]/5; % tri y-pts  
  
>> xs = -10; % ship x-position  
  
>> ys = 5; % ship y-position  
  
>> plot(ax, xs-Rs*xt, ys+Rs*yt)
```



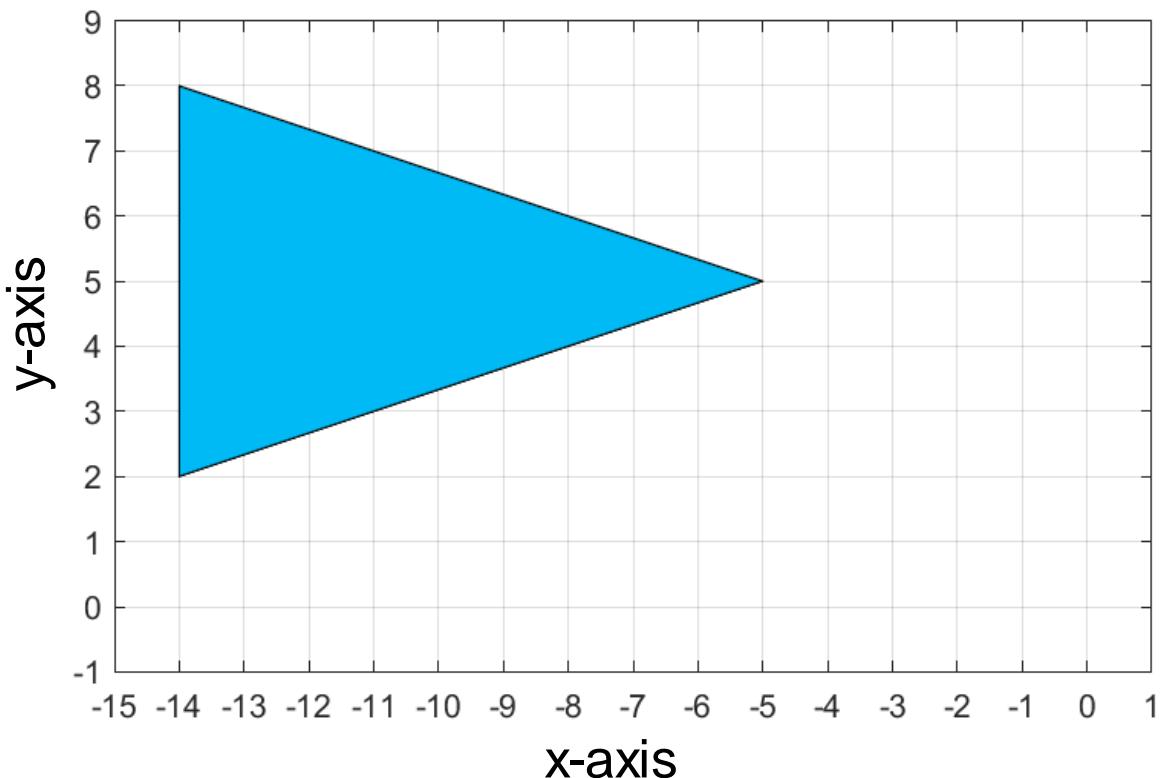
# Plotting Spaceships (Triangles)



How might we *fill* the spaceship with color?

```
>> Rs = 5; % ship size  
  
>> xt = [-5 4 4 -5]/5; % tri x-pts  
  
>> yt = [ 0 3 -3 0]/5; % tri y-pts  
  
>> xs = -10; % ship x-position  
  
>> ys = 5; % ship y-position  
  
>> plot(ax,xs-Rs*xt,ys+Rs*yt)
```

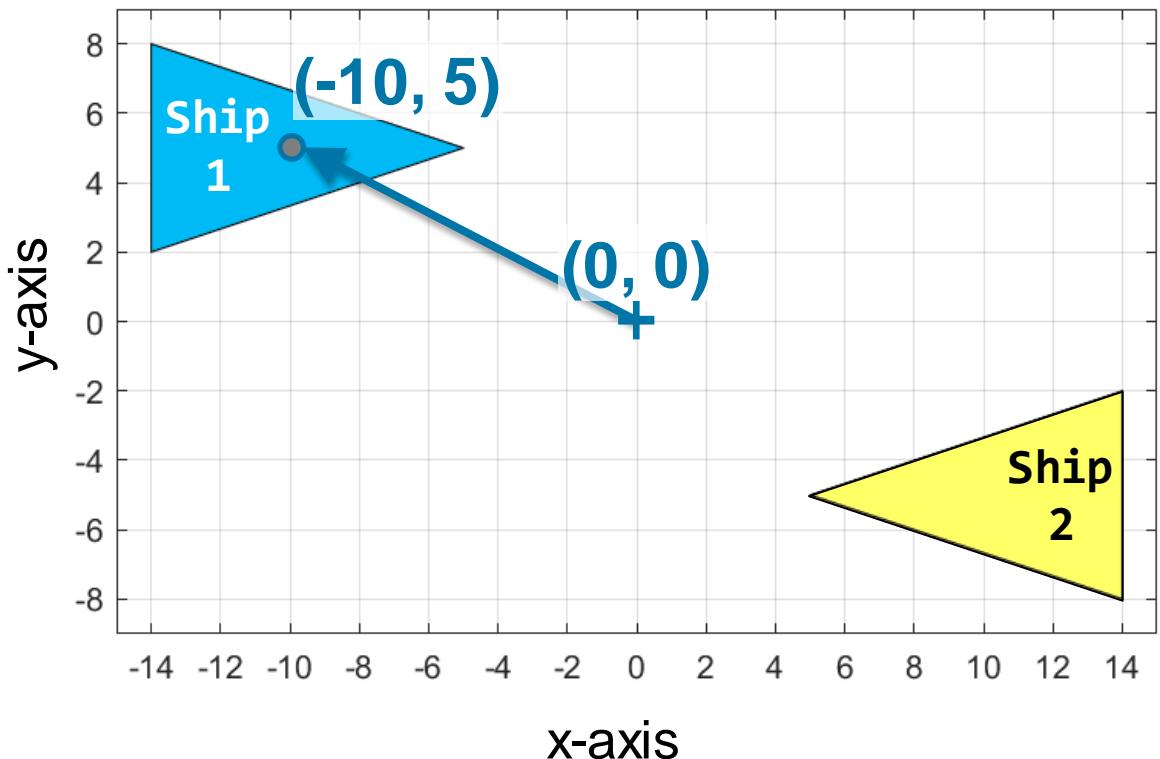
# Plotting Spaceships (Triangles)



How might we *fill* the spaceship with color?

```
>> Rs = 5; % ship size  
  
>> xt = [-5 4 4 -5]/5; % tri x-pts  
  
>> yt = [ 0 3 -3 0]/5; % tri y-pts  
  
>> xs = -10; % ship x-position  
  
>> ys = 5; % ship y-position  
  
>> fill(ax,xs-Rs*xt,ys+Rs*yt, ...  
[0 .73 .96]) % RGB Color
```

# Plotting Spaceships (Triangles)

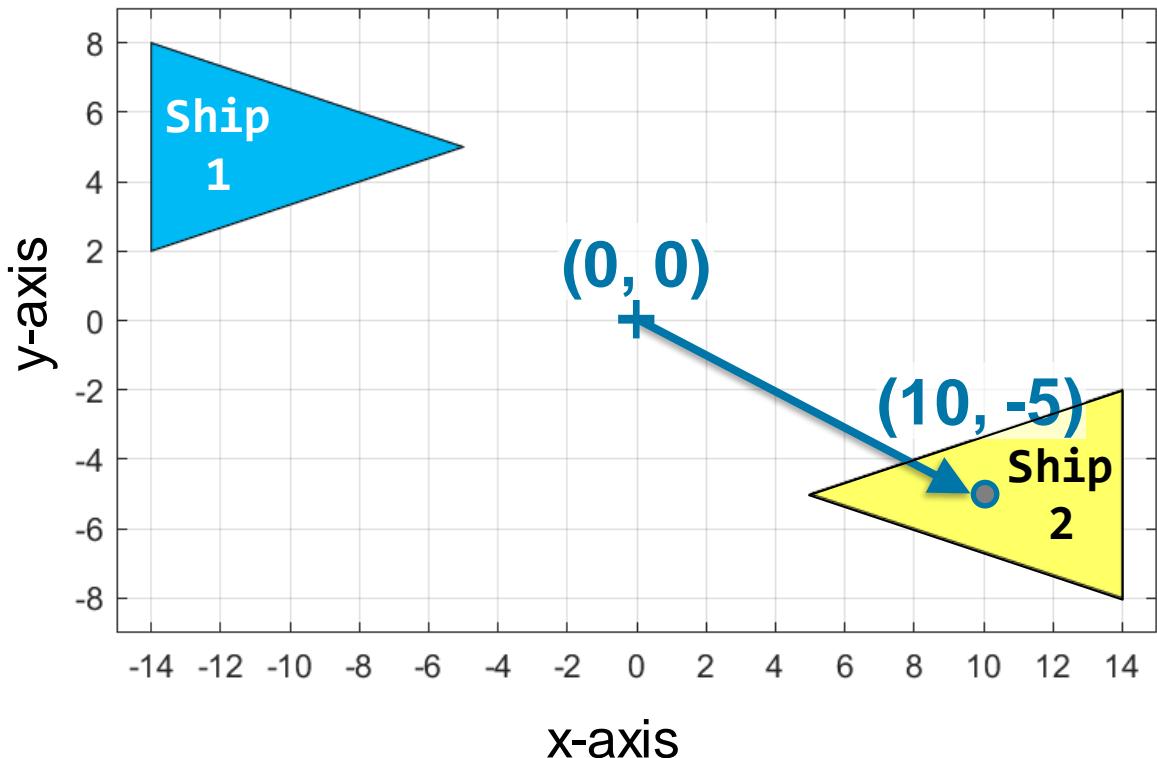


Ship 1

```
>> xs = [-10]; % ship x-pos  
>> ys = [ 5]; % ship y-pos  
>> Rs = [ 5]; % ship size  
  
>> fill(ax,xs(1)-Rs(1)*xt, ...  
      ys(1)+Rs(1)*yt, ...  
      [0 .73 .96]) % RGB Color
```

How might we add  
a second spaceship?

# Plotting Spaceships (Triangles)



Ship 1      Ship 2

```
>> xs = [-10 10]; % ship x-pos
```

```
>> ys = [ 5 -5]; % ship y-pos
```

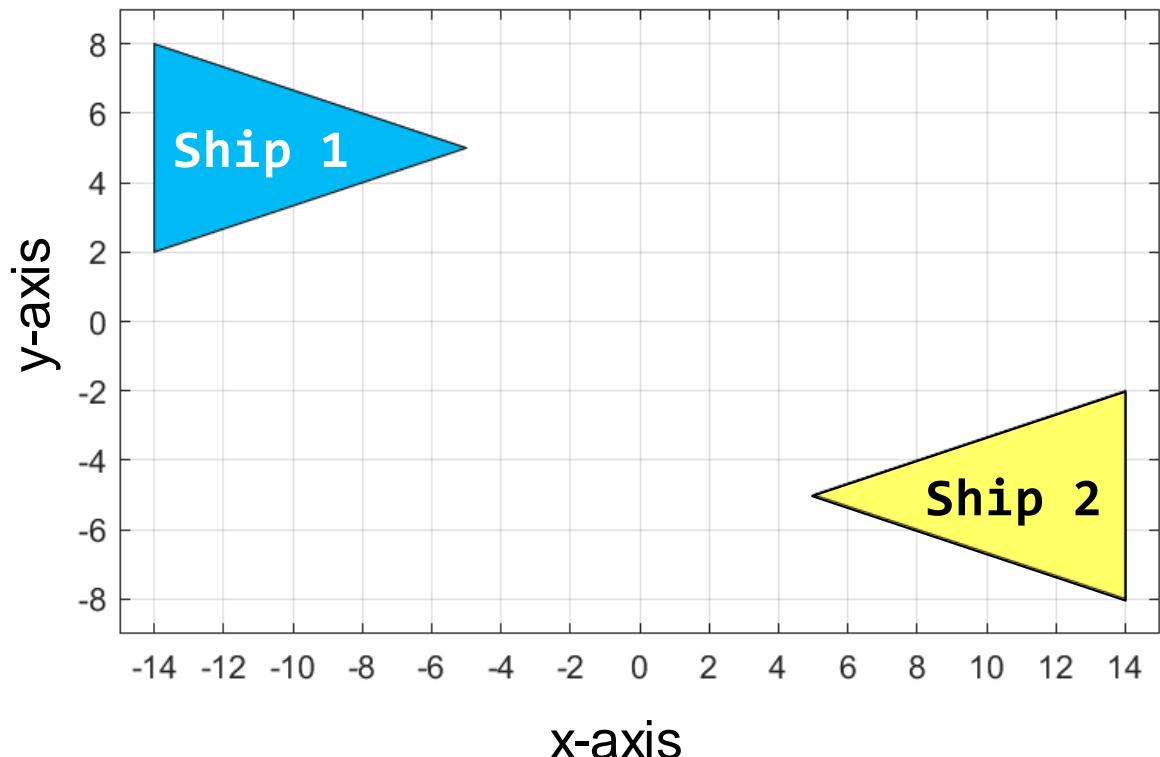
```
>> Rs = [ 5 5]; % ship size
```

```
>> fill(ax,xs(1)-Rs(1)*xt, ...  
ys(1)+Rs(1)*yt, ...
```

```
[0 .73 .96]) % RGB Color
```

How might we add  
a second spaceship?

# Plotting Spaceships (Triangles)



How might we add  
a second spaceship?

Ship 1      Ship 2

>> xs = [-10 10]; % ship x-pos

>> ys = [ 5 -5]; % ship y-pos

>> Rs = [ 5 5]; % ship size

>> fill(ax, xs(1)-Rs(1)\*xt, ...  
ys(1)+Rs(1)\*yt, ...

[0 .73 .96]) % RGB Color

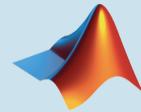
>> hold(ax, "on")

>> fill(ax, xs(2)+Rs(2)\*xt, ...  
ys(2)+Rs(2)\*yt, ...

[1 1 .4]) % RGB Color

# Course Lesson Plan

- Installing and Playing AstroVolley
- Plotting Ships (Triangles)



## Plotting Grav. Bodies (Circles)

- Gravity and Orbits
- Launching Volleys
  - Creating a Mission Map Function
  - Plotting Velocity Vectors (quiver)
  - Animating Trajectories (comet)

## App Designer

- 1) UIAxes, Properties, & Functions
- 2) Sliders & Value Changing Callback
- 3) Push Button Callback
- 4) Generating Random Maps
- 5) Keeping Score

# Represent game pieces using basic shapes...

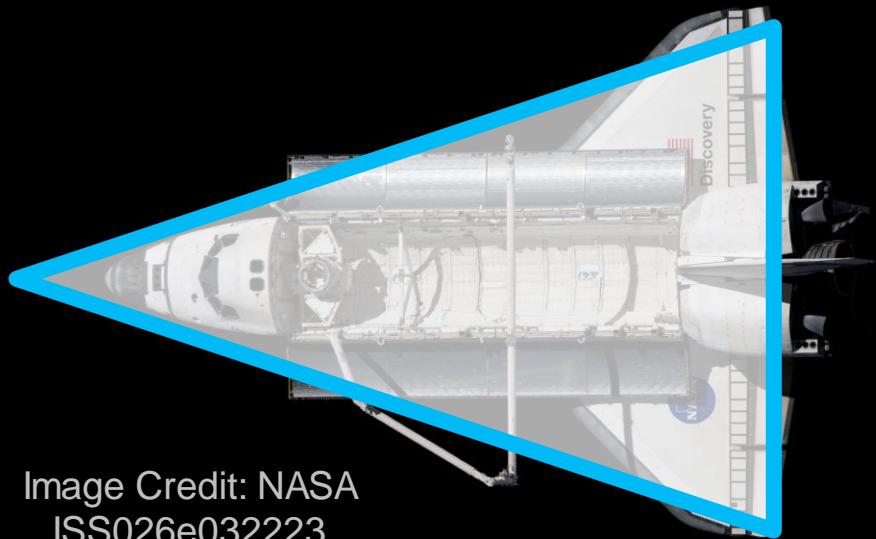
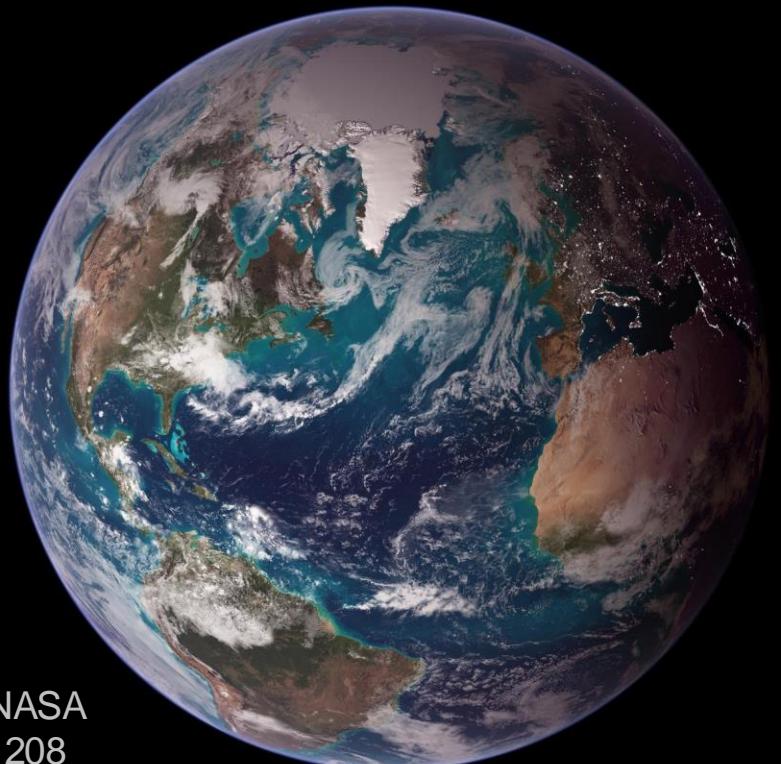


Image Credit: NASA  
ISS026e032223

Triangles for the  
Spaceships  
(Space Shuttle)

# Represent game pieces using basic shapes...

Circles for the  
Gravitational Bodies  
(Planets)

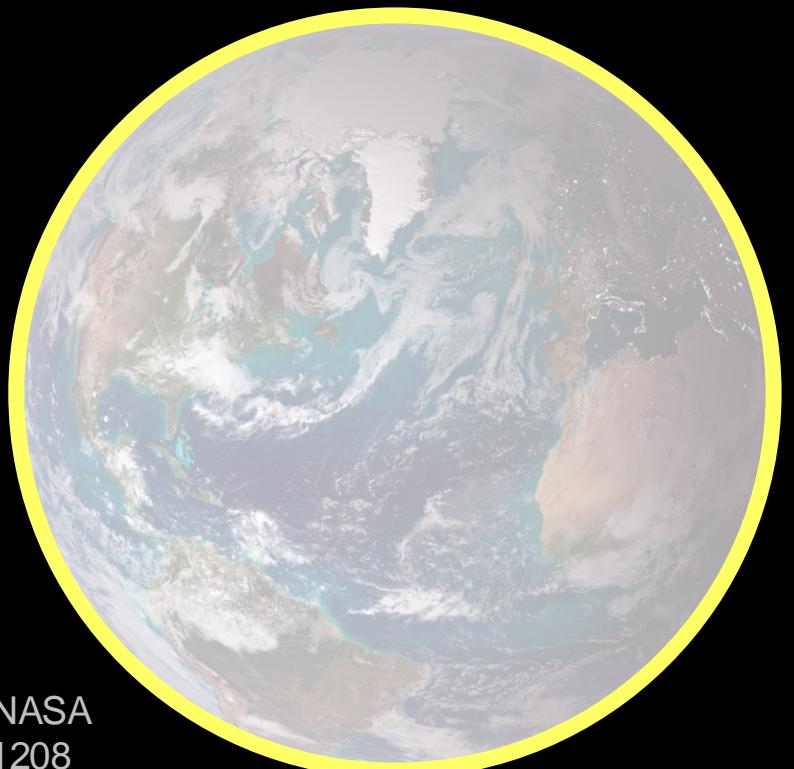


Image Credit: NASA  
GSFC\_20171208  
Archive\_e002131

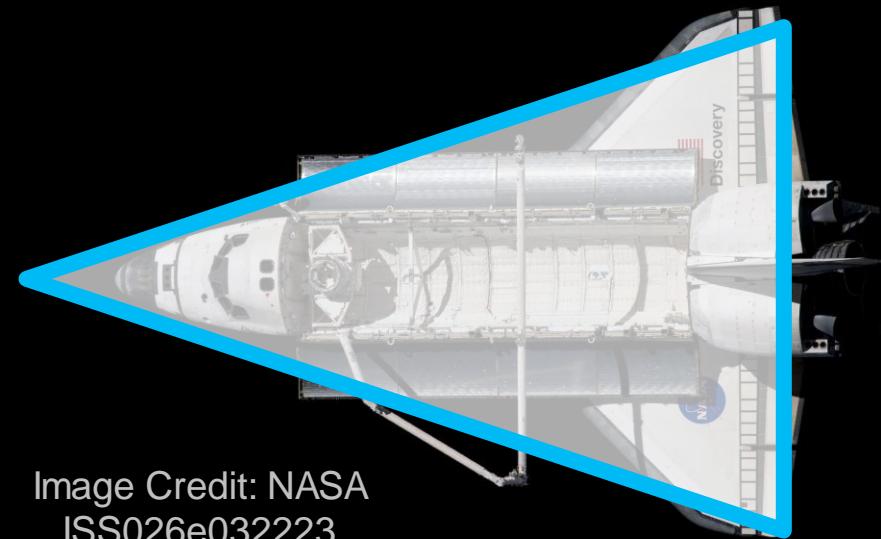
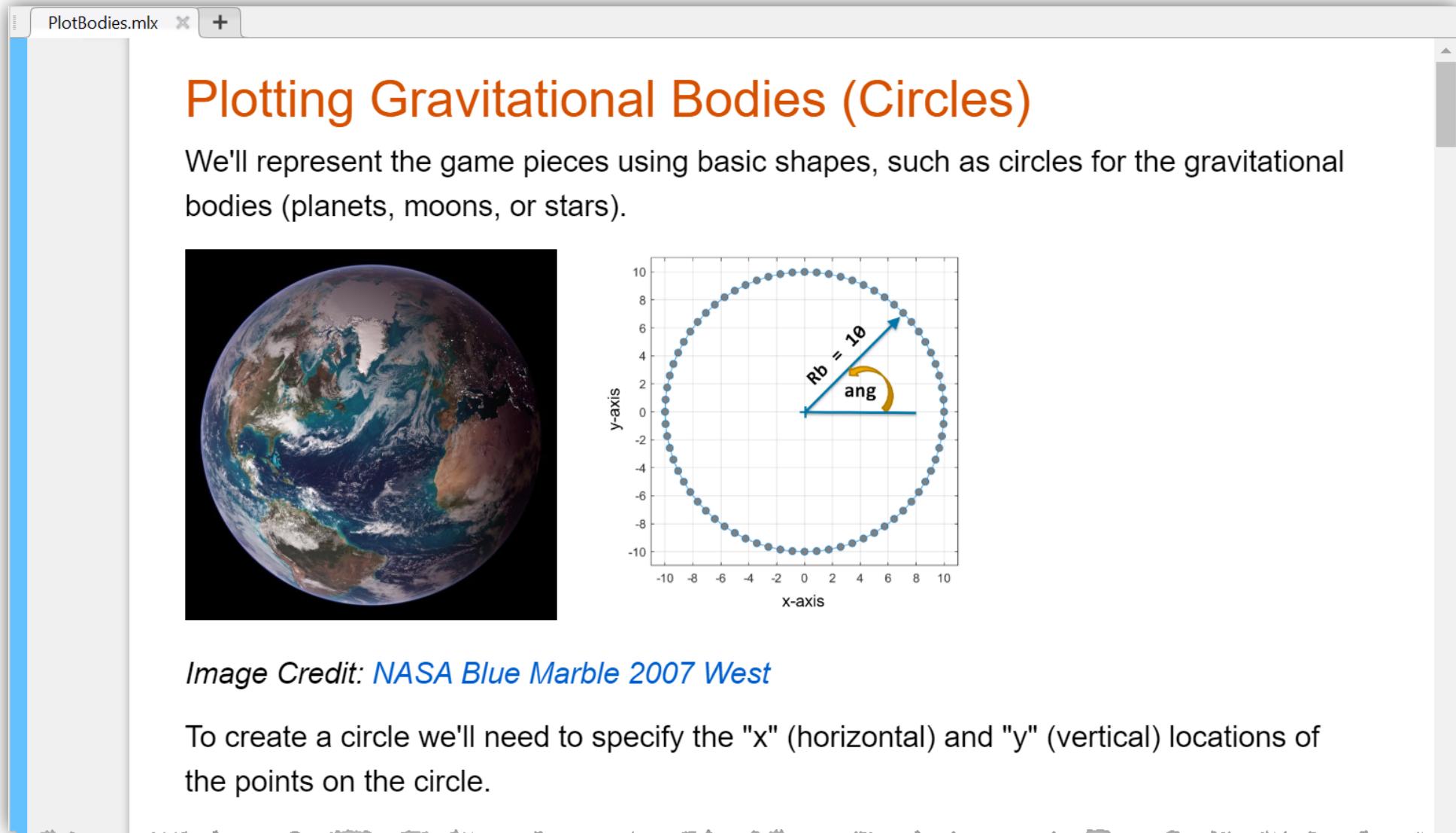


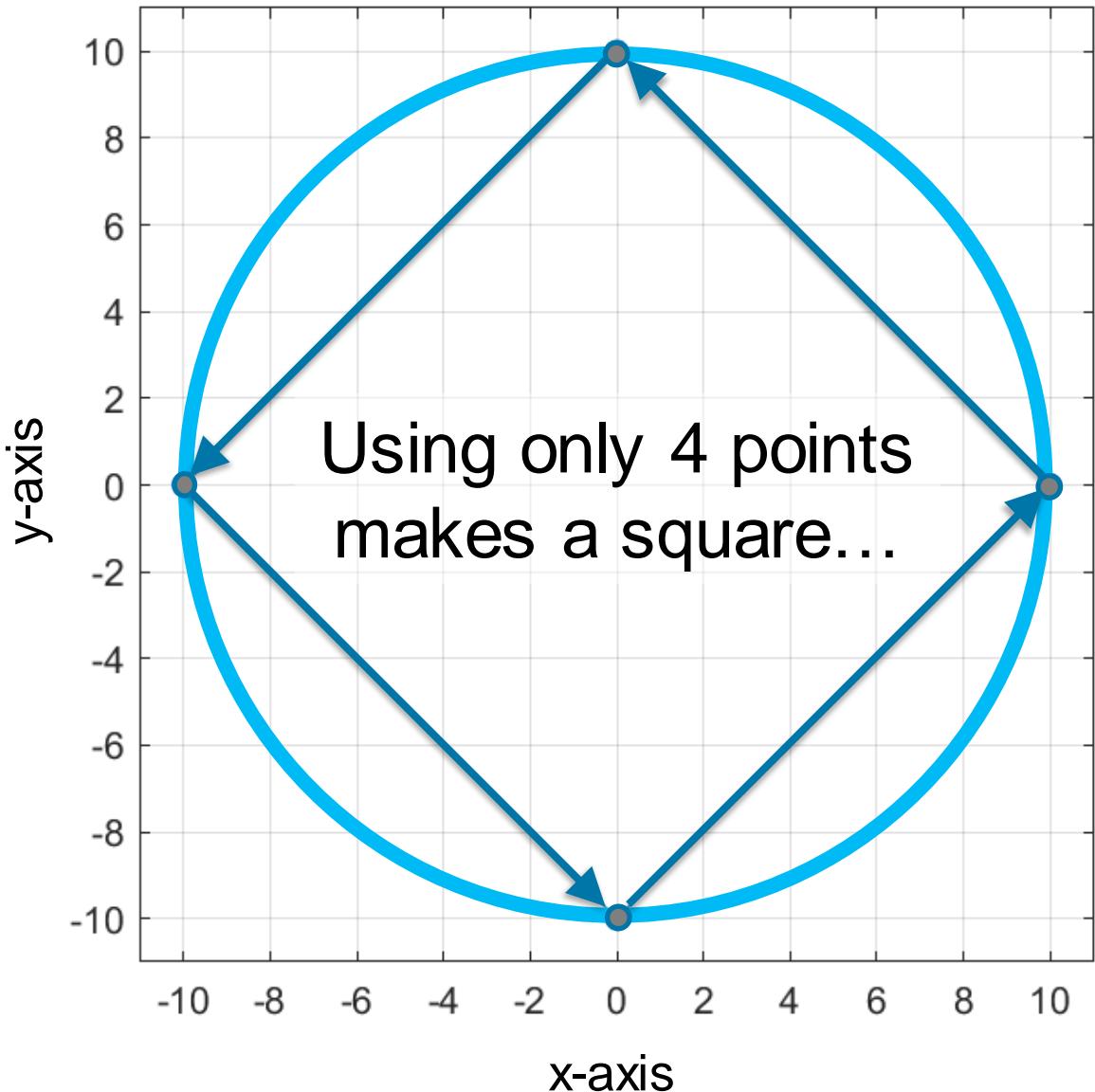
Image Credit: NASA  
ISS026e032223

Triangles for the  
Spaceships  
(Space Shuttle)

```
>> open PlotBodies mlx
```

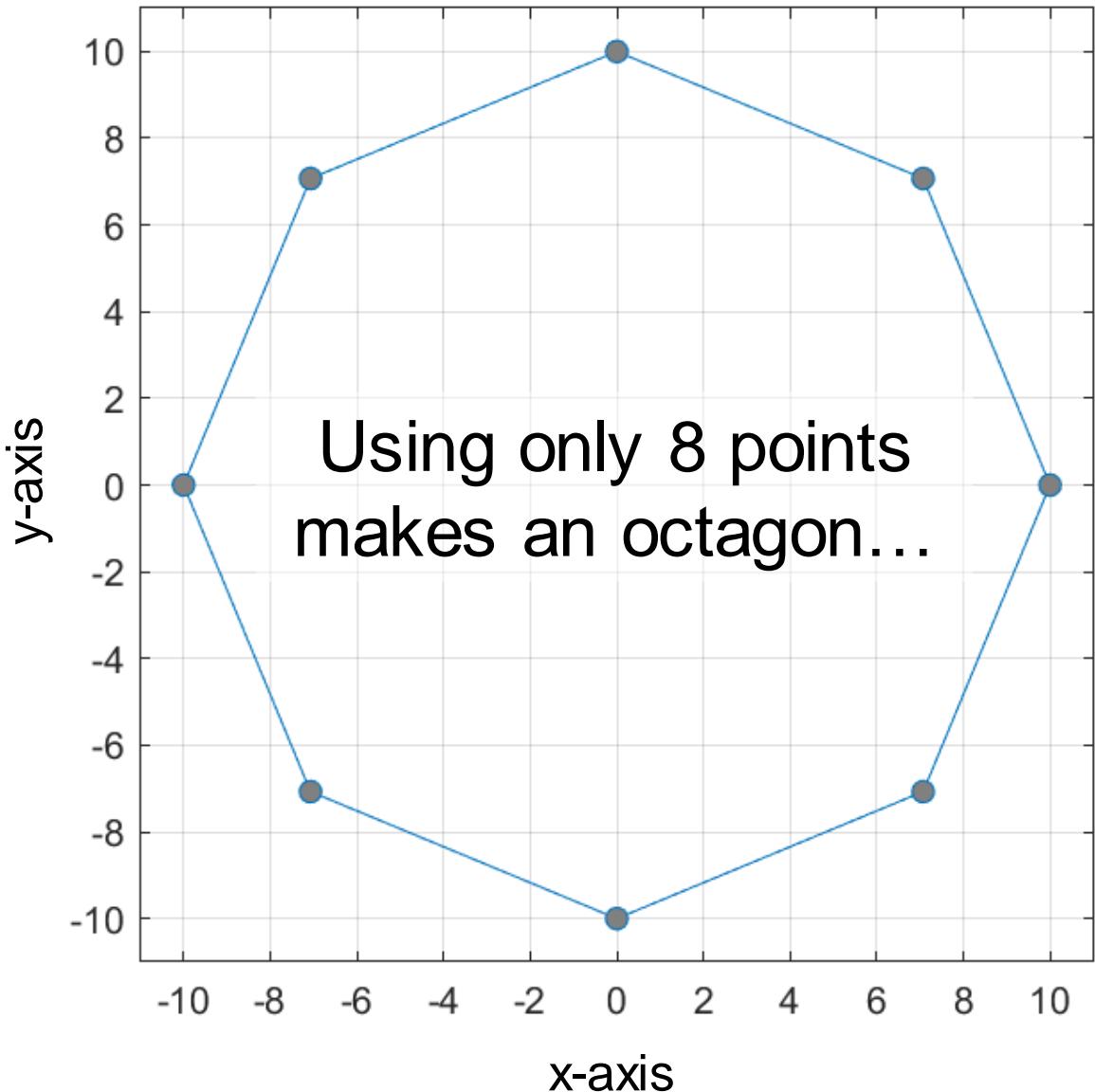


## Plotting Grav. Bodies (Circles)



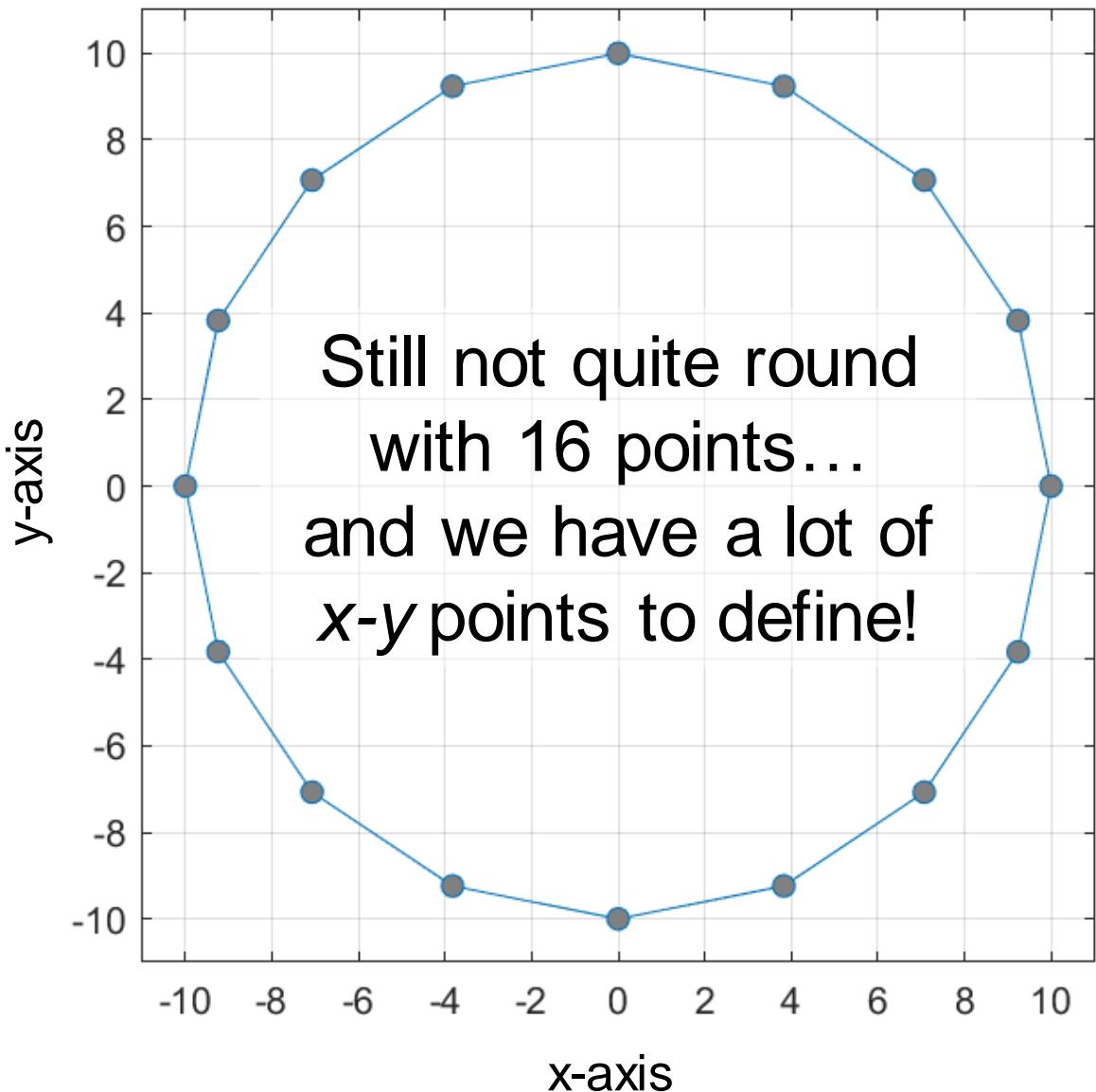
```
>> ax = gca; % get current axes  
  
>> xc = [10 0 -10 0 10];  
  
>> yc = [0 10 0 -10 0];  
  
>> plot(ax,xc,yc)
```

## Plotting Grav. Bodies (Circles)



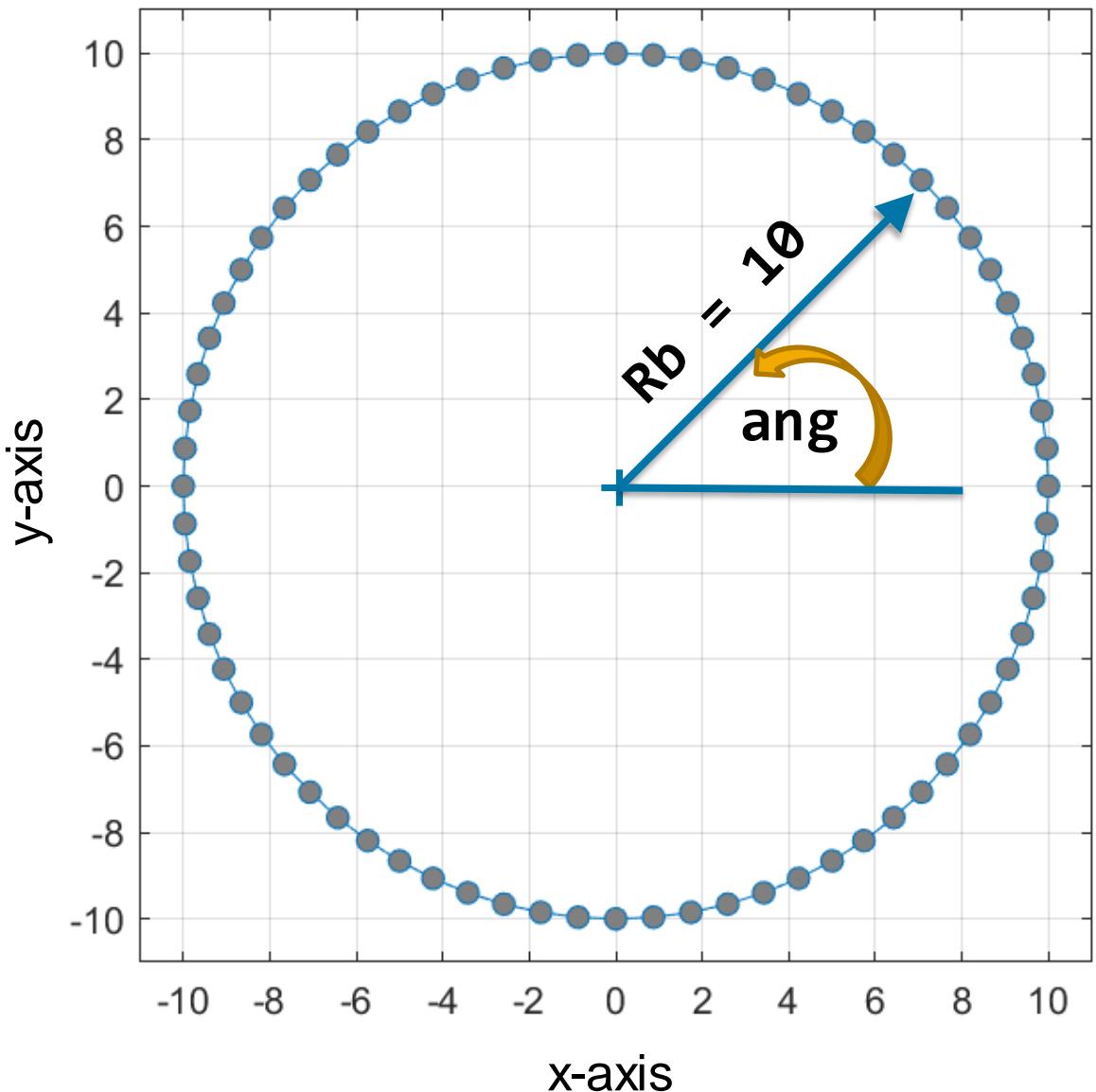
```
>> ax = gca; % get current axes  
  
>> xc = [10 7.07 0 -7.07 ...  
         -10 -7.07 0 -7.07 10];  
  
>> yc = [0 7.07 10 7.07 ...  
         0 -7.07 -10 -7.07 0];  
  
>> plot(ax,xc,yc)
```

## Plotting Grav. Bodies (Circles)



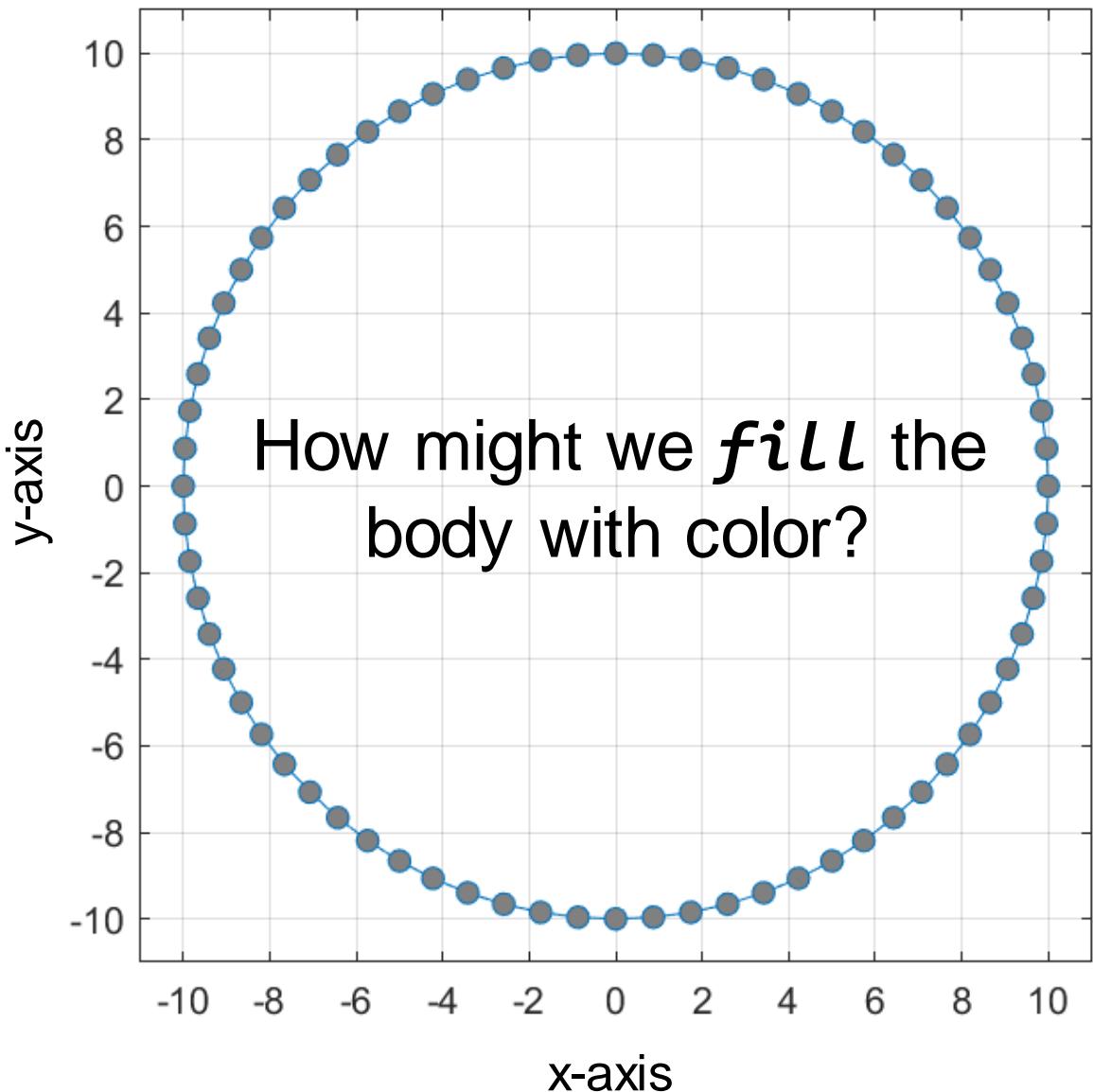
```
>> ax = gca; % get current axes  
  
>> xc = [10 9.24 7.07 3.83 ...  
          0 -3.83 -7.07 -9.24 ...  
         -10 -9.24 -7.07 -3.83 ...  
          0 3.83 7.07 9.24 10];  
  
>> yc = [ 0 3.83 7.07 9.24 ...  
          10 9.24 7.07 3.83 ...  
          0 -3.83 -7.07 -9.24 ...  
         -10 -9.24 -7.07 -3.83 0];  
  
>> plot(ax,xc,yc)
```

# Plotting Grav. Bodies (Circles)



```
>> Rb = 10; % body size  
>> ang = 0:5:360; % degrees  
>> xc = cosd(ang); % circle x-pts  
>> yc = sind(ang); % circle y-pts  
>> plot(ax,Rb*xc,Rb*yc)  
>> axis(ax,"equal")
```

## Plotting Grav. Bodies (Circles)



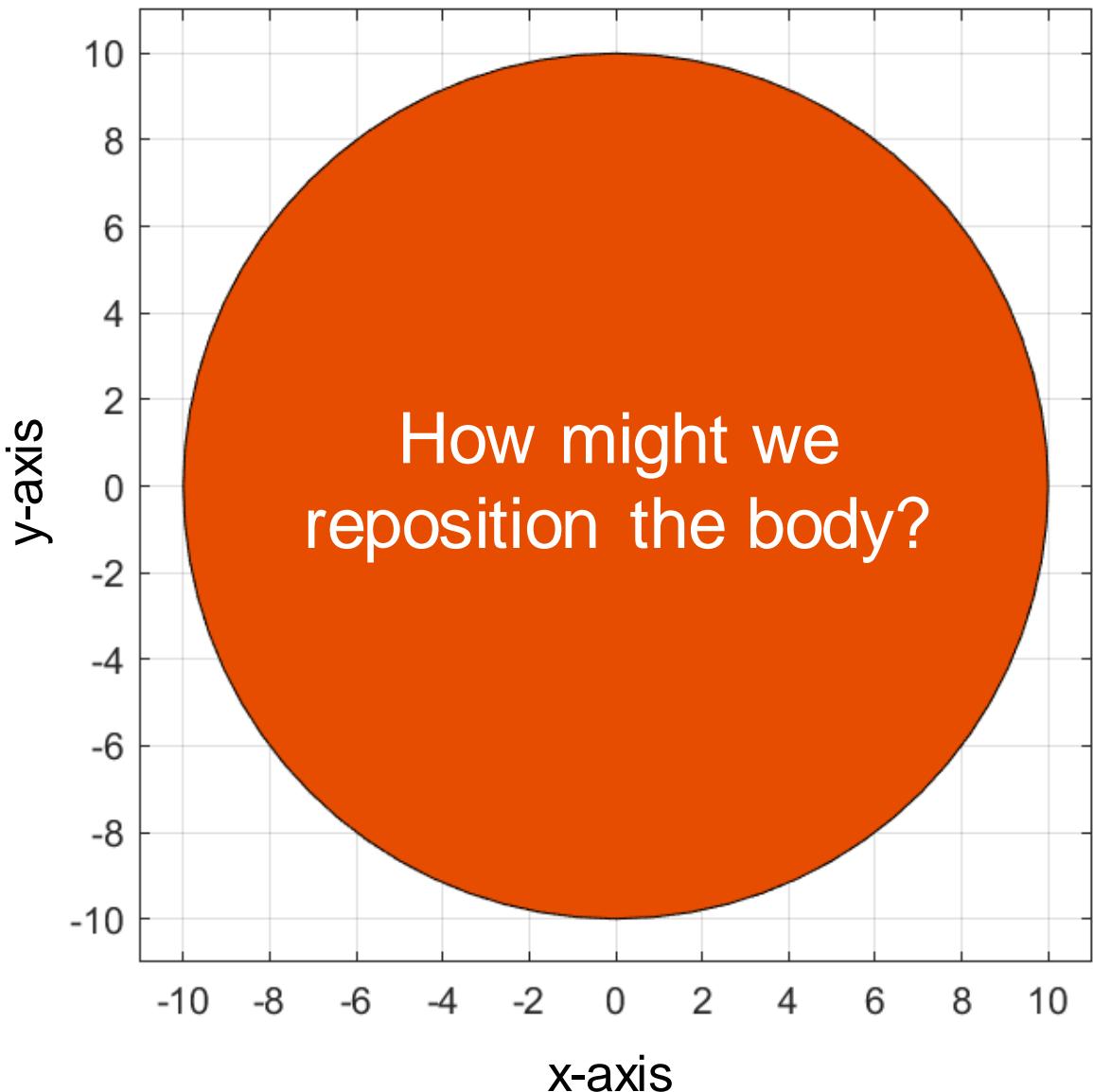
```
>> Rb = 10; % body size  
>> ang = 0:5:360; % degrees  
>> xc = cosd(ang); % circle x-pts  
>> yc = sind(ang); % circle y-pts  
>> plot(ax,Rb*xc,Rb*yc)  
>> axis(ax,"equal")
```

## Plotting Grav. Bodies (Circles)



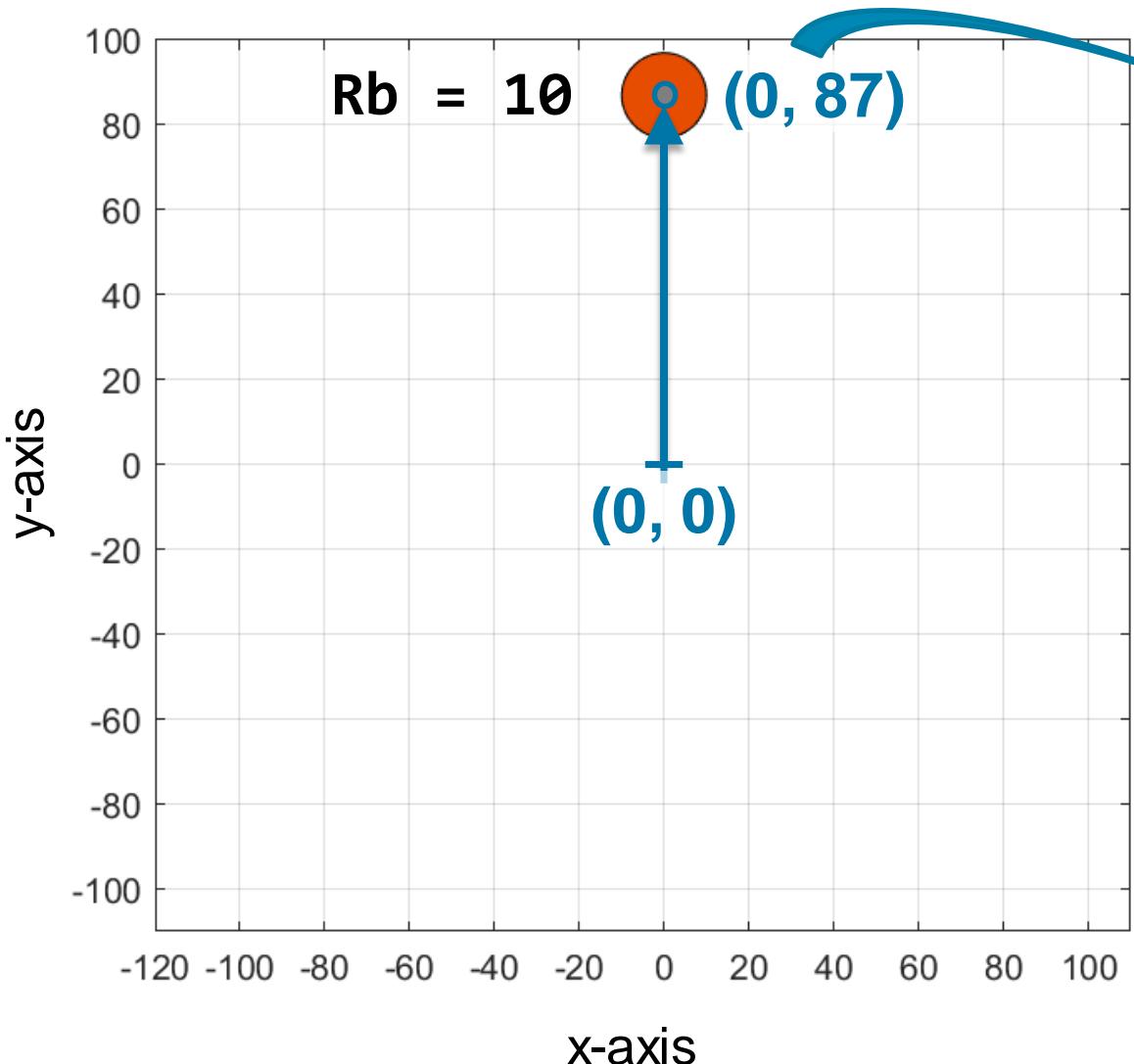
```
>> Rb = 10; % body size  
  
>> ang = 0:5:360; % degrees  
  
>> xc = cosd(ang); % circle x-pts  
  
>> yc = sind(ang); % circle y-pts  
  
>> fill(ax,Rb*xc,Rb*yc, ...  
      [.9 .3 0]) % RGB Color  
  
>> axis(ax,"equal")
```

## Plotting Grav. Bodies (Circles)



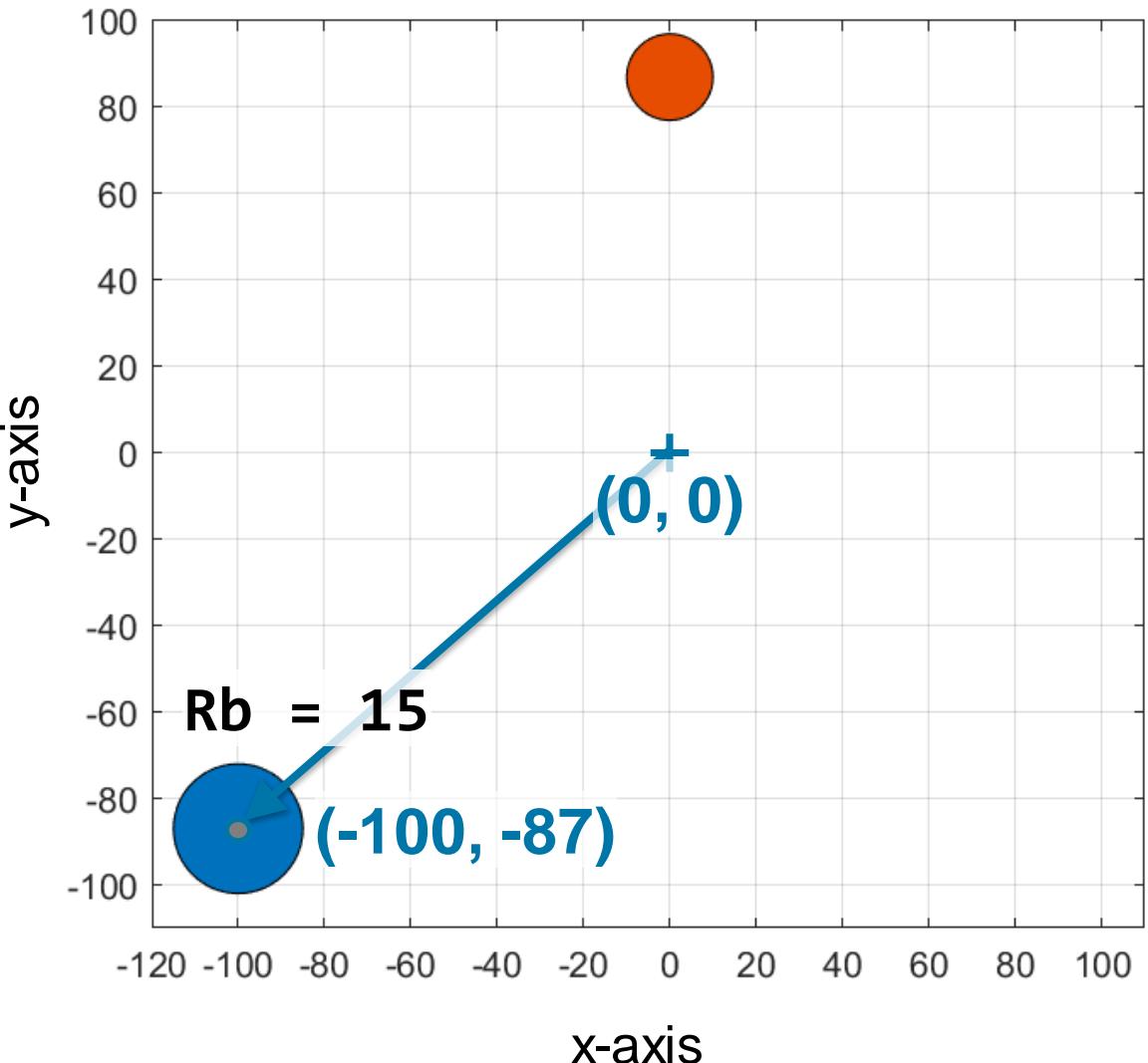
```
>> Rb = 10; % body size  
  
>> ang = 0:5:360; % degrees  
  
>> xc = cosd(ang); % circle x-pts  
  
>> yc = sind(ang); % circle y-pts  
  
>> fill(ax,Rb*xc,Rb*yc, ...  
      [.9 .3 0]) % RGB Color  
  
>> axis(ax,"equal")
```

# Plotting Gravitational Bodies (Circles)



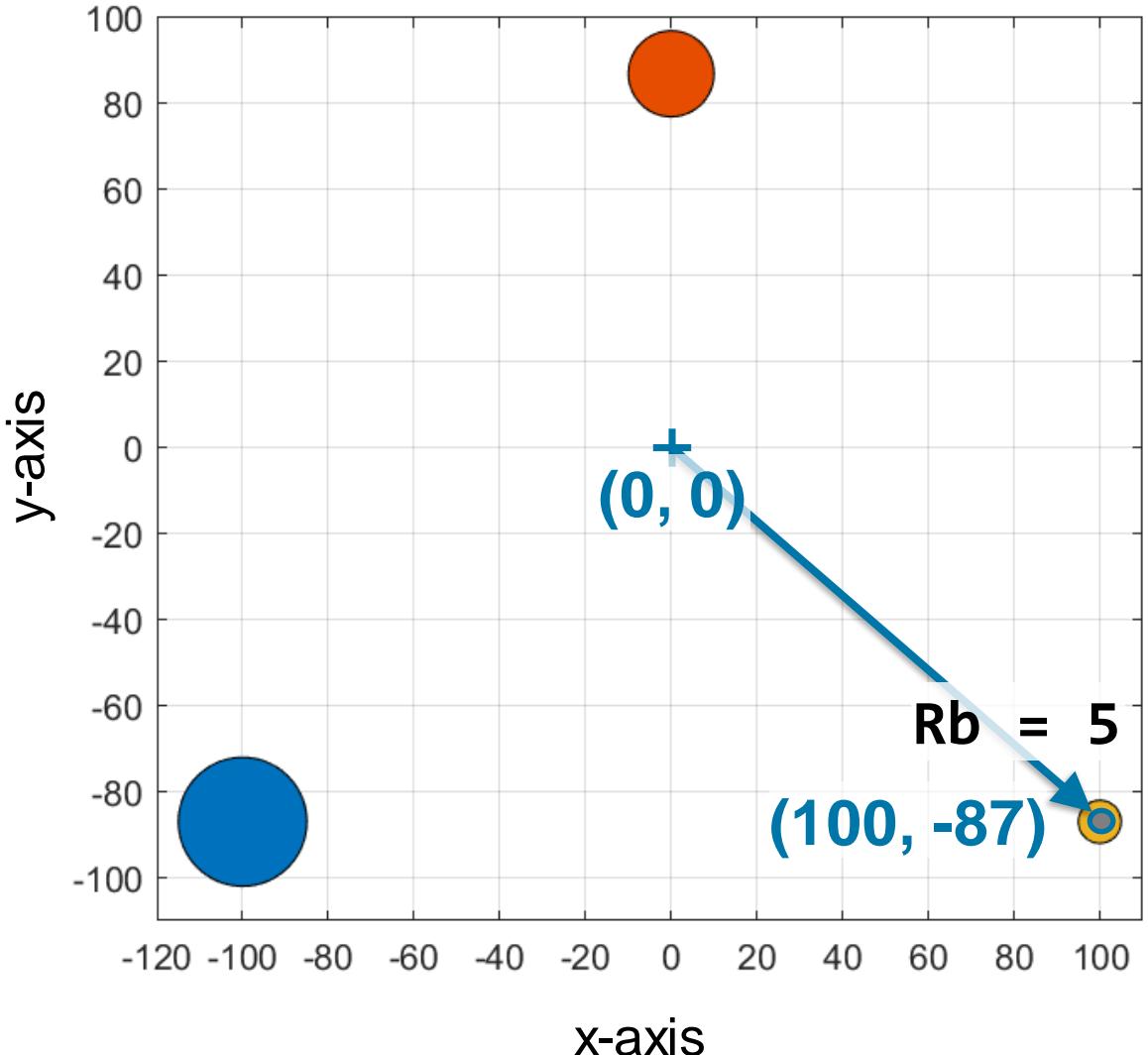
```
>> fill(ax, 0+10*xc, ...
          87+10*yc, ...
          [.9 .3 0]) % RGB Color
```

# Plotting Gravitational Bodies (Circles)



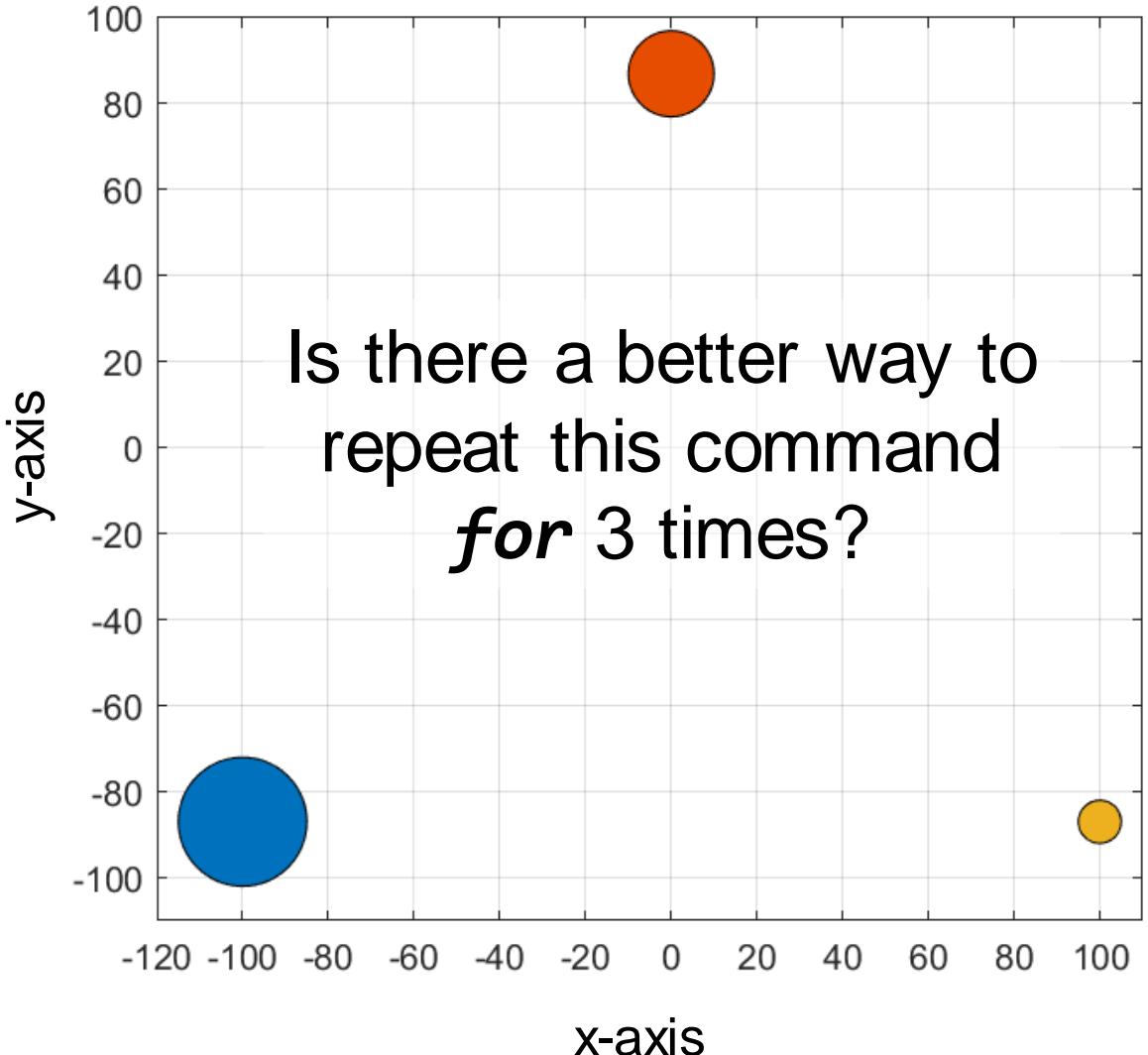
```
>> fill(ax, -100+15*xc, ...  
      -87+15*yc, ...  
      [0 .4 .7]) % RGB Color  
  
>> fill(ax, 0+10*xc, ...  
      87+10*yc, ...  
      [.9 .3 0]) % RGB Color
```

# Plotting Gravitational Bodies (Circles)



```
>> fill(ax, -100+15*xc, ...  
      -87+15*yc, ...  
      [0 .4 .7]) % RGB Color  
  
>> fill(ax, 0+10*xc, ...  
      87+10*yc, ...  
      [.9 .3 0]) % RGB Color  
  
>> fill(ax, 100+5*xc, ...  
      -87+5*yc, ...  
      [.9 .7 .1]) % RGB Color
```

# Plotting Gravitational Bodies (Circles)



```
>> fill(ax, -100+15*xc, ...
         -87+15*yc, ...
         [0 .4 .7]) % RGB Color
```

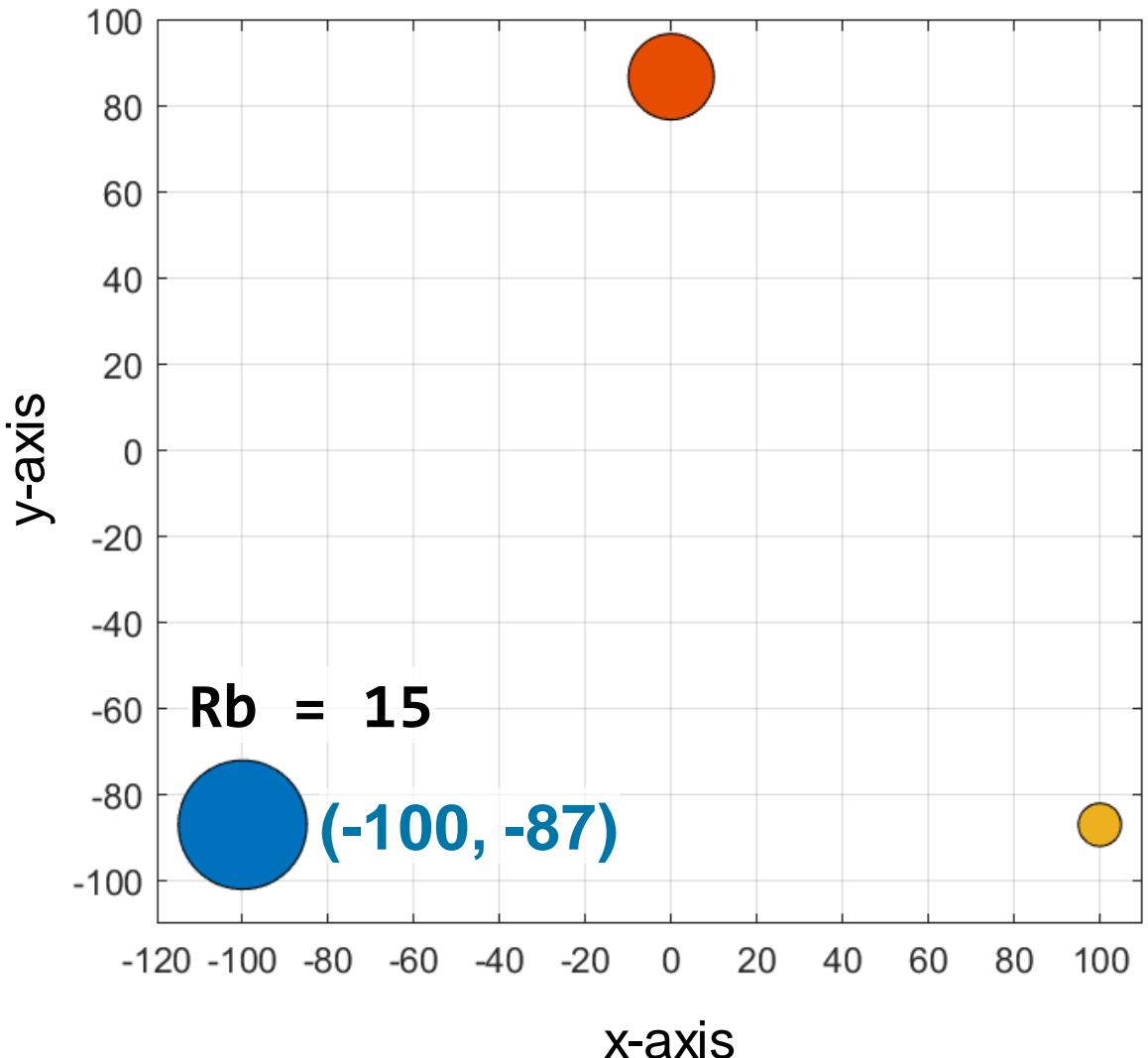
  

```
>> fill(ax, 0+10*xc, ...
         87+10*yc, ...
         [.9 .3 0]) % RGB Color
```

```
>> fill(ax, 100+5*xc, ...
         -87+5*yc, ...
         [.9 .7 .1]) % RGB Color
```

# Plotting Gravitational Bodies (Circles)



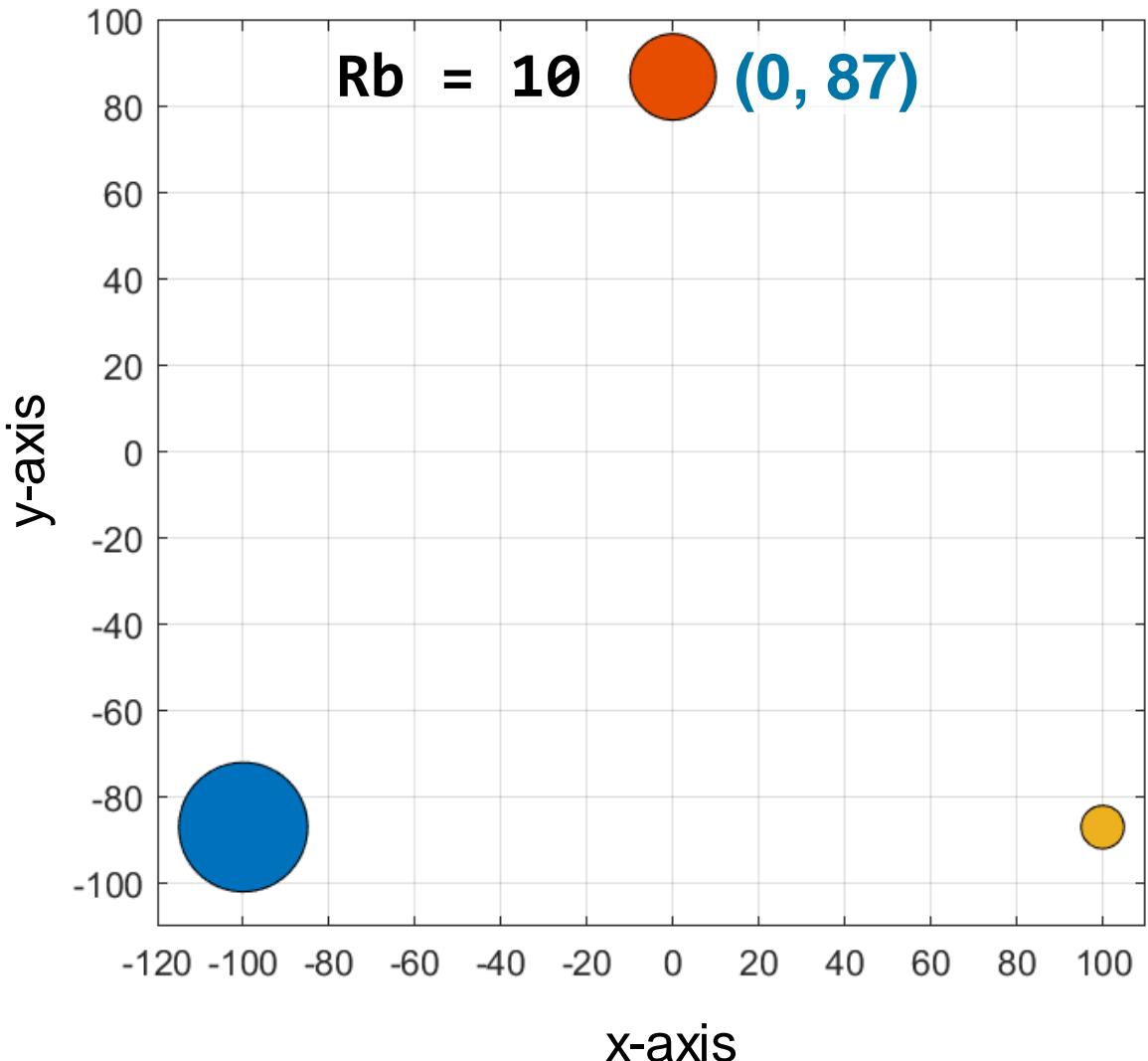
Body 1  
(k = 1)

```
>> xb = [-100 0 100]; % body x-pos
>> yb = [-87 87 -87]; % body y-pos
>> Rb = [15 10 5]; % body size
>> cb = lines(3); % body color

>> for k = 1:3
    fill(ax, xb(k)+Rb(k)*xc, ...
          yb(k)+Rb(k)*yc, ...
          cb(k,:))

end
>> axis(ax, "equal")
```

# Plotting Gravitational Bodies (Circles)



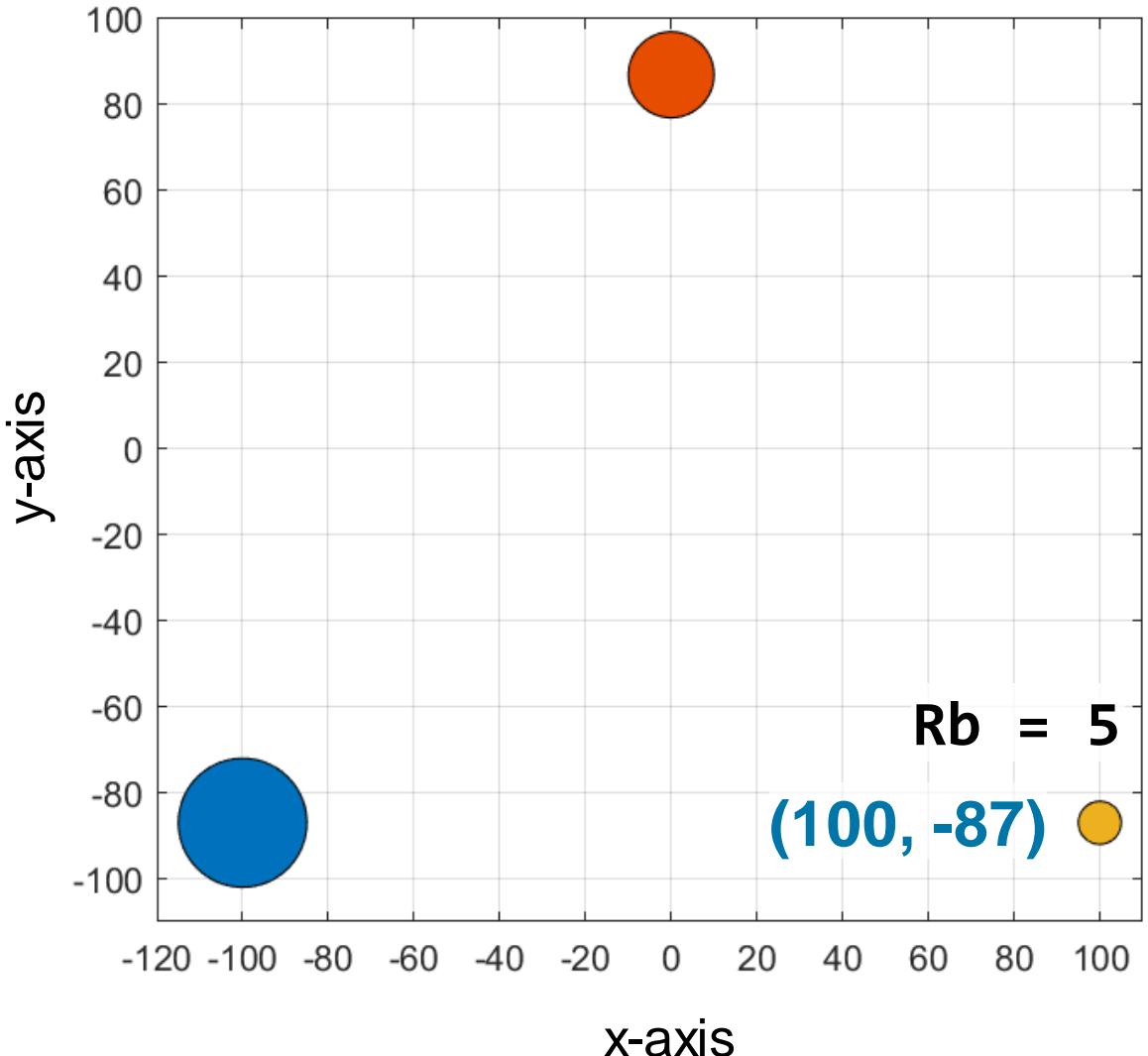
Body 2  
( $k = 2$ )

```
>> xb = [-100 0 100]; % body x-pos
>> yb = [ -87 87 -87]; % body y-pos
>> Rb = [ 15 10 5]; % body size
>> cb = lines(3); % body color

>> for k = 1:3
    fill(ax,xb(k)+Rb(k)*xc, ...
          yb(k)+Rb(k)*yc, ...
          cb(k,:))

end
>> axis(ax, "equal")
```

# Plotting Gravitational Bodies (Circles)



Body 3  
( $k = 3$ )

```
>> xb = [ -100 0 100]; % body x-pos
>> yb = [ -87 87 -87]; % body y-pos
>> Rb = [ 15 10 5]; % body size
>> cb = lines(3); % body color

>> for k = 1:3
    fill(ax,xb(k)+Rb(k)*xc, ...
          yb(k)+Rb(k)*yc, ...
          cb(k,:))

end
>> axis(ax, "equal")
```

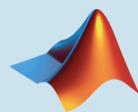
# Our code so far... (spread out the ships and save as mapMission.m)

```
ax = gca; % get current axes  
cla(ax) % clear axes  
xs = [-225 225]; % ship x-positions  
ys = [ 100 -100]; % ship y-positions  
Rs = [ 5 5]; % ship sizes  
cs = [0 .73 .96; 1 1 .4]; % ship colors  
  
xt = [-5 4 4 -5]/5; % triangle x-points  
yt = [ 0 3 -3 0]/5; % triangle y-points  
  
fill(ax,xs(1)-Rs(1)*xt, ...  
      ys(1)+Rs(1)*yt,cs(1,:))  
hold(ax,"on")  
fill(ax,xs(2)+Rs(2)*xt, ...  
      ys(2)+Rs(2)*yt,cs(2,:))
```

```
xb = [-100 0 100]; % body x-positions  
yb = [ -87 87 -87]; % body y-positions  
Rb = [ 15 10 5]; % body sizes  
cb = lines(3); % body colors  
  
ang = 0:5:360; % angle, degrees  
xc = cosd(ang); % circle x-points  
yc = sind(ang); % circle y-points  
  
for k = 1:3  
    fill(ax,xb(k)+Rb(k)*xc, ...  
          yb(k)+Rb(k)*yc,cb(k,:))  
end  
axis(ax,"equal")
```

# Course Lesson Plan

- Installing and Playing AstroVolley
- Plotting Ships (Triangles)
- Plotting Grav. Bodies (Circles)



## Gravity and Orbits

- Launching Volleys
  - Creating a Mission Map Function
  - Plotting Velocity Vectors (quiver)
  - Animating Trajectories (comet)

## App Designer

- 1) UIAxes, Properties, & Functions
- 2) Sliders & Value Changing Callback
- 3) Push Button Callback
- 4) Generating Random Maps
- 5) Keeping Score

# The International Space Station (ISS)

It's big, it's 400 km up, and it orbits the Earth every 90 minutes!

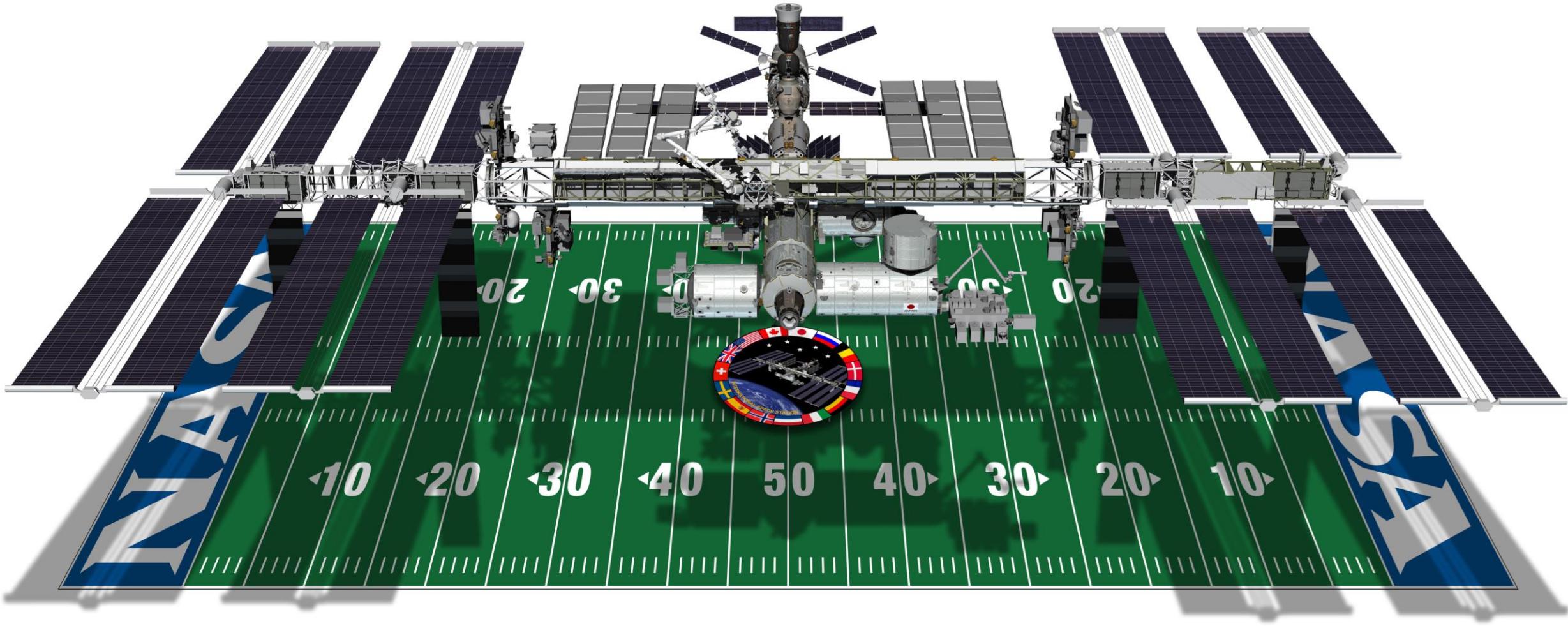


Image Credit: NASA

# Spot the International Space Station from Your Town!

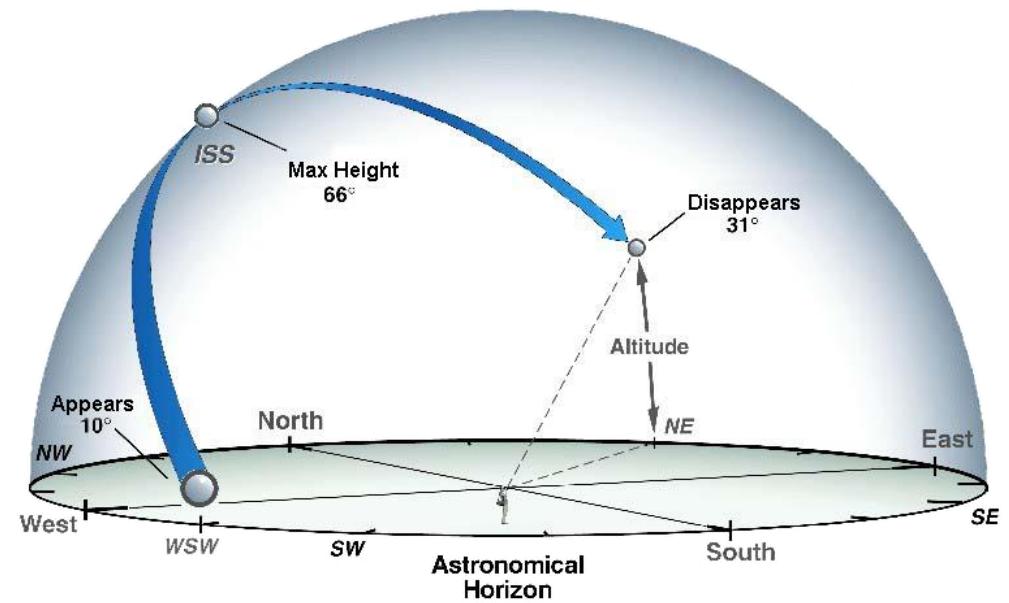
<https://spotthestation.nasa.gov>



The International Space Station is seen in this 30 second exposure.  
Photo Credit: NASA/Bill Ingalls

## What does all this sighting information mean?

Date/Time: Sun Mar 20, 8:33 PM  
Visible: 3 min  
Max Height:  $43^\circ$   
Appears:  $10^\circ$  above SW  
Disappears:  $43^\circ$  above SSE



# Astronauts Perform 'Synchronized Space Swimming' on ISS

Why are they floating in the video? What happened to gravity?



Image Credit: NASA  
ISS065e166107

What will happen when this object is released with no velocity? Why?



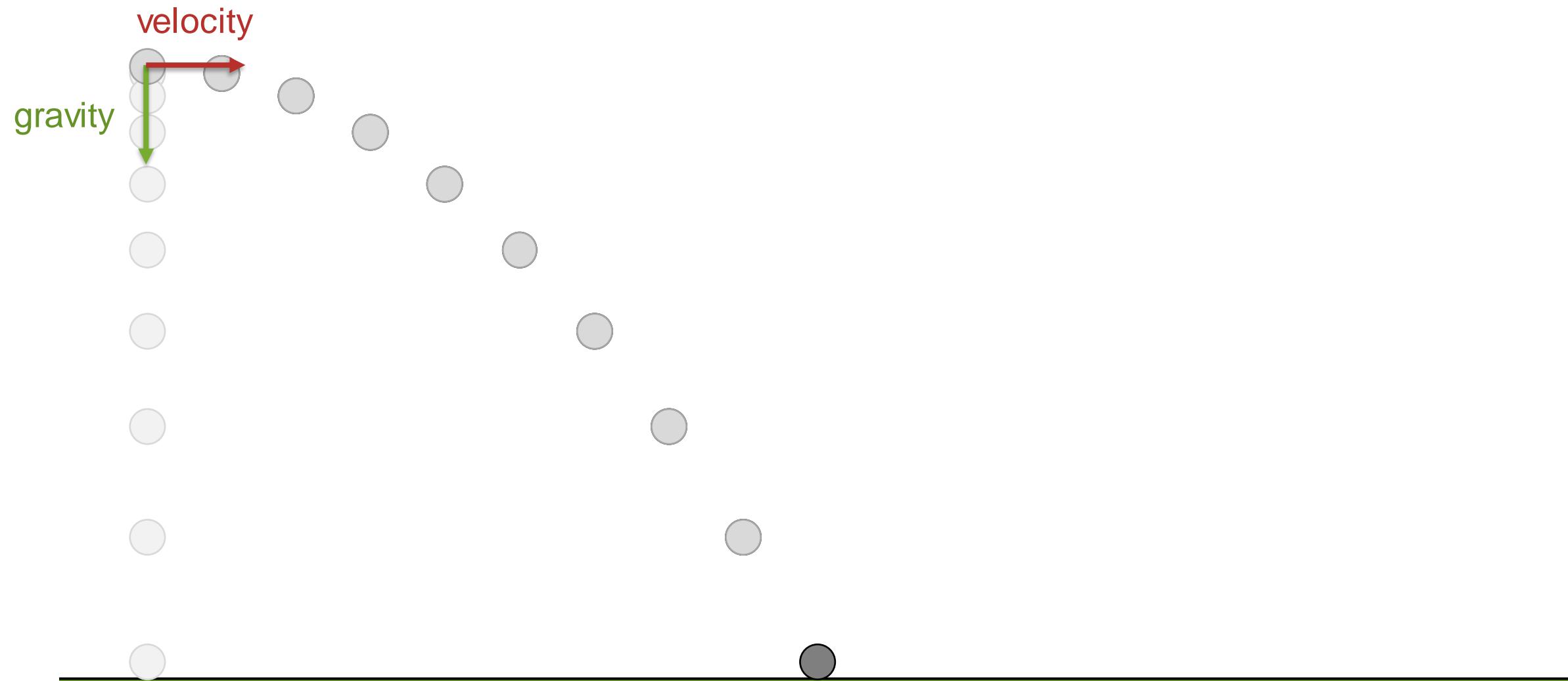
What will happen when this object is released with no velocity? Why?



# What if we give the object a horizontal velocity?



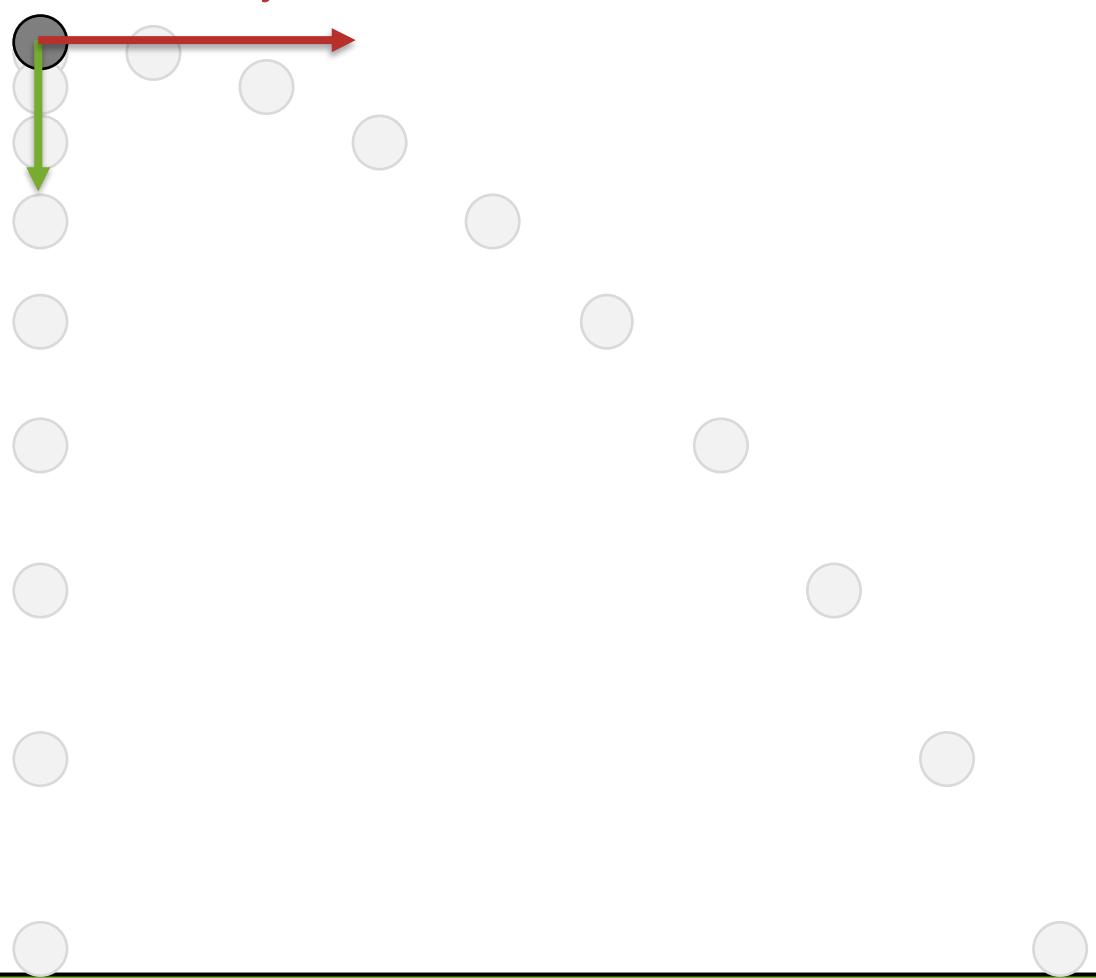
# What if we give the object a horizontal velocity?



# What if we give the object a bigger horizontal velocity?

velocity

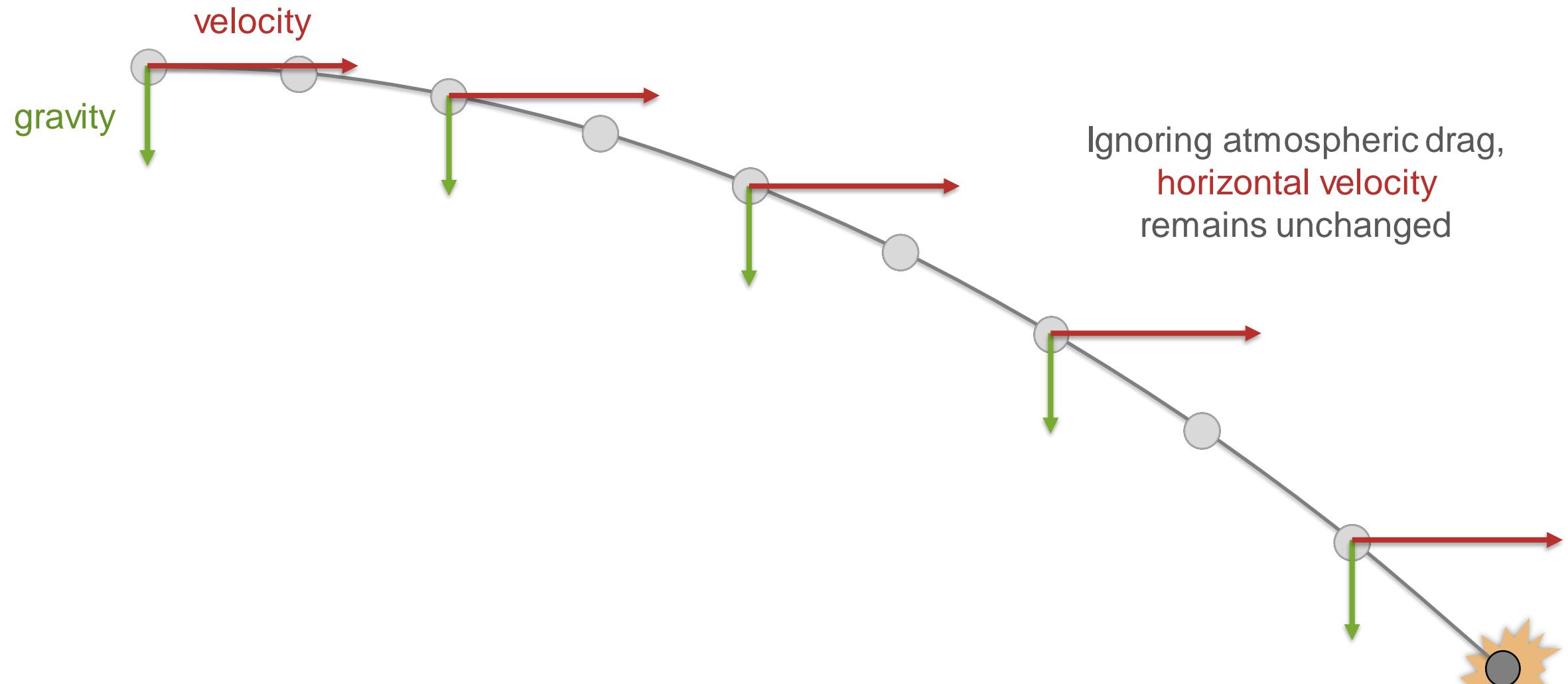
gravity



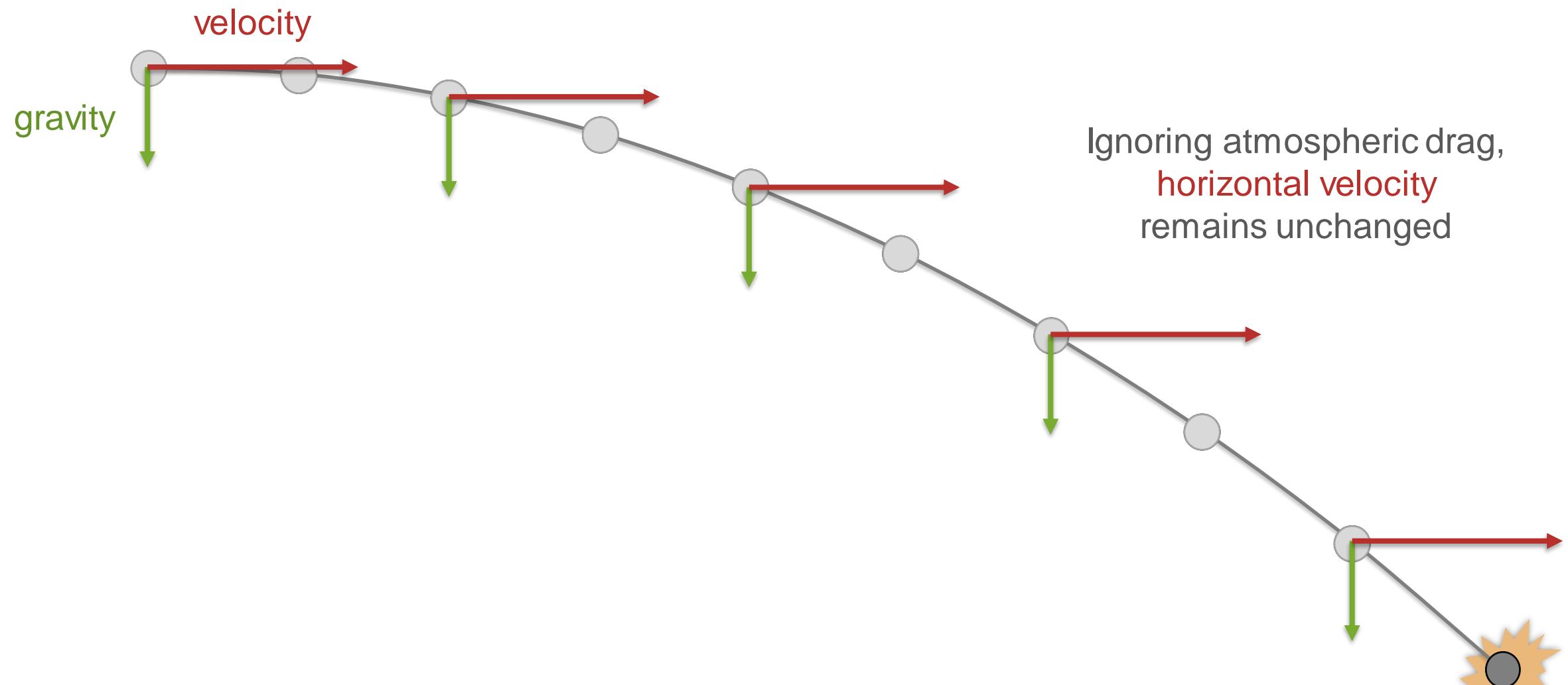
# What if we give the object a bigger horizontal velocity?



# Why does the object stop?

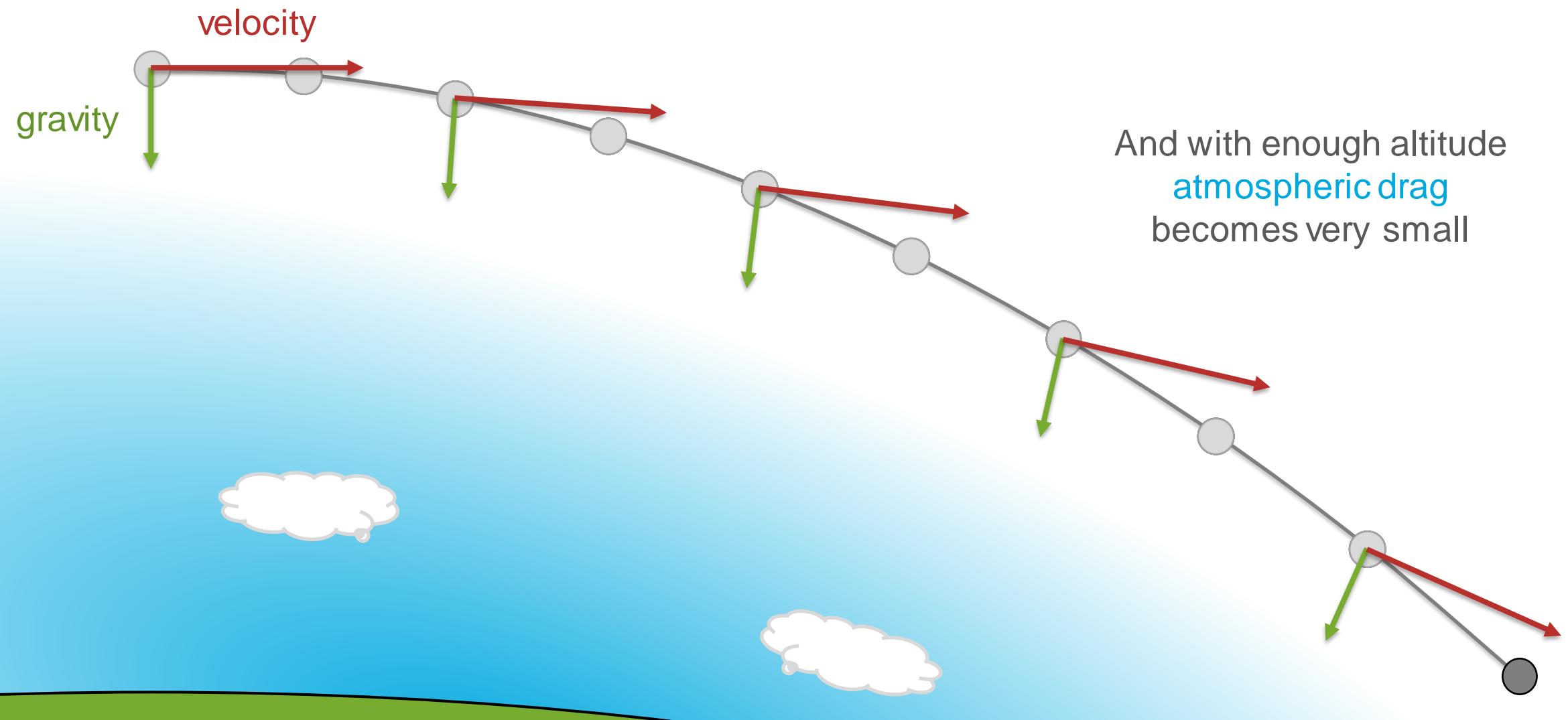


Why does the object stop? What happens to the surface further out?

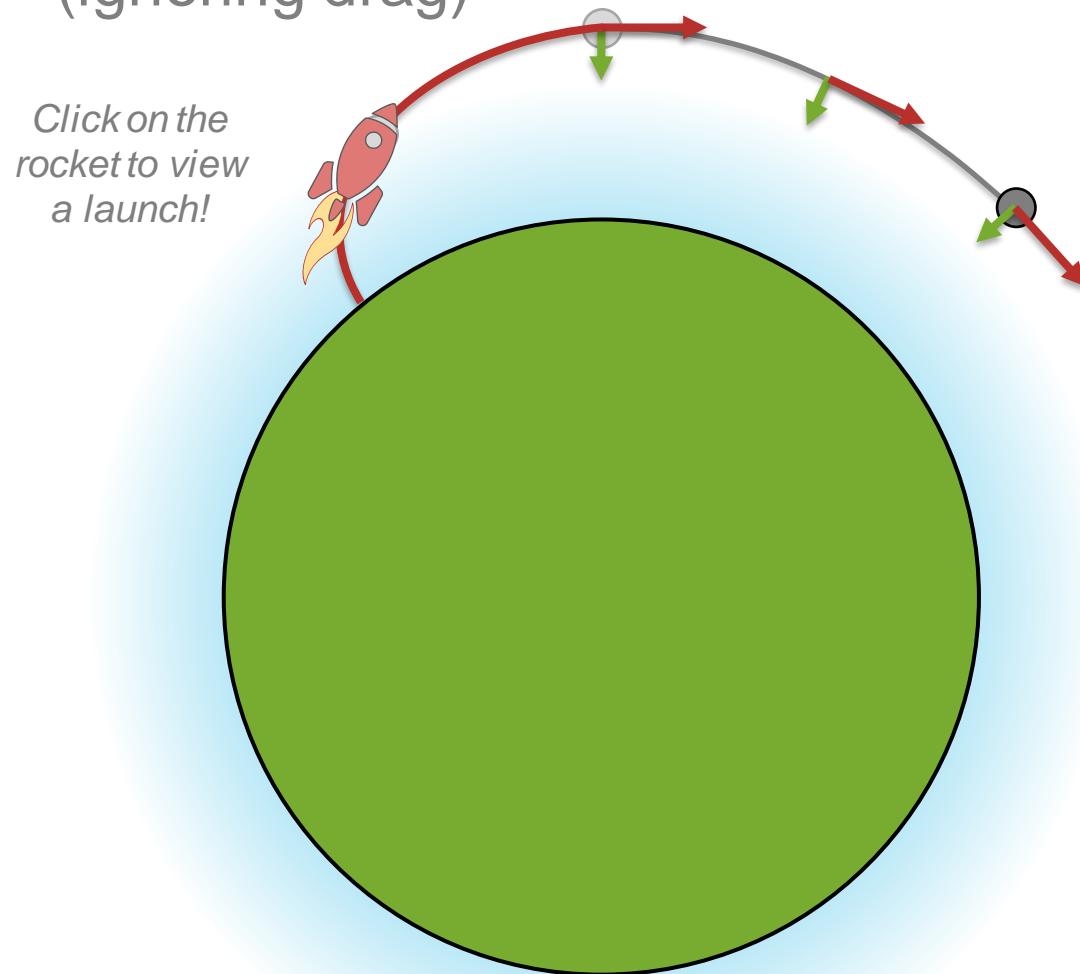


Is the surface flat?

With enough velocity the object falls ***with*** the curve of the surface!



# Objects in orbit are in constant *free fall* about the gravitational body (ignoring drag)

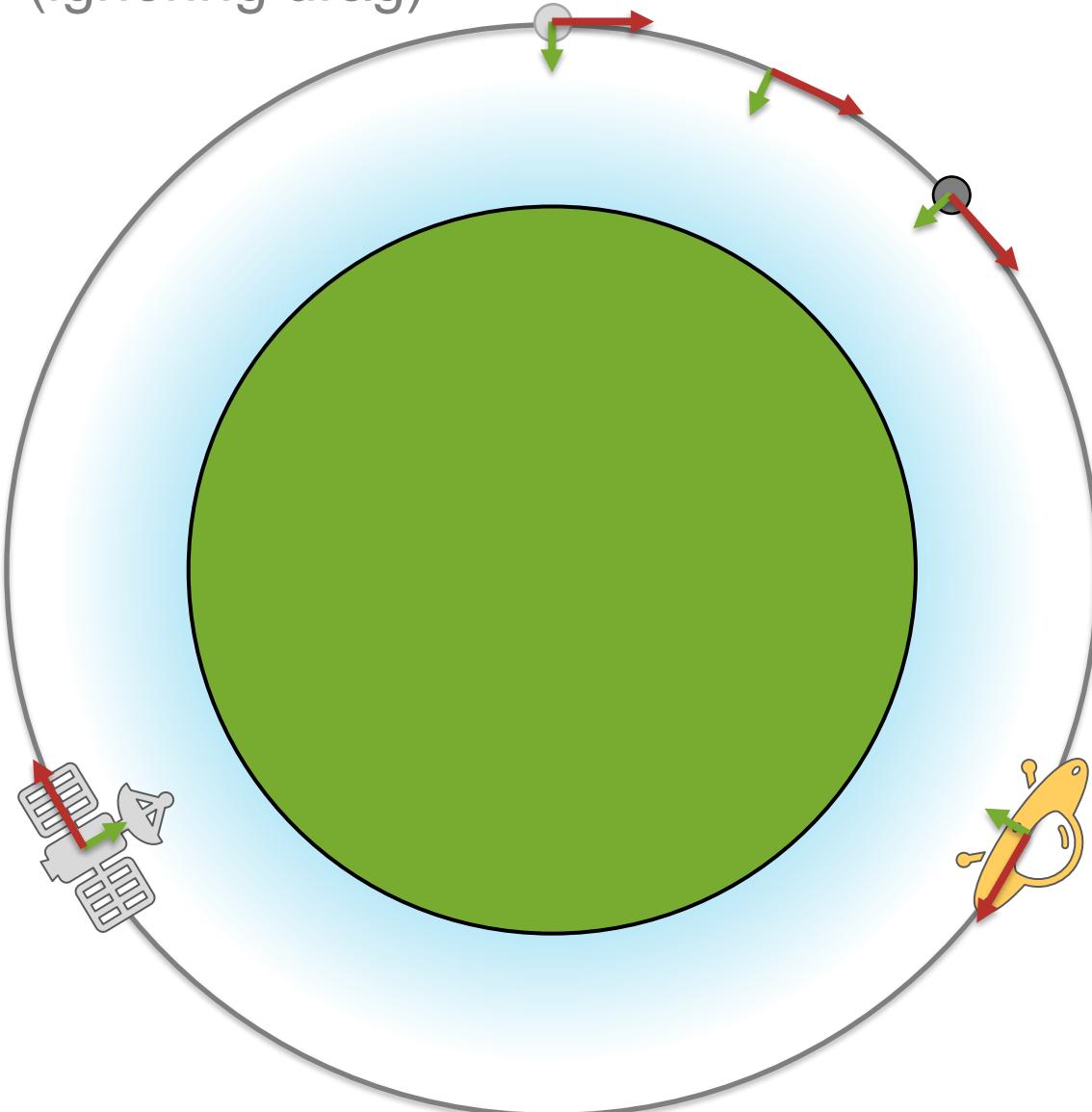


"Free fall" is the motion of an object acted upon solely by gravity (no drag)



"Freefalling at Skydive Miami"  
by Norcal21jg created 20 Sept 2010 (Public Domain)

Objects in orbit are in constant *free fall* about the gravitational body  
(ignoring drag)



"Free fall" is the motion of an object acted upon solely by gravity (no drag)

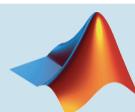


"Freeflying at Skydive Miami"  
by Norcal21jg created 20 Sept 2010 (Public Domain)

>> run OrbitalVelocity

# Course Lesson Plan

- Installing and Playing AstroVolley
- Plotting Ships (Triangles)
- Plotting Grav. Bodies (Circles)
- Gravity and Orbits



## Launching Volleys

- Creating a Mission Map Function
- Plotting Velocity Vectors (quiver)
- Animating Trajectories (comet)

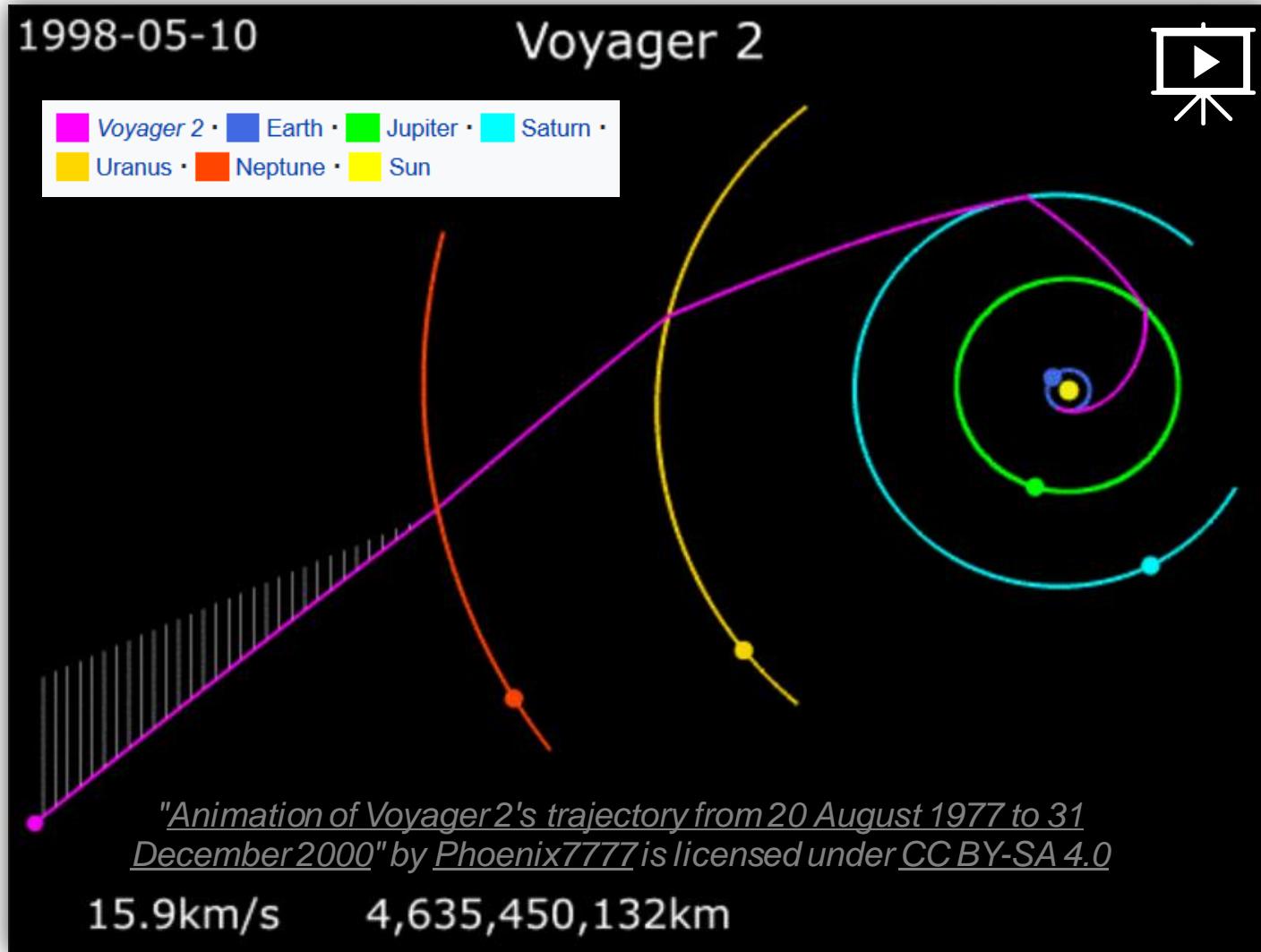
## App Designer

- 1) UIAxes, Properties, & Functions
- 2) Sliders & Value Changing Callback
- 3) Push Button Callback
- 4) Generating Random Maps
- 5) Keeping Score

```
>> open mapMission.m
```

# A Real Ongoing Astro Volley!

Animation of Voyager 2's Trajectory from 20 Aug 1977 to 31 Dec 2000

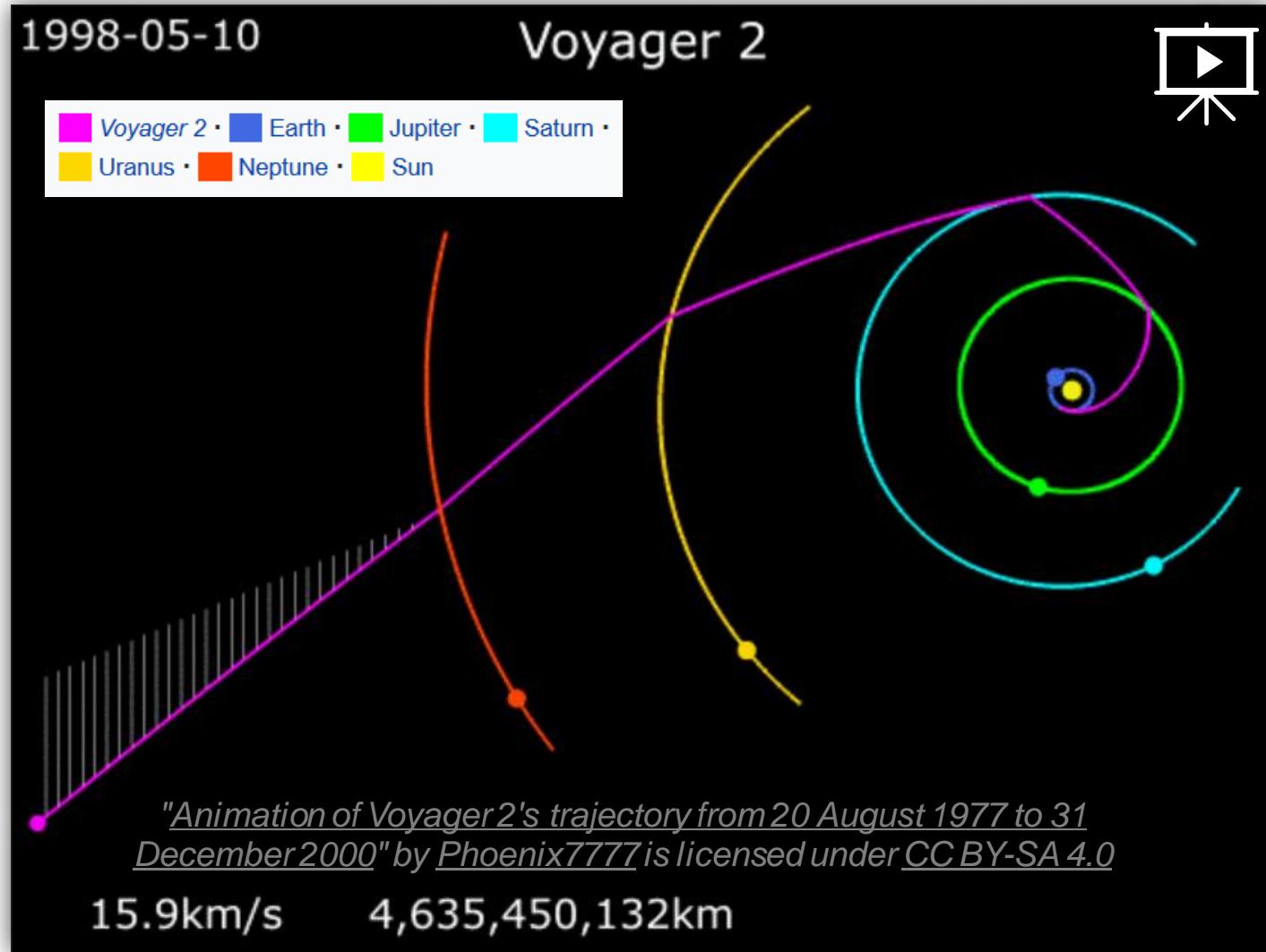


Observations...



# A Real Ongoing Astro Volley!

Animation of Voyager 2's Trajectory from 20 Aug 1977 to 31 Dec 2000



## Observations...

- Voyager 2 and each planet are in orbit about the Sun
- Voyager 2's trajectory gets deflected when it flies by another planet

## Just a Note...

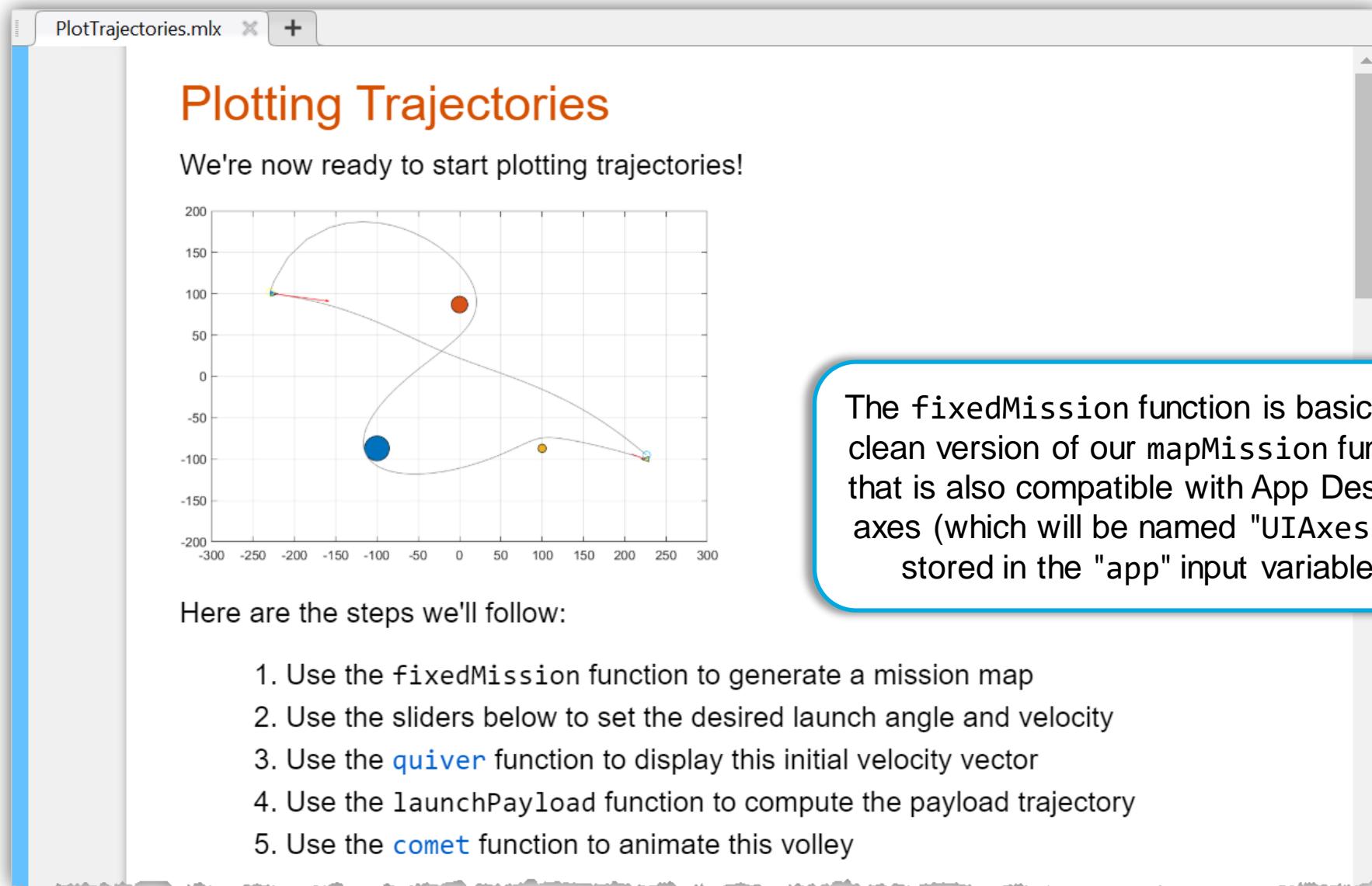
- Gravity of each body pulls on all the others (e.g. tides)
- For simplicity, we'll ignore these interactions and "lock" each body in place

# Convert mapMission to a function and try calling it

```
function app = mapMission  
ax = gca; % get axes  
cla(ax) % clear axes  
app.xs = [-225 225]; % ship x-pos  
app.ys = [ 100 -100]; % ship y-pos  
app.Rs = [ 5 5]; % ship sizes  
cs = [0 .73 .96; 1 1 .4]; % ship colors  
  
xt = [-5 4 4 -5]/5; % triangle x-points  
yt = [ 0 3 -3 0]/5; % triangle y-points  
  
fill(ax,app.xs(1)-app.Rs(1)*xt, ...  
     app.ys(1)+app.Rs(1)*yt,cs(1,:))  
hold(ax,"on")  
fill(ax,app.xs(2)+app.Rs(2)*xt, ...  
     app.ys(2)+app.Rs(2)*yt,cs(2,:))
```

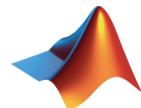
```
app.xb = [-100 0 100]; % body x-pos  
app.yb = [ -87 87 -87]; % body y-pos  
app.Rb = [ 15 10 5]; % body sizes  
cb = lines(3); % body colors  
  
ang = 0:5:360; % angle, degrees  
xc = cosd(ang); % circle x-points  
yc = sind(ang); % circle y-points  
for k = 1:3  
    fill(ax,app.xb(k)+app.Rb(k)*xc, ...  
         app.yb(k)+app.Rb(k)*yc,cb(k,:))  
end  
axis(ax,"equal")  
xlim(ax,[-300 300]); ylim(ax,[-200 200])  
end % function
```

```
>> open PlotTrajectories mlx
```



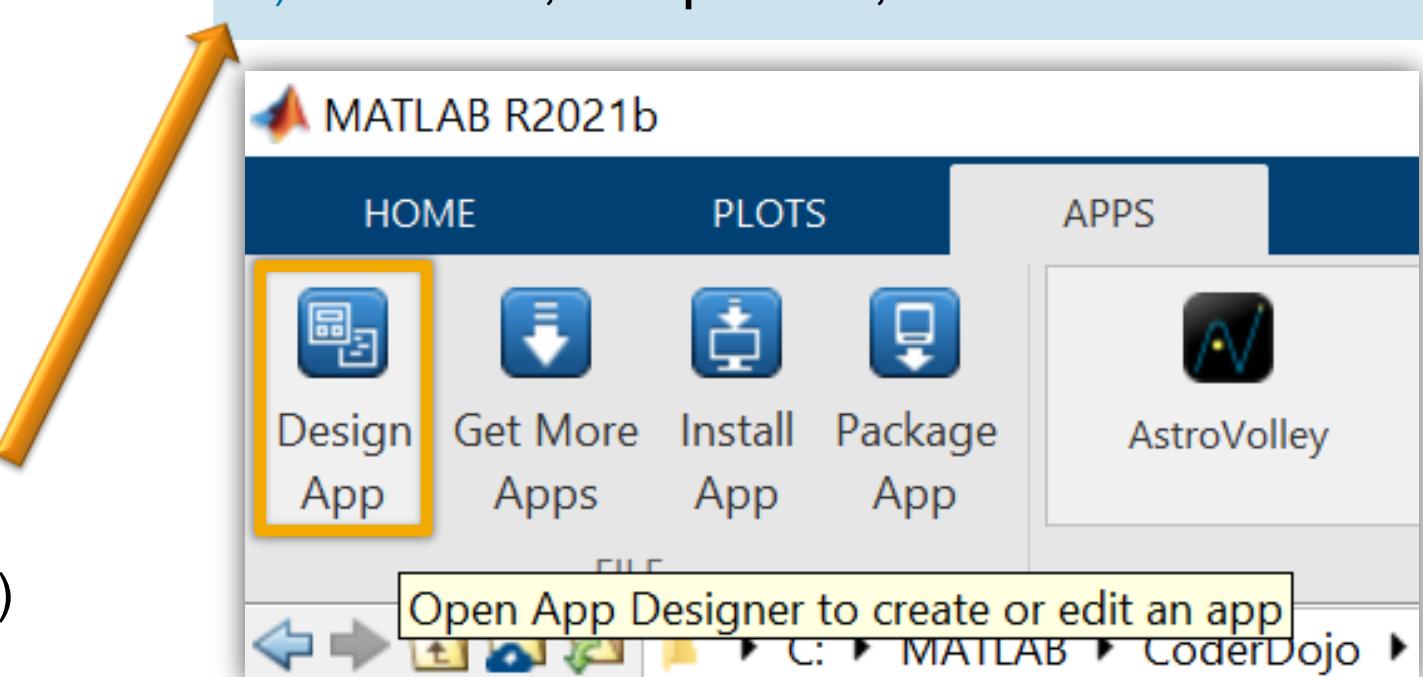
# Course Lesson Plan

- Installing and Playing AstroVolley
- Plotting Ships (Triangles)
- Plotting Grav. Bodies (Circles)
- Gravity and Orbits
- Launching Volleys
  - Creating a Mission Map Function
  - Plotting Velocity Vectors (quiver)
  - Animating Trajectories (comet)



## App Designer

### 1) UIAxes, Properties, & Functions



## MATLAB® App Designer

New to App Designer? Try a 3-minute tutorial.

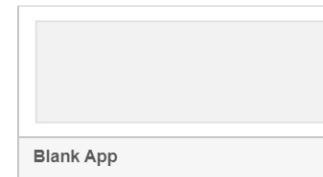
[Start Tutorial](#)[Open...](#)

## Recent Apps

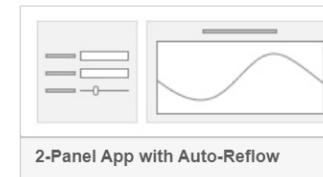
No Recent Apps

[Getting Started](#) | [How-to Videos](#) | [GUIDE Migration Strategies](#) | [Display Graphics in App Designer](#) | [Release Notes](#)

## ▼ New



Blank App

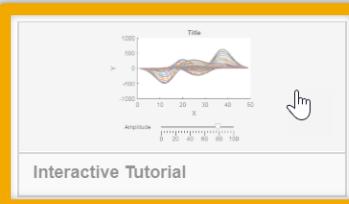


2-Panel App with Auto-Reflow

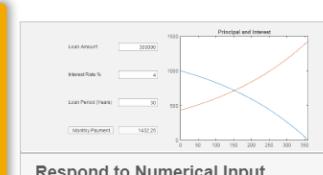


3-Panel App with Auto-Reflow

## ▼ Examples: General



Interactive Tutorial



Respond to Numerical Input



Respond to User Selections



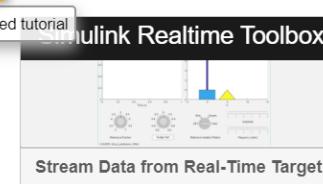
Embed HTML Content



Lay Out Controls in a Grid

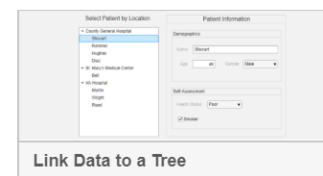


Display and Animate Flight Status



Stream Data from Real-Time Target

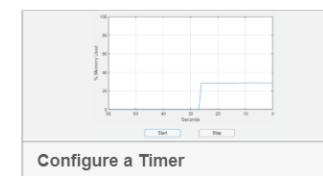
## ▼ Examples: Programming Tasks



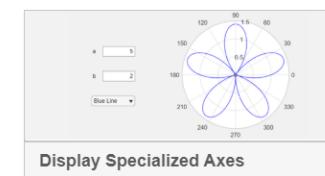
Link Data to a Tree



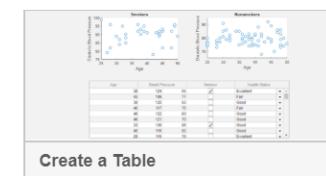
Analyze an Image



Configure a Timer



Display Specialized Axes



Create a Table

DESIGNER

CANVAS

VIEW



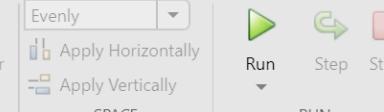
Save Convert



ALIGN



ARRANGE



SPACE



Run



Step



Stop



RUN

MyAstroVolley.mlapp x

Component Library

Search

COMMON



Axes



Push



Check Box



Date Picker



Drop Down



Edit Field (Numeric)



Edit Field



HTML



Hyperlink



Image



Label



List Box



Radio Button Group



Slider



Spinner



State Button



Table



Text Area



Toggle Button Group

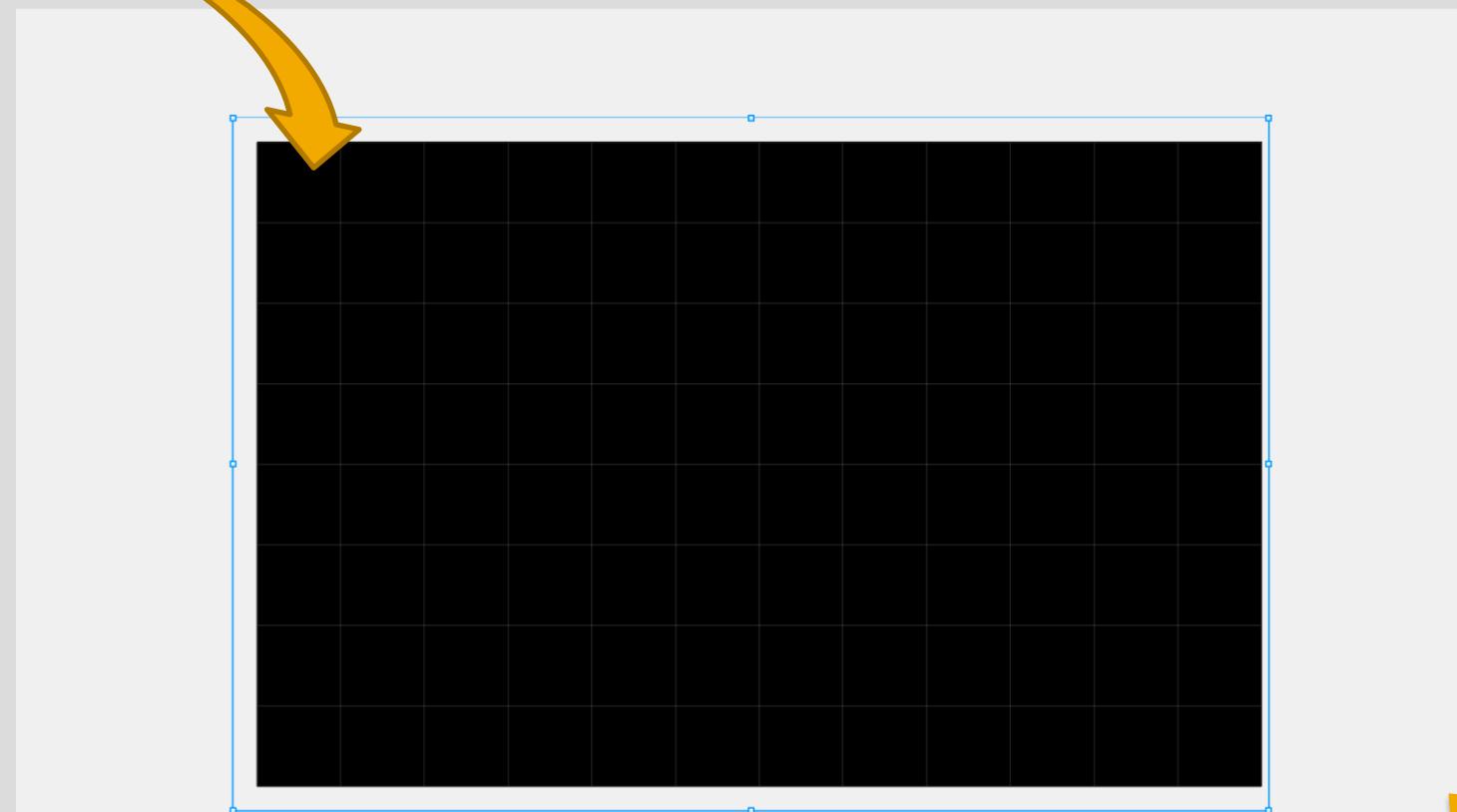


Tree



Tree (Check Box)

CONTAINERS



Design View Code View

Component Browser

Search

MyAstroVolley

app.UIFigure

app.UIAxes

LABELS

Title.String

XLabel.String

YLabel.String

ZLabel.String

Subtitle.String

TitleHorizontalAlignment center

FONT

TICKS

XTick

XTickLabel

YTick

YTickLabel

RULERS

GRIDS

XGrid

YGrid

ZGrid

GridColor

Color



1.00,1.00,



0.00,0.00,

65



DESIGNER CANVAS VIEW

FILE ALIGN ARRANGE RUN

Evenly Same Size Grouping Reorder

Run Step Stop

SPACE

MyAstroVolley.mlapp

Component Library

Search

Design View Code View

Component Browser

Search

MyAstroVolley app.UIFigure app.UIAxes

Axes | Callbacks

Search

Labels

Title.String  
XLabel.String  
YLabel.String  
ZLabel.String  
Subtitle.String  
TitleHorizontalAlignment center

Font

FontName Helvetica  
FontWeight Bold  
FontSize 12

Ticks

XTick -300,-250,-200,-150,-1  
XTickLabel  
YTICK -200,-150,-100,-50,0,5  
YTICKLABEL

Axes

COMMON

Axes Button Check Box

Date Picker Drop Down Edit Field (Numeric)

Edit Field (Text) HTML Hyperlink

Image Label List Box

Radio Button Group Slider Spinner

State Button Table Text Area

Toggle Button Group Tree Tree (Check Box)

CONTAINERS

The screenshot shows the MATLAB App Designer interface. The top menu bar includes DESIGNER, CANVAS, and VIEW. Below the menu are various toolbar icons for file operations (Save, Convert), alignment (Evenly, Same Size, Grouping, Reorder), and run control (Run, Step, Stop). The main workspace displays a large black rectangular component with a blue border, representing an axes or figure. To the left is the Component Library with sections for COMMON components like Axes, Button, and Check Box, and CONTAINERS like Table and Tree. To the right is the Component Browser showing the current application structure: MyAstroVolley > app.UIFigure > app.UIAxes. The Code View tab is highlighted with a yellow box. The Properties panel on the right shows settings for the selected axes component, including Labels, Font, and Ticks.

DESIGNER EDITOR VIEW

FILE NAVIGATE INSERT CODE RUN

MyAstroVolley.mlapp

Code Browser

Callbacks | Functions Properties

Search

app.xs  
app.ys  
app.Rs  
app.xb  
app.yb  
app.Rb

Properties

Private Property Private properties store data to be shared within the app only

Public Property Public properties store data to be shared inside and outside of the app

Design View Code View

Component Browser

Search

MyAstroVolley

app.UIFigure  
app.UIAxes

App Layout

App | Callbacks

Search

SHARING DETAILS

CODE OPTIONS

```
classdef MyAstroVolley < matlab.apps.AppBase
    % Properties
    properties (Access = public)
        xs      % ship x-positions
        ys      % ship y-positions
        Rs      % ship size
        xb      % body x-positions
        yb      % body y-positions
        Rb      % body sizes
    end

    % Callbacks that handle component events
    methods (Access = private)

        % Code that executes after component creation
        function startupFcn(app)
            fixedMission(app);
        end
    end

    % Component initialization
    methods (Access = private)

        % Create UIFigure and components
        function createComponents(app)

            % Create UIFigure and hide until all components are created
            app.UIFigure = uifigure('Visible', 'off');
```

DESIGNER EDITOR VIEW

FILE NAVIGATE INSERT CODE RUN

MyAstroVolley.mlapp

Code Browser

Callbacks Properties

Search

startupFcn

Component Browser

Design View Code View

MyAstroVolley

app.UIFigure

app.UIAxes

App | Callbacks

SHARING DETAILS

CODE OPTIONS

StartupFcn

MyAstroVolley

Properties

Access = public

UIFigure matlab.ui.Figure

UIAxes matlab.ui.control.UIAxes

end

properties (Access = public)

xs % ship x-position

ys % ship y-position

Rs % ship size

xb % body x-position

yb % body y-position

Rb % body sizes

end

% Code that executes after component creation

function startupFcn(app)

fixedMission(app);

end

% Component initialization

methods (Access = private)

% Create UIFigure and components

function createComponents(app)

% Create UIFigure and hide until all components are created

app.UIFigure = uifigure('Visible', 'off');

Add Callback Function

App: MyAstroVolley

Callback: StartupFcn

Name: startupFcn

Add Callback Cancel

DESIGNER

EDITOR

VIEW



Save



Print



Compare



Go To



Bookmark



Callback



Function



Property



App Input Arguments



App Help Text



Comment



%



%



%



Indent



Step



Stop

CODE

RUN



Run

MyAstroVolley.mlapp

Code Browser

Callbacks | Functions | Properties

Search



startupFcn

App Layout



```
1 classdef MyAstroVolley < matlab.apps.AppBase
2
3 % Properties that correspond to app components
4 properties (Access = public)
5     UIFigure matlab.ui.Figure
6     UIAxes matlab.ui.control.UIAxes
7 end
8
9
10 properties (Access = public)
11     xs % ship x-positions
12     ys % ship y-positions
13     Rs % ship size
14     xb % body x-positions
15     yb % body y-positions
16     Rb % body sizes
17 end
18
19
20 % Callbacks that handle component events
21 methods (Access = private)
22
23     % Code that executes after component creation
24     function startupFcn(app)
25         fixedMission(app);
26     end
27 end
28
29 % Component initialization
30 methods (Access = private)
31
32     % Create UIFigure and components
33     function createComponents(app)
34
35         % Create UIFigure and hide until all components are created
36         app.UIFigure = uifigure('Visible', 'off');
```

Design View

Code View

Component Browser

Search

MyAstroVolley

app.UIFigure

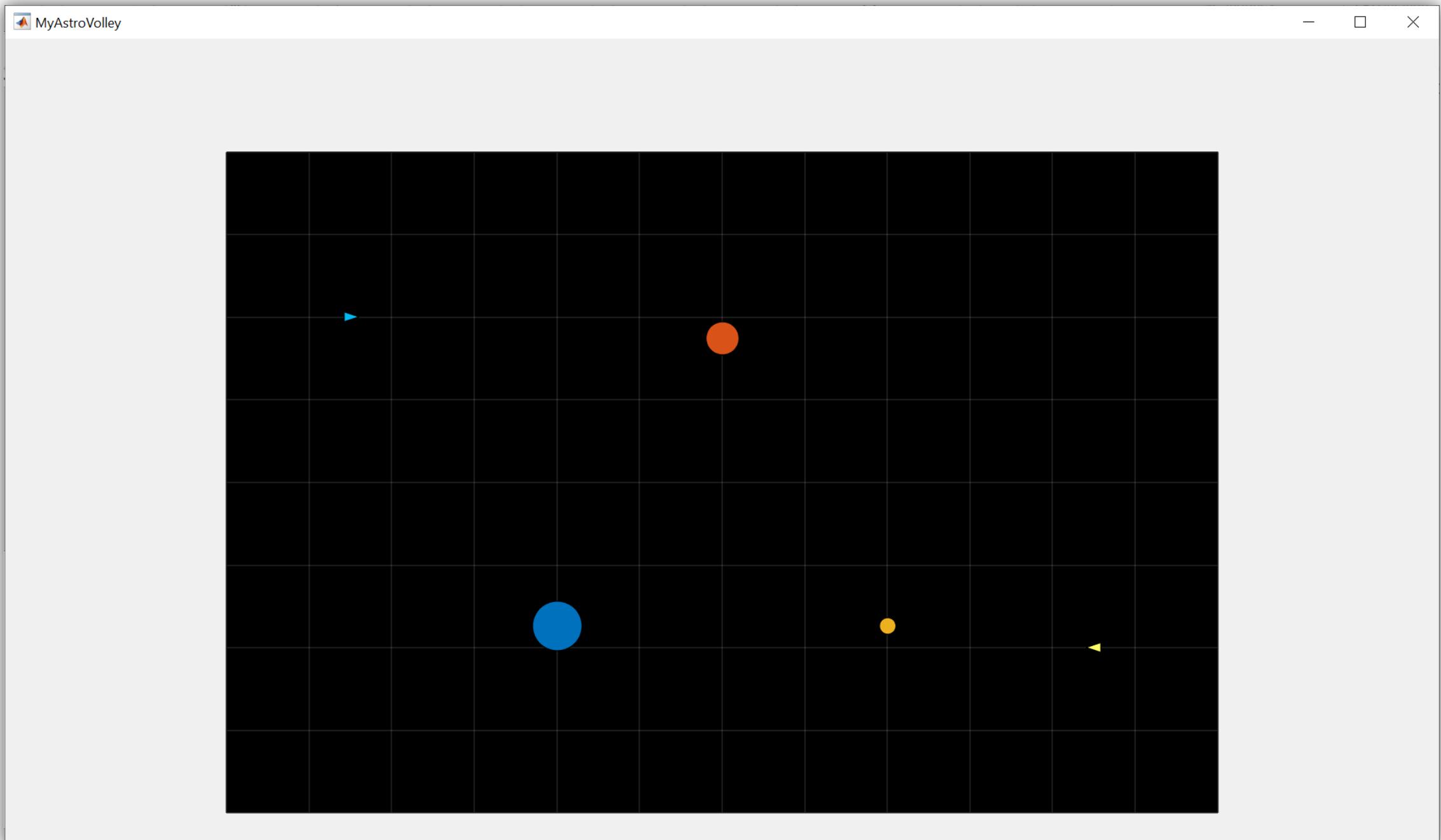
app.UIAxes

App | Callbacks

Search

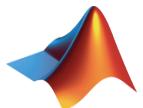
▶ SHARING DETAILS

▶ CODE OPTIONS



# Course Lesson Plan

- Installing and Playing AstroVolley
- Plotting Ships (Triangles)
- Plotting Grav. Bodies (Circles)
- Gravity and Orbits
- Launching Volleys
  - Creating a Mission Map Function
  - Plotting Velocity Vectors (quiver)
  - Animating Trajectories (comet)



## App Designer

- 1) UIAxes, Properties, & Functions
- 2) Sliders & Value Changing Callback
- 3) Push Button Callback
- 4) Generating Random Maps
- 5) Keeping Score



```
>> open MyAstroVolley1.mlapp
```

DESIGNER

CANVAS

VIEW



Save

Convert

FILE



ALIGN



Same Size

Grouping

Reorder



Evenly

Apply Horizontally

Apply Vertically



Run

Step

Stop

SPACE

RUN

MyAstroVolley.mlapp

Component Library

Search



COMMON



Axes



Button



Check Box



Date Picker



Drop Down



Edit Field (Numeric)



Edit Field (Text)



HTML



Hyperlink



Image



Label



List Box



Radio Button



Slider



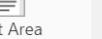
Spinner



State Button



Table



Text Area



Toggle Button



Tree



Tree (Check Box)



Design View Code View

Component Browser

Search

MyAstroVolley

app.UIFigure

app.AngleSlider1

app.UIAxes

Slider | Callbacks

Search

Label Angle(deg)

SLIDER

Value

Limits

Orientation

TICKS

MajorTicks

MajorTickLabels

MinorTicks

FONT

FontName

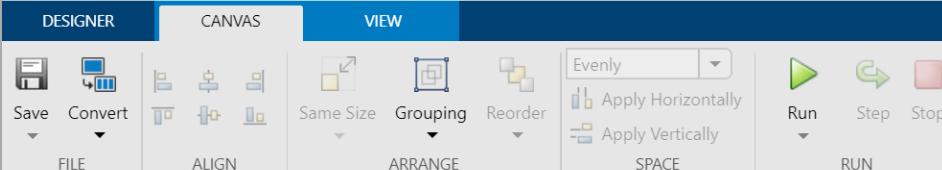
FontSize

FontWeight

FontAngle

FontColor

INTERACTIVITY



**NOTE:** Name slider components as indicated in the Component Browser for compatibility with "launchPayload" later.

MyAstroVolley.mlapp

Component Library

Search

Design View    Code View

COMMON

Axes	Button	Check Box
Date Picker	Drop Down	Edit Field (Numeric)
Edit Field (Text)	HTML	Hyperlink
Image	Label	List Box
Radio Button Group	Slider	Spinner
State Button	Table	Text Area
Toggle Button Group	Tree	Tree (Check Box)

Component Browser

Search

- MyAstroVolley
  - app.UIFigure
    - app.AngleSlider1

Slider | Callbacks

Search

Label: Angle(deg)

SLIDER

Value: 0

Limits: -90, 90

Orientation: Horizontal

TICKS

MajorTicks: -90, -80, -70, -60, -50, -40, -30, -20, -10, 0, 10, 20, 30, 40, 50, 60, 70, 80, 90

MajorTickLabels: -90,-80,-70,-60,-50,-40,-30,-20,-10,0,10,20,30,40,50,60,70,80,90

MinorTicks: -90,-88,-86,-84,-82,-80

FONT

FontName: Consolas

FontSize: 12

FontWeight: Normal

FontAngle: Normal

FontColor: 0.00,0.00,1

INTERACTIVITY

DESIGNER

CANVAS

VIEW



Save

Convert

FILE



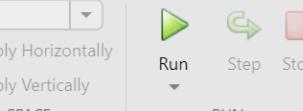
ALIGN



Same Size

Grouping

Reorder



Evenly

Apply Horizontally

Apply Vertically

SPACE

RUN

Run

Step

Stop

MyAstroVolley.mlapp

Design View Code View

Component Library

Search

COMMON



Axes



Button



Check Box



Date Picker



Drop Down



Edit Field (Numeric)



Edit Field (Text)



HTML



Hyperlink



Image



Label



List Box



Radio Button



Slider



Spinner



State Button



Table



Text Area



Toggle Button



Tree



Tree (Check Box)

CONTAINERS



Component Browser

Search

MyAstroVolley

app.UIFigure

app.AngleSlider1

app.UIAxes

Slider | Callbacks

Search

Label Angle(deg)

SLIDER

Value

Limits

Orientation

TICKS

MajorTicks

MajorTickLabels

MinorTicks

FONTS

FontName Consolas

FontSize 12

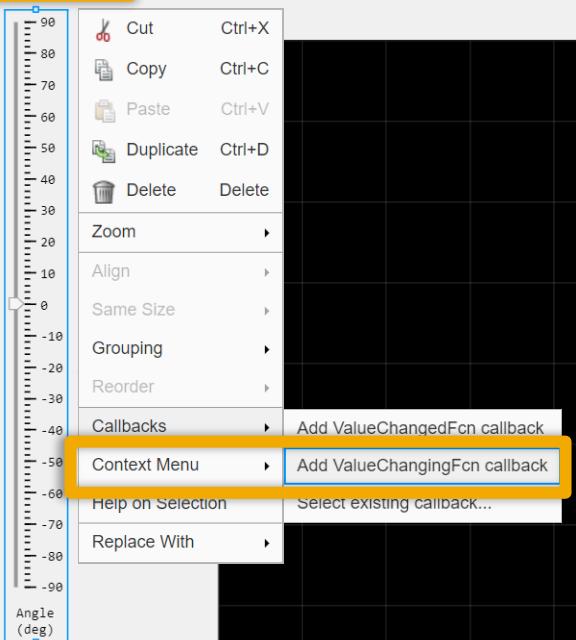
FontWeight B

FontAngle I

FontColor 0.00,0.00,1

INTERACTIVITY

Right-Click



DESIGNER

EDITOR

VIEW



Save



Print



Compare



Go To



Bookmark



Callback



Function



Property



App Input Arguments



App Help Text



Comment



Indent



Run



Step



Stop



CODE



RUN

MyAstroVolley.mlapp

## Code Browser

Callbacks | Functions | Properties

Search

```
11 properties (Access = public)
12     xs    % ship x-positions
13     ys    % ship y-positions
14     Rs    % ship size
15     xb    % body x-positions
16     yb    % body y-positions
17     Rh    % body sizes
18     hq    % quiver handle
19
20 end
```

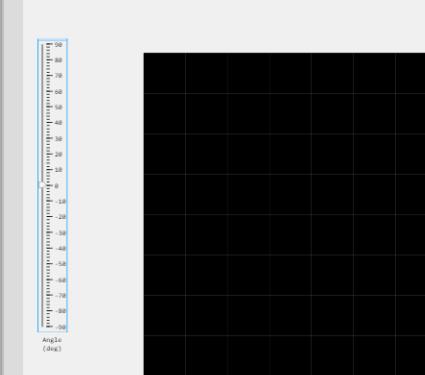
21

```
22
23 % Callbacks that handle component events
24 methods (Access = private)
```

25

```
26     % Code that executes after component creation
27     function startupFcn(app)
28         fixedMission(app);
29     end
```

## App Layout



```
30
31 % Value changing function: AngleSlider1
32 function AngleSlider1ValueChanging(app, event)
33     ang = event.Value;
34     vel = app.VelocitySlider1.Value;
35     vx = vel*cosd(ang);
36     vy = vel*sind(ang);
37     delete(app.hq)
38     app.hq = quiver(app.UIAxes,app.xs(1),app.ys(1),vx,vy,"Color","red");
39 end
40
```

```
41
42 % Component initialization
43 methods (Access = private)
```

```
44
45 % Create UIFigure and components
46 function createComponents(app)
```

Design View

Code View

## Component Browser

Search

MyAstroVolley

app.UIFigure

app.AngleSlider1

app.UIAxes

## Slider | Callbacks

Search

SLIDER

Value

0

Limits

-90,90

Orientation

TICKS

MajorTicks

-90,-80,-70,-60,-50,-40

MajorTickLabels

-90,-80,-70,-60,-50,-40

MinorTicks

-90,-88,-86,-84,-82,-80

FONT

FontName

Consolas

FontSize

12

FontWeight

DESIGNER

CANVAS

VIEW



Save

Convert

FILE

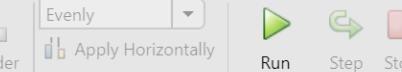


ALIGN



Same Size

Grouping



Evenly

Apply Horizontally

Space

Apply Vertically



Run

Step

Stop

RUN

MyAstroVolley.mlapp

Component Library

Search



COMMON



Axes



Button



Check Box



Date Picker



Drop Down



Edit Field (Numeric)



Edit Field (Text)



HTML



Hyperlink



Image



Label



List Box



Radio Button



Slider



Spinner



State Button



Table



Text Area



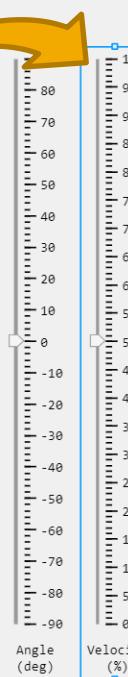
Toggle Button



Tree



Tree (Check Box)



Design View Code View

Component Browser

Search

MyAstroVolley

app.UIFigure

app.VelocitySlider1

app.AngleSlider1

app.UIAxes

Slider | Callbacks

Search

Label Velocity(%)

SLIDER

Value

Limits

Orientation

TICKS

MajorTicks

MajorTickLabels

MinorTicks

FONT

FontName

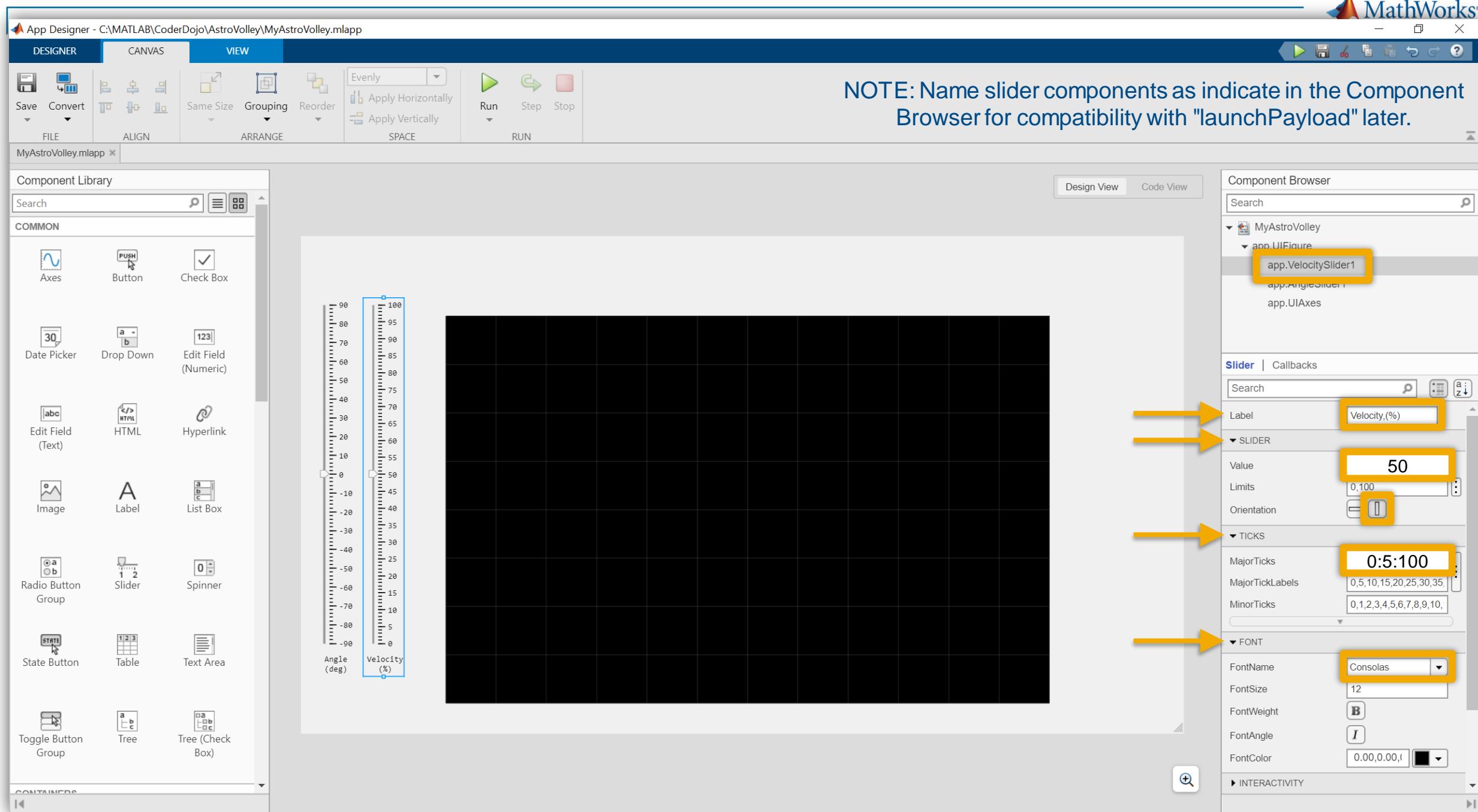
FontSize

FontWeight

FontAngle

FontColor

**NOTE:** Name slider components as indicated in the Component Browser for compatibility with "launchPayload" later.





DESIGNER

EDITOR

VIEW



Save



Print



Compare



Go To



Bookmark



Callback



Function



Property



App Input Arguments



App Help Text



Comment



Indent



Run



Step



Stop



CODE



RUN

MyAstroVolley.mlapp

## Code Browser

Callbacks | Functions | Properties

Search

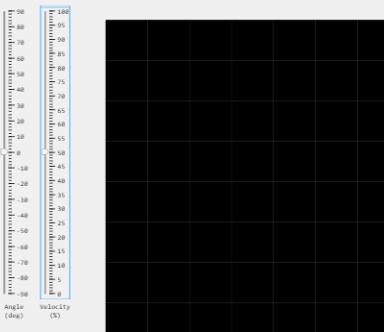


startupFcn

AngleSlider1ValueChanging

VelocitySlider1ValueChanging

## App Layout



```

17     Rs    % ship size
18     xb    % body x-positions
19     yb    % body y-positions
20     Rb    % body sizes
21     hq    % quiver handle
22 end

23

24 % Callbacks that handle component events
25 methods (Access = private)
26
27     % Code that executes after component creation
28 function startupFcn(app)
29     fixedMission(app);
30 end
31
32
33 % Value changing function: AngleSlider1
34 function AngleSlider1ValueChanging(app, event)
35     ang = event.Value;
36     vel = app.VelocitySlider1.Value;
37     vx = vel*cosd(ang);
38     vy = vel*sind(ang);
39     delete(app.hq)
40     app.hq = quiver(app.UIAxes,app.xs(1),app.ys(1),vx,vy,"Color","red");
41 end
42
43 % Value changing function: VelocitySlider1
44 function VelocitySlider1ValueChanging(app, event)
45     ang = app.AngleSlider1.Value;
46     vel = event.Value; 
47     vx = vel*cosd(ang);
48     vy = vel*sind(ang);
49     delete(app.hq)
50     app.hq = quiver(app.UIAxes,app.xs(1),app.ys(1),vx,vy,"Color","red");
51 end
52

```

Design View Code View

## Component Browser

Search

MyAstroVolley

app.UIFigure

app.VelocitySlider1

app.AngleSlider1

app.UIAxes

## Slider | Callbacks

Search

SLIDER

Value

50

Limits

0,100

Orientation



TICKS

MajorTicks

0,5,10,15,20,25,30,35

MajorTickLabels

0,5,10,15,20,25,30,35

MinorTicks

0,1,2,3,4,5,6,7,8,9,10,

FONT

FontName

Consolas

FontSize

12

FontWeight



DESIGNER

EDITOR

VIEW



Save



Print



Compare



Go To



Bookmark



Callback



Function



Property



App Input Arguments



App Help Text



Comment



Indent



Step



Stop



Run



CODE



RUN

MyAstroVolley.mlapp

## Code Browser

Callbacks | Functions | Properties

Search

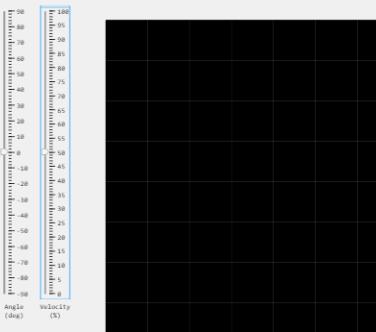


startupFcn

AngleSlider1ValueChanging

VelocitySlider1ValueChanging

## App Layout



Design View Code View

```

17     Rs    % ship size
18     xb    % body x-positions
19     yb    % body y-positions
20     Rb    % body sizes
21     hq    % quiver handle
22 end
23
24 % Callbacks that handle component events
25 methods (Access = private)
26
27     % Code that executes after component creation
28 function startupFcn(app)
29     fixedMission(app);
30 end
31
32
33     % Value changing function: AngleSlider1
34 function AngleSlider1ValueChanging(app, event)
35     ang = event.Value;
36     vel = app.VelocitySlider1.Value;
37     vx = vel*cosd(ang);
38     vy = vel*sind(ang);
39     delete(app.hq)
40     app.hq = quiver(app.UIAxes,app.xs(1),app.ys(1),vx,vy,"Color","red");
41 end
42
43     % Value changing function: VelocitySlider1
44 function VelocitySlider1ValueChanging(app, event)
45     ang = app.AngleSlider1.Value;
46     vel = event.Value;
47     vx = vel*cosd(ang);
48     vy = vel*sind(ang);
49     delete(app.hq)
50     app.hq = quiver(app.UIAxes,app.xs(1),app.ys(1),vx,vy,"Color","red");
51 end
52 end

```

## Component Browser

Search

MyAstroVolley

app.UIFigure

app.VelocitySlider1

app.AngleSlider1

app.UIAxes

## Slider | Callbacks

Search

SLIDER

Value

50

Limits

0,100

Orientation



TICKS

MajorTicks

0,5,10,15,20,25,30,35

MajorTickLabels

0,5,10,15,20,25,30,35

MinorTicks

0,1,2,3,4,5,6,7,8,9,10,

FONT

FontName

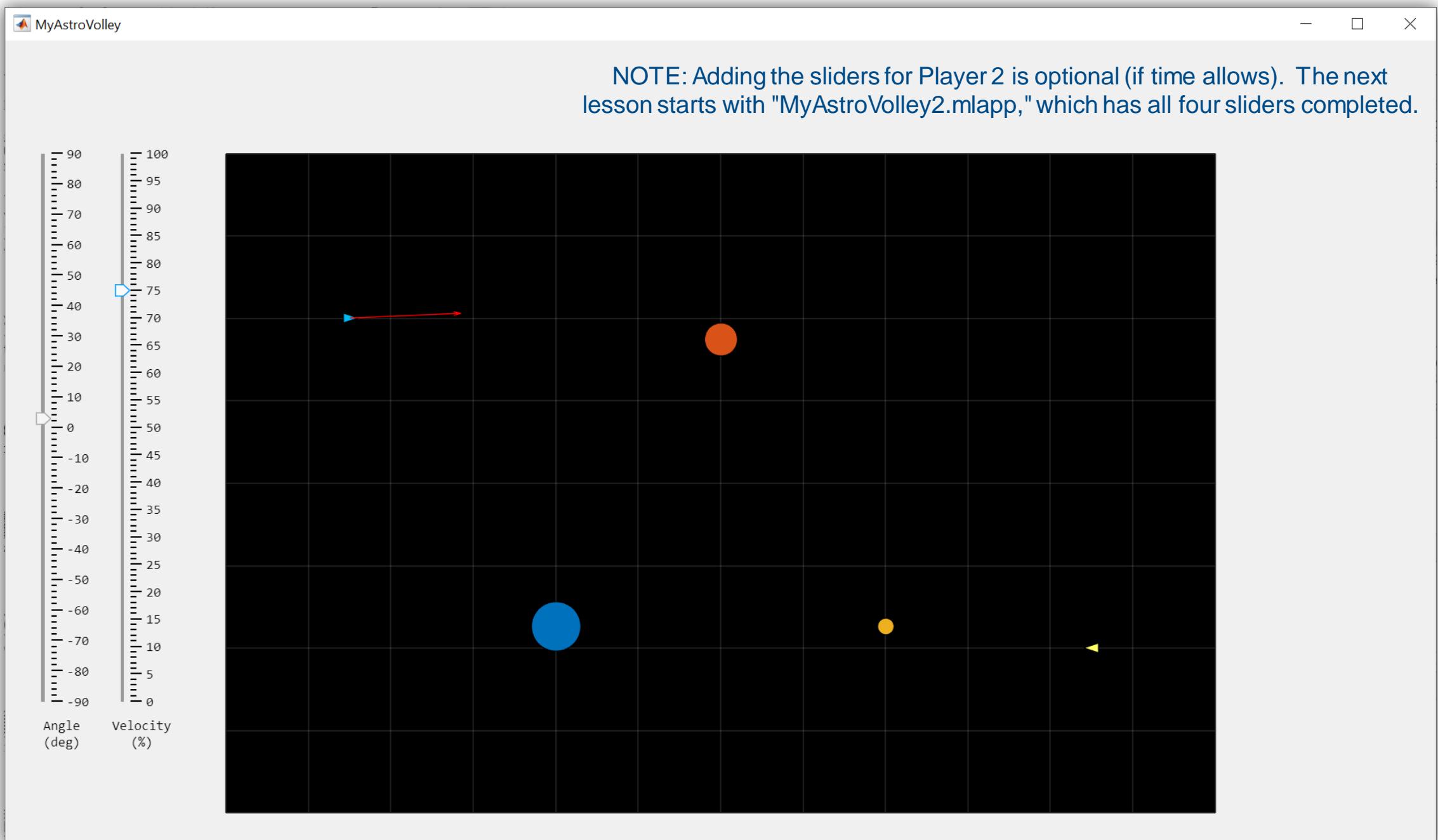
Consolas

FontSize

12

FontWeight





DESIGNER

CANVAS

VIEW



Save

Convert

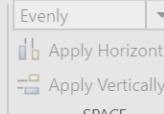
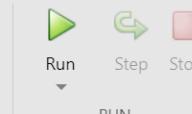
FILE



ALIGN

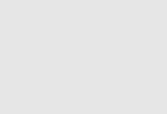


GROUPING REORDER

EVENLY  
SAME SIZE  
ARRANGERUN  
STEP  
STOP

SPACE

RUN

APPLY HORIZONTALLY  
APPLY VERTICALLY

RUN

STEP

STOP

MyAstroVolley.mlapp

Component Library

Search



COMMON



Axes



Button



Check Box



Date Picker



Drop Down



Edit Field (Numeric)



Edit Field (Text)



HTML



Hyperlink



Image



List Box



Radio Button Group



Slider



Spinner



State Button



Table



Text Area



Toggle Button Group



Tree

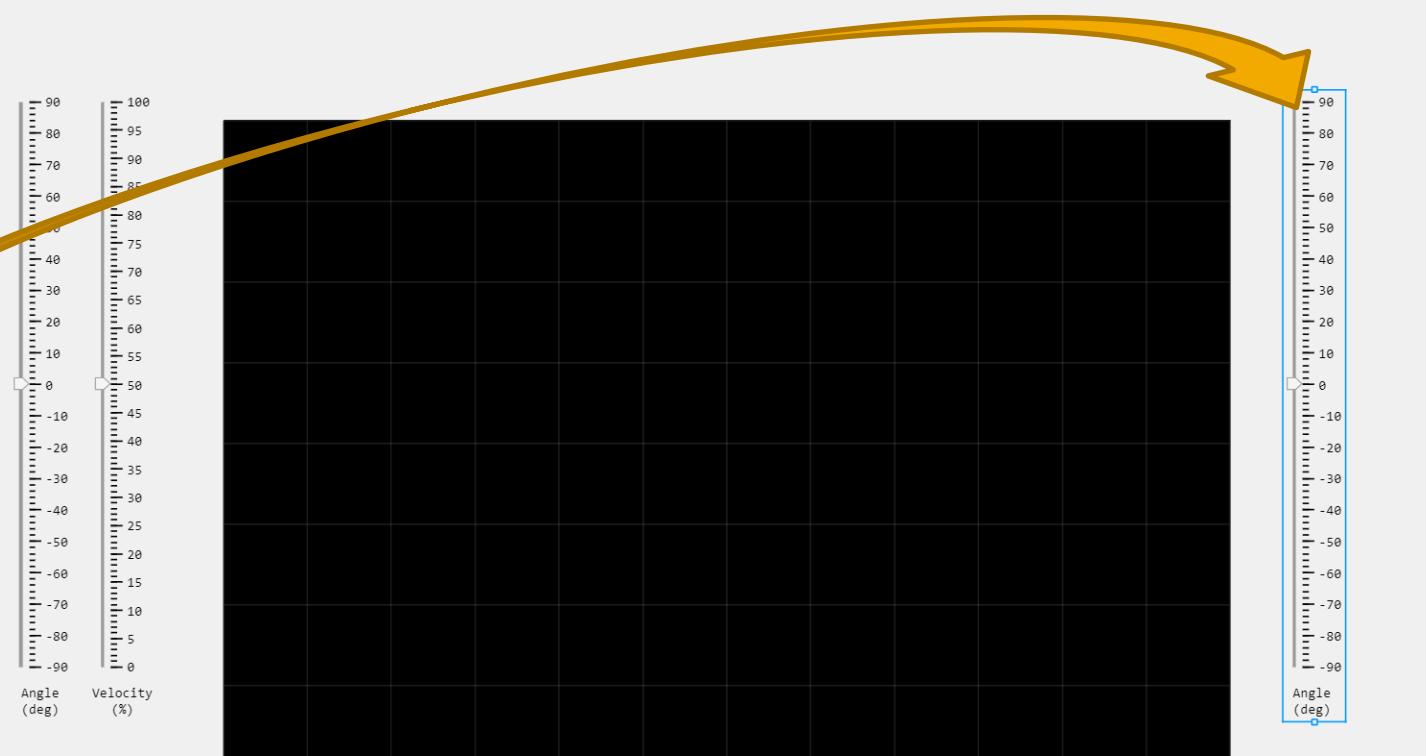


Tree (Check Box)

CONTAINERS



Design View Code View



Component Browser

Search

MyAstroVolley

app.UIFigure

app.AngleSlider2

app.VelocitySlider1

app.AngleSlider1

app.UIAxes

Slider | Callbacks

Search

Label Angle(deg)

SLIDER

Value

Limits

Orientation

TICKS

MajorTicks

MajorTickLabels

MinorTicks

FONT

FontName Consolas

FontSize

FontWeight

FontAngle

FontColor 0.00,0.00,1

**NOTE: Name slider components as indicated in the Component Browser for compatibility with "launchPayload" later.**

The screenshot shows the MATLAB App Designer interface. The top menu bar includes DESIGNER, CANVAS, and VIEW. The toolbar contains icons for Save, Convert, FILE, ALIGN, ARRANGE, and RUN. The Component Library on the left lists various UI components like Axes, Button, Check Box, Date Picker, Drop Down, Edit Field (Numeric), Edit Field (Text), HTML, Hyperlink, Image, Label, List Box, Radio Button Group, Slider, Spinner, State Button, Table, Text Area, Toggle Button Group, Tree, and Tree (Check Box). The main canvas area displays two vertical sliders. The left slider is labeled 'Angle (deg)' and has major ticks from -90 to 100 in increments of 5. The right slider is labeled 'Velocity (%)' and has major ticks from -90 to 100 in increments of 5. The Component Browser on the right shows the app structure under 'MyAstroVolley' and 'app.UIFigure'. It highlights 'app.AngleSlider2' with a yellow box. The properties for 'app.AngleSlider2' are shown, including 'Label' (Angle (deg)), 'Value' (0), 'Limits' (-90, 90), 'Orientation' (Vertical), 'TICKS' (MajorTicks: -90:-80:-70:-60:-50:-40, MinorTicks: -90:-88:-86:-84:-82:-80), and 'FONT' (FontName: Consolas, FontSize: 12, FontWeight: Normal, FontAngle: Normal, FontColor: 0.00, 0.00, 1). Arrows point from the sliders in the canvas to their corresponding properties in the Component Browser.

Design View    Code View

Component Library

Search

COMMON

- Axes
- Button
- Check Box
- Date Picker
- Drop Down
- Edit Field (Numeric)
- Edit Field (Text)
- HTML
- Hyperlink
- Image
- Label
- List Box
- Radio Button Group
- Slider
- Spinner
- State Button
- Table
- Text Area
- Toggle Button Group
- Tree
- Tree (Check Box)

Angle (deg)    Velocity (%)

Component Browser

Search

MyAstroVolley

app.UIFigure

app.AngleSlider2

app.velocitySlider

app.AngleSlider1

app.UIAxes

Slider | Callbacks

Search

Label: Angle (deg)

Value: 0

Limits: -90, 90

Orientation: Vertical

TICKS

MajorTicks: -90:-80:-70:-60:-50:-40

MinorTicks: -90:-88:-86:-84:-82:-80

FONT

FontName: Consolas

FontSize: 12

FontWeight: Normal

FontAngle: Normal

FontColor: 0.00, 0.00, 1

INTERACTIVITY

DESIGNER

CANVAS

VIEW



Save

Convert

FILE



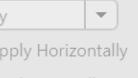
ALIGN



Same Size



Reorder



Evenly

Apply Horizontally

Apply Vertically



Run

Step

Stop



SPACE

RUN

MyAstroVolley.mlapp

Component Library

Search



COMMON



Axes



Button



Check Box



Date Picker



Drop Down



Edit Field (Numeric)



Edit Field (Text)



HTML



Hyperlink



Image



Label



List Box



Radio Button Group



Slider



Spinner



State Button



Table



Text Area



Toggle Button Group



Tree

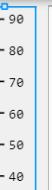


Tree (Check Box)

CONTAINERS



Right-Click



Cut

Ctrl+X

Copy

Ctrl+C

Paste

Ctrl+V

Duplicate

Ctrl+D

Delete

Delete



Zoom

►

Align

Same Size

Grouping

Reorder

►

Callbacks

►

Add ValueChangedFcn callback

►

Context Menu

►

Add ValueChangingFcn callback

►

Help on Selection

Select existing callback...

►

Replace With

►

FONTS

FontName

Consolas

FontSize

12

FontWeight

B

FontAngle

I

FontColor

0.00,0.00,1



Design View

Code View

Component Browser

Search

MyAstroVolley

app.UIFigure

app.AngleSlider2

app.VelocitySlider1

app.AngleSlider1

app.UIAxes

Callbacks

►

Angle(deg)

0

-90,90

►

Angle(deg)

-80,-40

►

Angle(deg)

-82,-8C

►

Angle(deg)

-80,-8C

►

Angle(deg)

-82,-8C

►

Angle(deg)

-80,-8C

►

Angle(deg)

-82,-8C

►

Angle(deg)

-80,-8C

►

Angle(deg)

-82,-8C

►

INTERACTIVITY

DESIGNER EDITOR VIEW

FILE NAVIGATE INSERT CODE RUN

MyAstroVolley.mlapp

Code Browser

Callbacks | Functions | Properties

Search

startupFcn

AngleSlider1ValueChanging

VelocitySlider1ValueChanging

AngleSlider2ValueChanging

VelocitySlider2ValueChanging

Design View Code View

Component Browser

Search

MyAstroVolley

app.UIFigure

app.VelocitySlider2

app.AngleSlider2

app.VelocitySlider1

app.AngleSlider1

app.UIAxes

Slider | Callbacks

Search

SLIDER

Value: 0

Limits: -90,90

Orientation:

TICKS

MajorTicks: -90,-80,-70,-60,-50,-40

MajorTickLabels: -90,-80,-70,-60,-50,-40

MinorTicks: -90,-88,-86,-84,-82,-80

FONT

FontName: Consolas

FontSize: 12

FontWeight: B

```
41         vx = vel*cosd(ang);
42         vy = vel*sind(ang);
43         delete(app.hq)
44         app.hq = quiver(app.UIAxes,app.xs(1),app.ys(1),vx,vy,"Color","red");
45     end
46
47     % Value changing function: VelocitySlider1
48     function VelocitySlider1ValueChanging(app, event)
49         ang = app.AngleSlider1.Value;
50         vel = event.Value;
51         vx = vel*cosd(ang);
52         vy = vel*sind(ang);
53         delete(app.hq)
54         app.hq = quiver(app.UIAxes,app.xs(1),app.ys(1),vx,vy,"Color","red");
55     end
56
57     % Value changing function: AngleSlider2
58     function AngleSlider2ValueChanging(app, event)
59         ang = event.Value; ←
60         vel = app.VelocitySlider2.Value; ↓
61         vx = -vel*cosd(ang); ↑
62         vy = vel*sind(ang); ↓
63         delete(app.hq)
64         app.hq = quiver(app.UIAxes,app.xs(2),app.ys(2),vx,vy,"Color","red");
65     end
66
67     % Value changing function: VelocitySlider2
68     function VelocitySlider2ValueChanging(app, event)
69
70
71
72
73
74
75     end
76 end
```

DESIGNER

CANVAS

VIEW



Save

Convert

FILE



ALIGN



Same Size

Grouping

Reorder

Evenly

Apply Horizontally

Apply Vertically



Run

Step

Stop

SPACE

RUN

MyAstroVolley.mlapp

Component Library

Search



COMMON



Axes



Button



Check Box



Date Picker



Drop Down



Edit Field (Numeric)



Edit Field (Text)



HTML



Hyperlink



Image



List Box



Radio Button



Slider



Spinner



State Button



Table



Text Area



Toggle Button

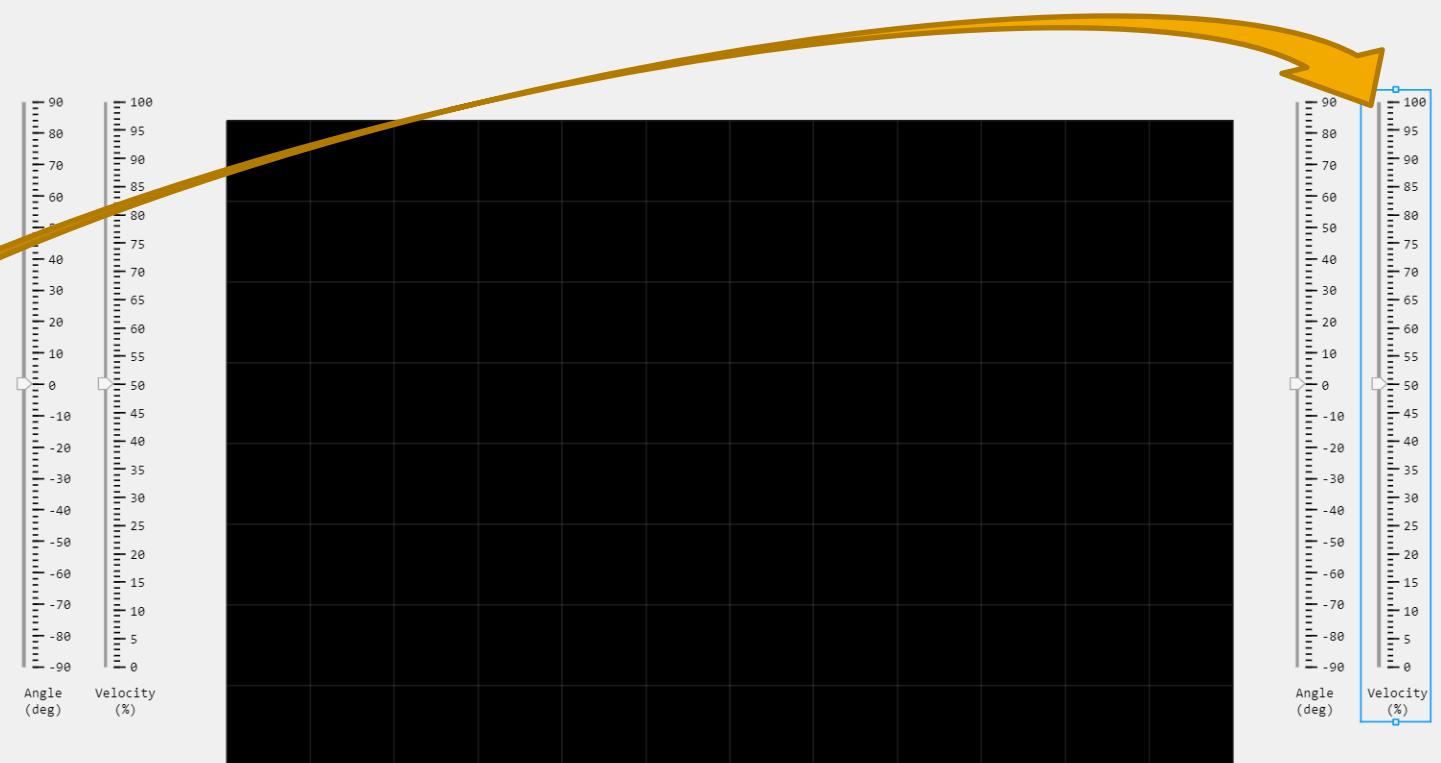


Tree



Tree (Check Box)

CONTAINERS



Design View

Code View

Component Browser

Search

MyAstroVolley

app.UIFigure

app.VelocitySlider2

app.AngleSlider2

app.VelocitySlider1

app.AngleSlider1

app.UIAxes

Slider | Callbacks

Search

Label Velocity,(%)

▼ SLIDER

Value

Limits

Orientation

▼ TICKS

MajorTicks 0,5,10,15,20,25,30,35,

MajorTickLabels 0,5,10,15,20,25,30,35,

MinorTicks 0,1,2,3,4,5,6,7,8,9,10,

▼ FONT

FontName Consolas

FontSize

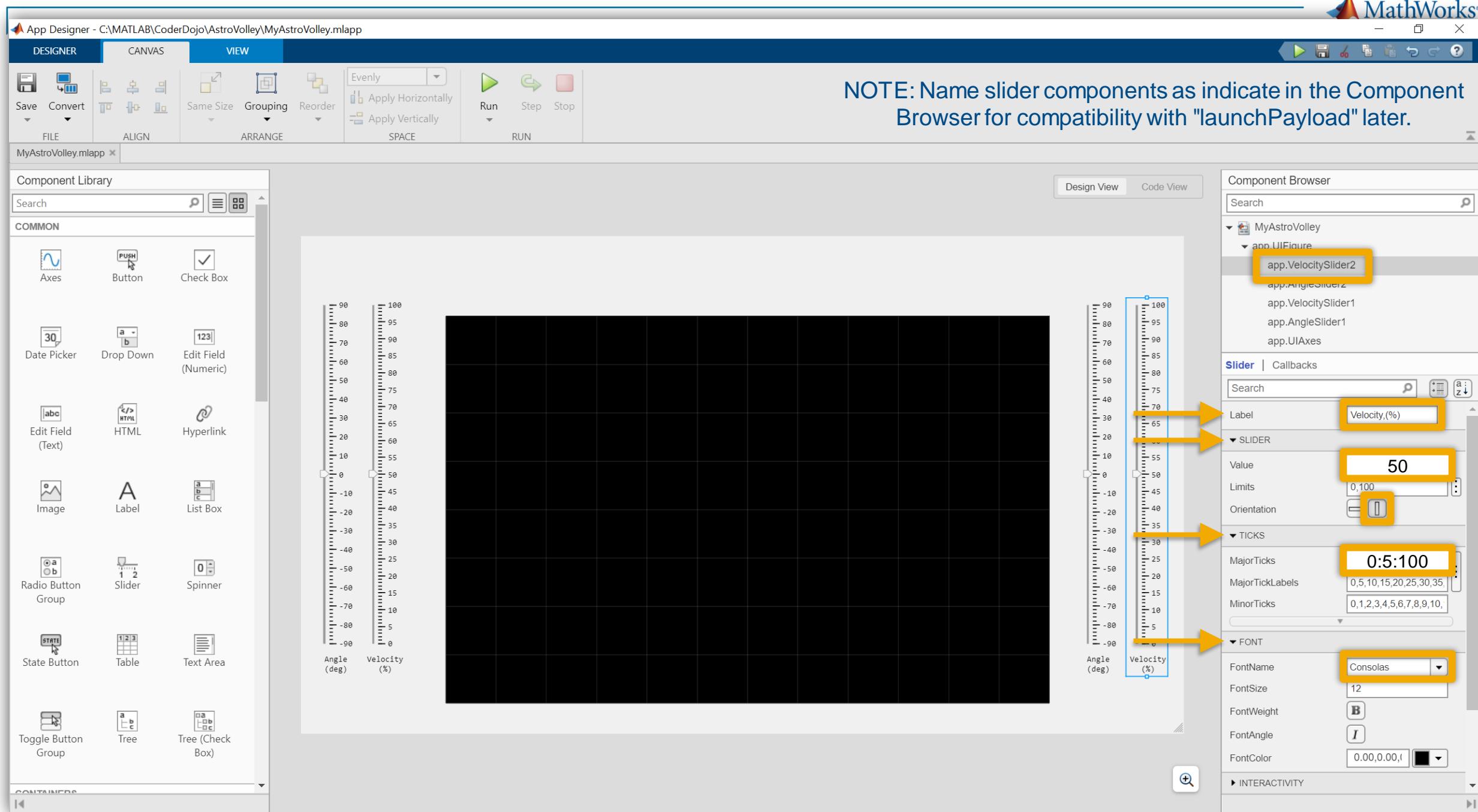
FontWeight

FontAngle

FontColor 0.00,0.00,1

► INTERACTIVITY

**NOTE:** Name slider components as indicated in the Component Browser for compatibility with "launchPayload" later.



DESIGNER

CANVAS

VIEW



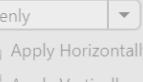
FILE



ALIGN



ARRANGE



SPACE



RUN

MyAstroVolley.mlapp

Component Library

Search



COMMON



Axes



Button



Check Box



Date Picker



Drop Down



Edit Field (Numeric)



Edit Field (Text)



HTML



Hyperlink



Image



Label



List Box



Radio Button Group



Slider



Spinner



State Button



Table



Text Area



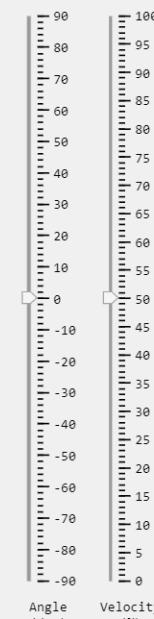
Toggle Button Group



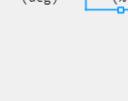
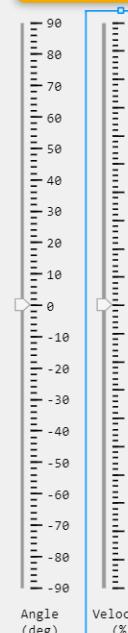
Tree



Tree (Check Box)



Right-Click



Component Browser

Search

MyAstroVolley  
app.UIFigure  
app.VelocitySlider2  
app.AngleSlider2

VelocitySlider1  
AngleSlider1  
IAxes  
Velocity, (%)

50  
0,100

100  
95  
90  
85  
80  
75  
70  
65  
60  
55  
50  
45  
40  
35  
30  
25  
20  
15  
10  
5  
0  
-5  
-10  
-15  
-20  
-25  
-30  
-35  
-40  
-45  
-50  
-55  
-60  
-65  
-70  
-75  
-80  
-85  
-90  
-95  
-100

Angle (deg)  
Velocity (%)

Cut Ctrl+X  
Copy Ctrl+C  
Paste Ctrl+V  
Duplicate Ctrl+D  
Delete

Zoom  
Align  
Same Size  
Grouping  
Reorder

Callbacks Add ValueChangedFcn callback  
Add ValueChangingFcn callback

Help on Selection Select existing callback...  
Replace With

Context Menu

FONT  
FontName Consolas  
FontSize 12  
FontWeight  
FontAngle  
FontColor 0.00,0.00,1

INTERACTIVITY

DESIGNER

EDITOR

VIEW



Save



Print



Compare



Go To



Bookmark



Callback



Function



Property



App Input Arguments



App Help Text



Comment



Indent



Run



Step



Stop

FILE

MyAstroVolley.mlapp

## Code Browser

Callbacks | Functions | Properties

Search

startupFcn

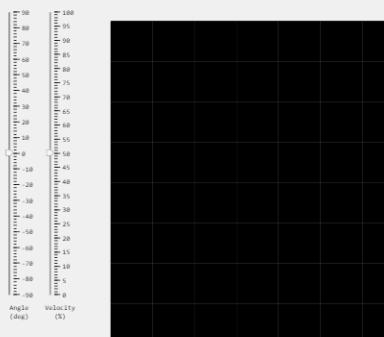
AngleSlider1ValueChanging

VelocitySlider1ValueChanging

AngleSlider2ValueChanging

VelocitySlider2ValueChanging

## App Layout



```
        vx = vel*cosd(ang);
        vy = vel*sind(ang);
        delete(app.hq)
        app.hq = quiver(app.UIAxes,app.xs(1),app.ys(1),vx,vy,"Color","red");
    end

    % Value changing function: VelocitySlider1
    function VelocitySlider1ValueChanging(app, event)
        ang = app.AngleSlider1.Value;
        vel = event.Value;
        vx = vel*cosd(ang);
        vy = vel*sind(ang);
        delete(app.hq)
        app.hq = quiver(app.UIAxes,app.xs(1),app.ys(1),vx,vy,"Color","red");
    end

    % Value changing function: AngleSlider2
    function AngleSlider2ValueChanging(app, event)
        ang = event.Value;
        vel = app.VelocitySlider2.Value;
        vx = -vel*cosd(ang);
        vy = vel*sind(ang);
        delete(app.hq)
        app.hq = quiver(app.UIAxes,app.xs(2),app.ys(2),vx,vy,"Color","red");
    end

    % Value changing function: VelocitySlider2
    function VelocitySlider2ValueChanging(app, event)
        ang = app.AngleSlider2.Value;
        vel = event.Value;
        vx = -vel*cosd(ang);
        vy = vel*sind(ang);
        delete(app.hq)
        app.hq = quiver(app.UIAxes,app.xs(2),app.ys(2),vx,vy,"Color","red");
    end
```

Design View Code View

## Component Browser

Search

- MyAstroVolley
  - app.UIFigure
  - app.VelocitySlider2
  - app.AngleSlider2
  - app.VelocitySlider1
  - app.AngleSlider1
  - app.UIAxes

## Slider | Callbacks

Search

SLIDER

Value	<input type="text" value="0"/>
Limits	<input type="text" value="-90,90"/>
Orientation	

TICKS

MajorTicks	<input type="text" value="-90,-80,-70,-60,-50,-40"/>
MajorTickLabels	<input type="text" value="-90,-80,-70,-60,-50,-40"/>
MinorTicks	<input type="text" value="-90,-88,-86,-84,-82,-80"/>

FONT

FontName	<input type="text" value="Consolas"/>
FontSize	<input type="text" value="12"/>
FontWeight	

DESIGNER

EDITOR

VIEW



Save



Print



Compare



Go To



Bookmark



Callback



Function



Property



App Input Arguments



App Help Text



Comment



Indent



Step



Stop

CODE

RUN



Run

MyAstroVolley.mlapp

## Code Browser

Callbacks | Functions | Properties

Search

startupFcn

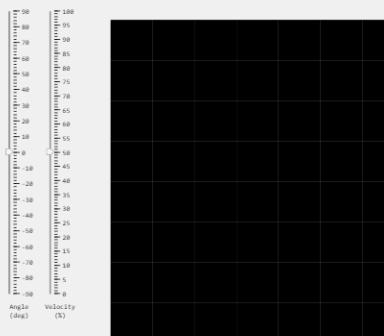
AngleSlider1ValueChanging

VelocitySlider1ValueChanging

AngleSlider2ValueChanging

VelocitySlider2ValueChanging

## App Layout



Design View Code View

## Component Browser

Search

MyAstroVolley

app.UIFigure

app.VelocitySlider2

app.AngleSlider2

app.VelocitySlider1

app.AngleSlider1

app.UIAxes

## Slider | Callbacks

Search

SLIDER

Value

0

Limits

-90,90

Orientation



TICKS

MajorTicks

-90,-80,-70,-60,-50,-40

MajorTickLabels

-90,-80,-70,-60,-50,-40

MinorTicks

-90,-88,-86,-84,-82,-80

FONT

FontName

Consolas

FontSize

12

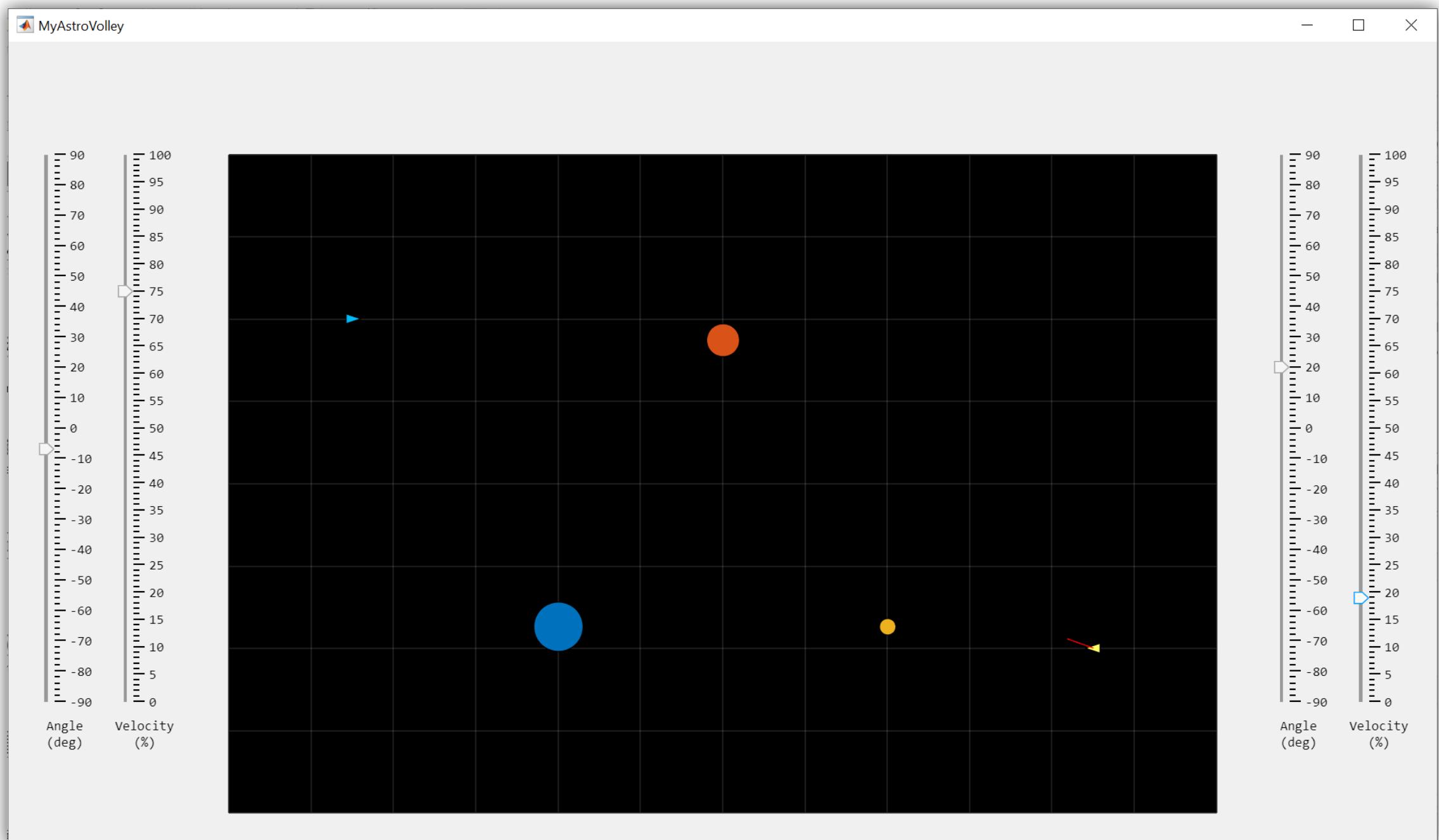
FontWeight



```

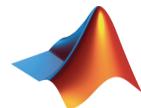
41         vx = vel*cosd(ang);
42         vy = vel*sind(ang);
43         delete(app.hq)
44         app.hq = quiver(app.UIAxes,app.xs(1),app.ys(1),vx,vy,"Color","red");
45     end
46
47     % Value changing function: VelocitySlider1
48     function VelocitySlider1ValueChanging(app, event)
49         ang = app.AngleSlider1.Value;
50         vel = event.Value;
51         vx = vel*cosd(ang);
52         vy = vel*sind(ang);
53         delete(app.hq)
54         app.hq = quiver(app.UIAxes,app.xs(1),app.ys(1),vx,vy,"Color","red");
55     end
56
57     % Value changing function: AngleSlider2
58     function AngleSlider2ValueChanging(app, event)
59         ang = event.Value;
60         vel = app.VelocitySlider2.Value;
61         vx = -vel*cosd(ang);
62         vy = vel*sind(ang);
63         delete(app.hq)
64         app.hq = quiver(app.UIAxes,app.xs(2),app.ys(2),vx,vy,"Color","red");
65     end
66
67     % Value changing function: VelocitySlider2
68     function VelocitySlider2ValueChanging(app, event)
69         ang = app.AngleSlider2.Value;
70         vel = event.Value;
71         vx = -vel*cosd(ang);
72         vy = vel*sind(ang);
73         delete(app.hq)
74         app.hq = quiver(app.UIAxes,app.xs(2),app.ys(2),vx,vy,"Color","red");
75     end
76 end

```



# Course Lesson Plan

- Installing and Playing AstroVolley
- Plotting Ships (Triangles)
- Plotting Grav. Bodies (Circles)
- Gravity and Orbits
- Launching Volleys
  - Creating a Mission Map Function
  - Plotting Velocity Vectors (quiver)
  - Animating Trajectories (comet)



## App Designer

- 1) UIAxes, Properties, & Functions
- 2) Sliders & Value Changing Callback
- 3) Push Button Callback
- 4) Generating Random Maps
- 5) Keeping Score

>> open MyAstroVolley2.mlapp



DESIGNER

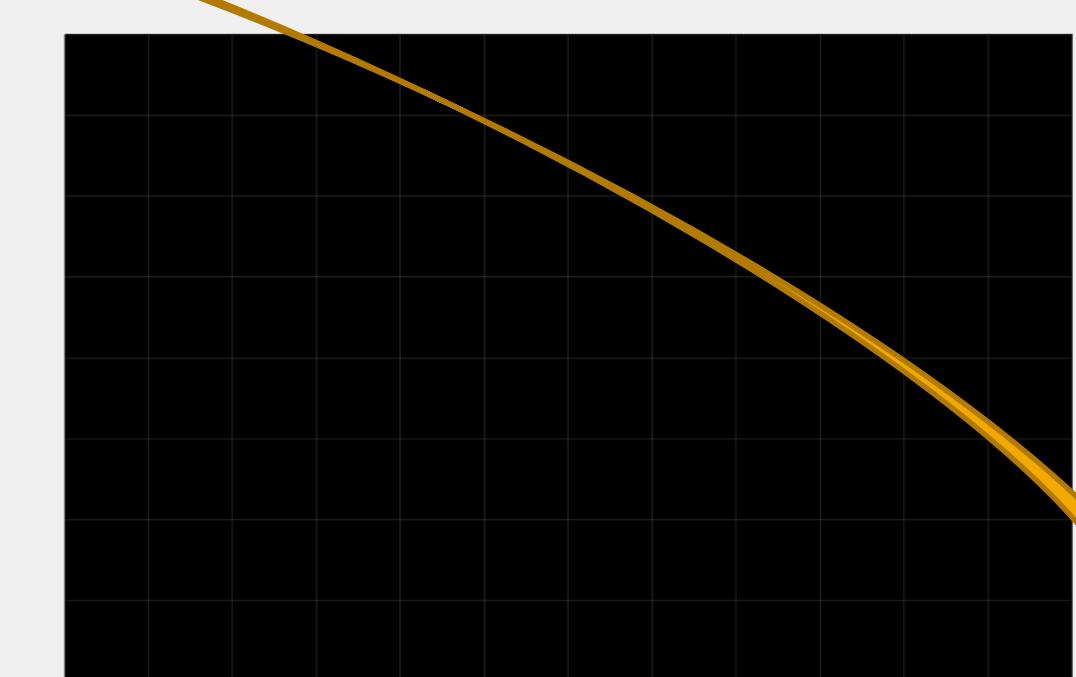
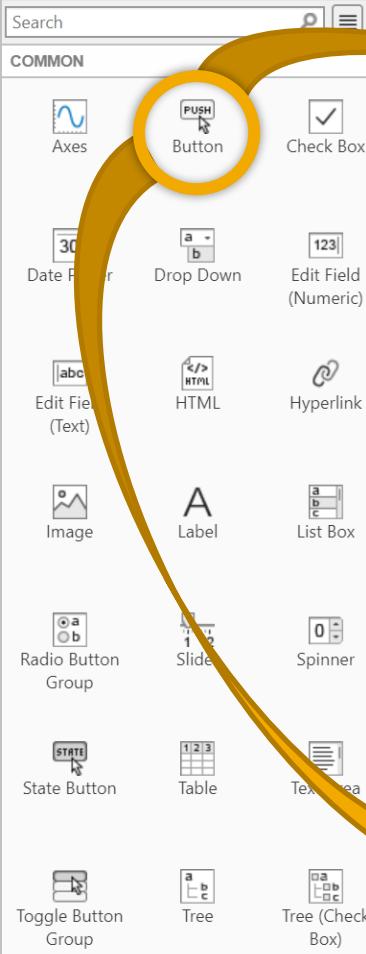
CANVAS

VIEW



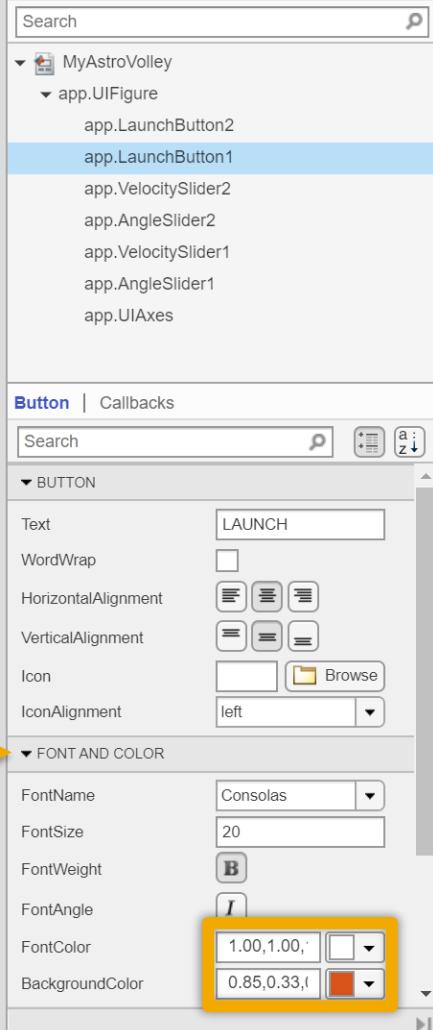
MyAstroVolley.mlapp

Component Library



Design View Code View

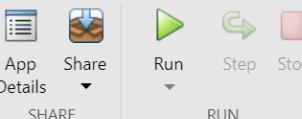
Component Browser



DESIGNER

CANVAS

VIEW



MyAstroVolley.mlapp

## Component Library

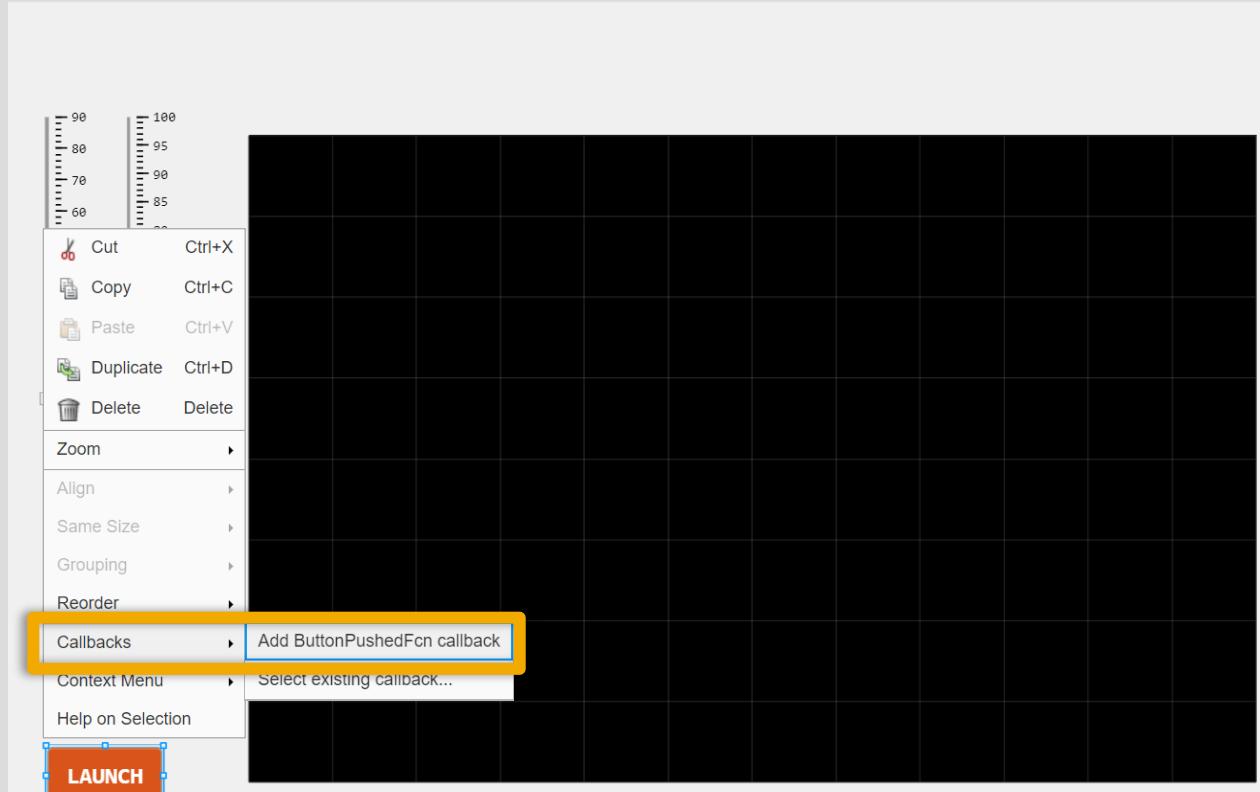
Search



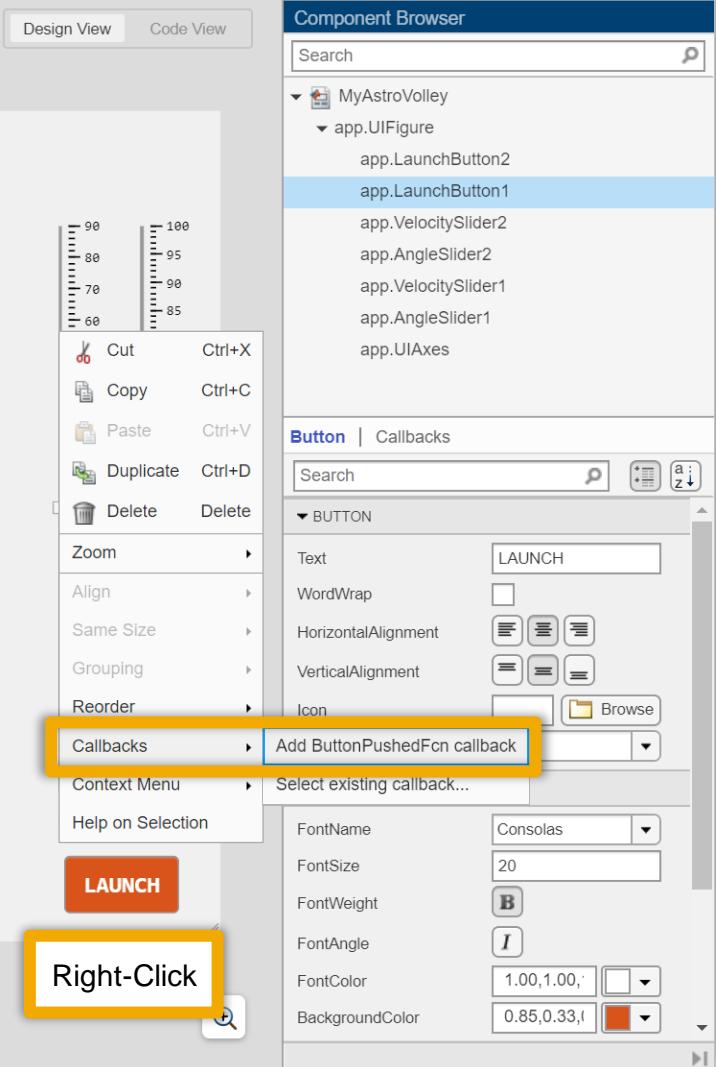
## COMMON



CONTAINERS



Right-Click



Right-Click

DESIGNER

EDITOR

VIEW

FILE

Compare

Save

Print

Go To

Bookmark

NAVIGATE

Callback

Function

Property

App Input

Arguments

Text

Comment

Indent

RUN

Step

Stop

CODE

RUN

NOTE: "launchPayload" requires the angle and velocity sliders be labelled as indicated in the Component Browser.

MyAstroVolley.mlapp

## Code Browser

Callbacks | Functions | Properties

Search

startupFcn

AngleSlider1ValueChanging

VelocitySlider1ValueChanging

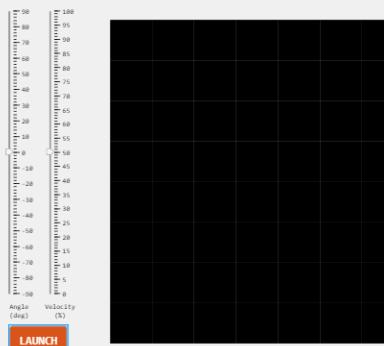
AngleSlider2ValueChanging

VelocitySlider2ValueChanging

LaunchButton1Pushed

LaunchButton2Pushed

## App Layout



```
% Value changing function: AngleSlider2
function AngleSlider2ValueChanging(app, event)
    ang = 180 - event.Value;
    vel = app.VelocitySlider2.Value;
    vx = vel*cosd(ang);
    vy = vel*sind(ang);
    delete(app.hq)
    app.hq = quiver(app.UIAxes,app.xs(2),app.ys(2),vx,vy,"Color","red");
end
```

```
% Value changing function: VelocitySlider2
function VelocitySlider2ValueChanging(app, event)
    ang = 180 - app.AngleSlider2.Value;
    vel = event.Value;
    vx = vel*cosd(ang);
    vy = vel*sind(ang);
    delete(app.hq)
    app.hq = quiver(app.UIAxes,app.xs(2),app.ys(2),vx,vy,"Color","red");
end
```

```
% Button pushed function: LaunchButton1
function LaunchButton1Pushed(app, event)
    [traj,iEvent] = launchPayload(app,1); % ship = 1
    comet(app.UIAxes,traj(:,1),traj(:,2))
end
```

```
% Button pushed function: LaunchButton2
function LaunchButton2Pushed(app, event)
    [traj,iEvent] = launchPayload(app,2); % ship = 2
    comet(app.UIAxes,traj(:,1),traj(:,2))
end
```

```
% Component initialization
methods (Access = private)
```

## Component Browser

Search

MyAstroVolley

app.UIFigure

app.LaunchButton2

app.LaunchButton1

app.VelocitySlider2

app.AngleSlider2

app.VelocitySlider1

app.AngleSlider1

app.UIAxes

## Button | Callbacks

Search

▼ BUTTON

Text

LAUNCH

WordWrap

HorizontalAlignment

VerticalAlignment

Icon

IconAlignment

▼ FONT AND COLOR

FontName

Consolas

FontSize

20

FontWeight

**B**

FontAngle

*I*

FontColor

1.00,1.00,

DESIGNER EDITOR VIEW

FILE NAVIGATE INSERT CODE RUN

**Run** (highlighted)

MyAstroVolley.mlapp

Code Browser

Callbacks | Functions | Properties

Search

startupFcn

AngleSlider1ValueChanging

VelocitySlider1ValueChanging

AngleSlider2ValueChanging

VelocitySlider2ValueChanging

LaunchButton1Pushed

LaunchButton2Pushed

App Layout

Component Browser

Search

- MyAstroVolley
  - app.UIFigure
    - app.LaunchButton2
    - app.LaunchButton1
    - app.VelocitySlider2
    - app.AngleSlider2
    - app.VelocitySlider1
    - app.AngleSlider1
    - app.UIAxes

Button | Callbacks

Search

BUTTON

Text: LAUNCH

WordWrap

HorizontalAlignment

VerticalAlignment

Icon

IconAlignment: left

FONT AND COLOR

FontName: Consolas

FontSize: 20

FontWeight: **B**

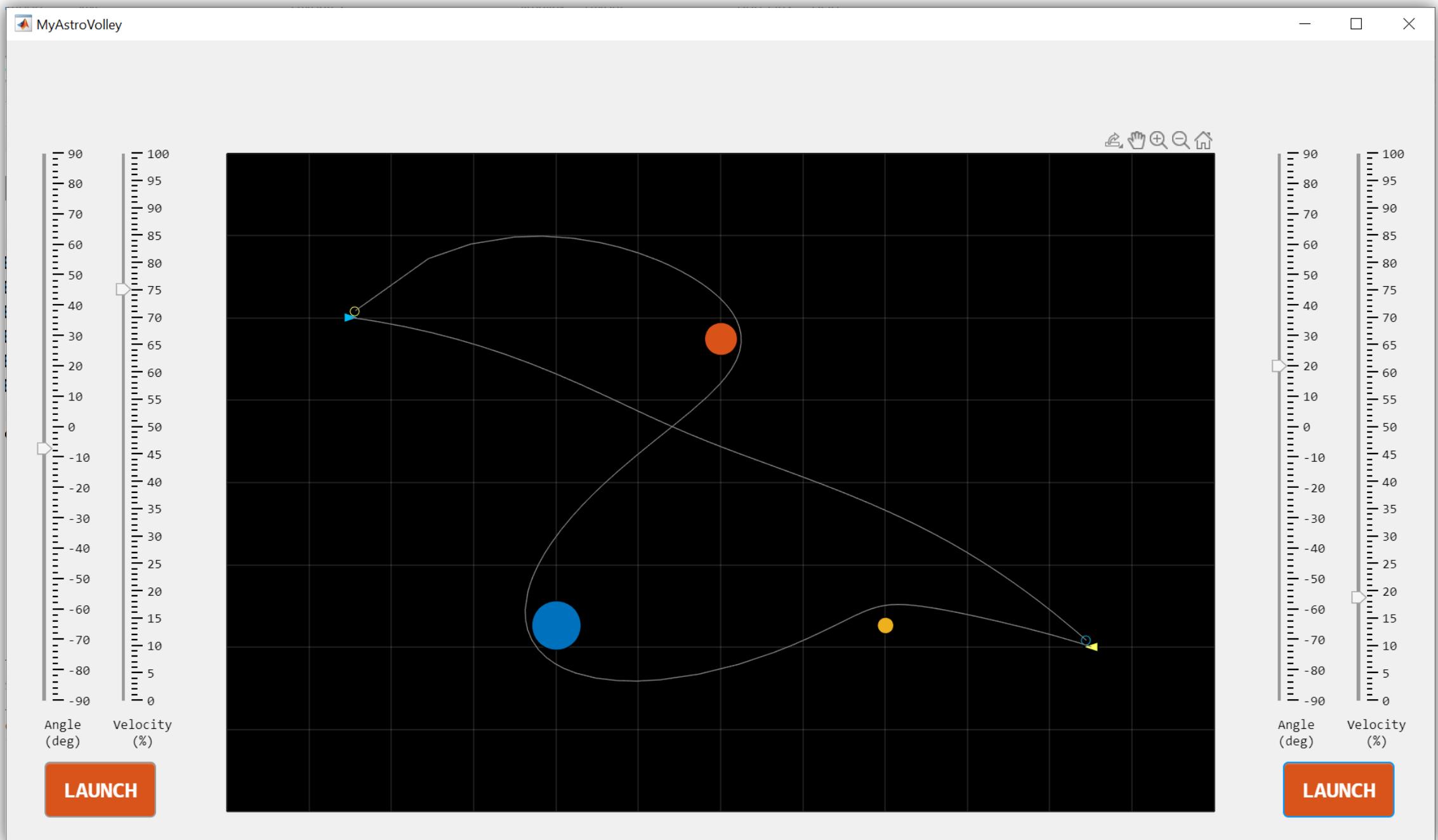
FontAngle: *I*

FontColor: 1.00,1.00,1.00

```

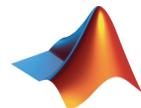
59 % Value changing function: AngleSlider2
60 function AngleSlider2ValueChanging(app, event)
61     ang = 180 - event.Value;
62     vel = app.VelocitySlider2.Value;
63     vx = vel*cosd(ang);
64     vy = vel*sind(ang);
65     delete(app.hq)
66     app.hq = quiver(app.UIAxes,app.xs(2),app.ys(2),vx,vy,"color","red");
67 end
68
69 % Value changing function: VelocitySlider2
70 function VelocitySlider2ValueChanging(app, event)
71     ang = 180 - app.AngleSlider2.Value;
72     vel = event.Value;
73     vx = vel*cosd(ang);
74     vy = vel*sind(ang);
75     delete(app.hq)
76     app.hq = quiver(app.UIAxes,app.xs(2),app.ys(2),vx,vy,"color","red");
77 end
78
79 % Button pushed function: LaunchButton1
80 function LaunchButton1Pushed(app, event)
81     [traj,iEvent] = launchPayload(app,1); % ship = 1
82     comet(app.UIAxes,traj(:,1),traj(:,2))
83 end
84
85 % Button pushed function: LaunchButton2
86 function LaunchButton2Pushed(app, event)
87     [traj,iEvent] = launchPayload(app,2); % ship = 2
88     comet(app.UIAxes,traj(:,1),traj(:,2))
89 end
90 end
91
92 % Component initialization
93 methods (Access = private)
94

```



# Course Lesson Plan

- Installing and Playing AstroVolley
- Plotting Ships (Triangles)
- Plotting Grav. Bodies (Circles)
- Gravity and Orbits
- Launching Volleys
  - Creating a Mission Map Function
  - Plotting Velocity Vectors (quiver)
  - Animating Trajectories (comet)



## App Designer

- 1) UIAxes, Properties, & Functions
- 2) Sliders & Value Changing Callback
- 3) Push Button Callback
- 4) Generating Random Maps
- 5) Keeping Score

```
>> open MyAstroVolley3.mlapp
```

# Creating a Random Mission Map

- First, use the command window (or a new live script) to show examples of how to create random numbers

`rand(M,N)` returns an M-by-N matrix of random numbers between 0 and 1

```
>> rand(1,5)
```

`randi(IMAX)` returns a single random integer between 1 and IMAX

```
>> randi(99)
```

```
>> open mapMission.m
```

# Update mapMission function to create random maps and try calling it

```

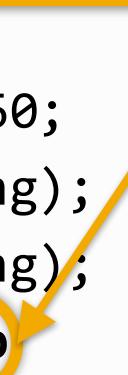
function app = mapMission
ax = gca; % get axes
cla(ax) % clear axes
app.xs = [-225 225]; % ship x-pos
app.ys = 300*rand(1,2) - 150; % y-pos
app.Rs = [ 5 5]; % ship sizes
cs = [0 .73 .96; 1 1 .4]; % ship colors
xt = [-5 4 4 -5]/5; % triangle x-points
yt = [ 0 3 -3 0]/5; % triangle y-points
fill(ax,app.xs(1)-app.Rs(1)*xt, ...
      app.ys(1)+app.Rs(1)*yt,cs(1,:))
hold(ax,"on")
fill(ax,app.xs(2)+app.Rs(2)*xt, ...
      app.ys(2)+app.Rs(2)*yt,cs(2,:))

```

```

nb = randi(5); % # of bodies
app.xb = 300*rand(1,nb) - 150; % x-pos
app.yb = 200*rand(1,nb) - 100; % y-pos
app.Rb = 5 + 10*rand(1,nb); % body sizes
cb = lines(nb); % body colors
ang = 0:5:360; % angle, degrees
xc = cosd(ang); % circle x-points
yc = sind(ang); % circle y-points
for k = 1:nb
    fill(ax,app.xb(k)+app.Rb(k)*xc, ...
          app.yb(k)+app.Rb(k)*yc,cb(k,:))
end
axis(ax,"equal")
xlim(ax,[-300 300]); ylim(ax,[-200 200])
end % function

```



DESIGNER

CANVAS

VIEW

 New  
 Open  
 Save  
 Compare App Details  
 Share Run  
 Step  
 Stop

FILE

SHARE

RUN

MyAstroVolley.mlapp x

Component Library

Search



COMMON



Axes

PUSH



Button



Check Box



Date Picker

30



Drop Down



123

Edit Field (Numeric)



abc



HTML



Hyperlink



Image



A



List Box



a b



1 2



0

Group



STATE



1 2 3



Text Area



a b



Tree



Tree (Check Box)

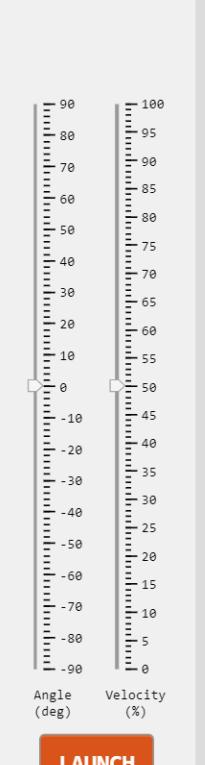
CONTAINERS



New Mission



Design View Code View



Component Browser

Search

MyAstroVolley

app.UIFigure

app.NewMissionButton

app.LaunchButton2

app.LaunchButton1

app.VelocitySlider2

app.AngleSlider2

app.VelocitySlider1

app.AngleSlider1

app.UIAxes

Button | Callbacks

Search

▼ BUTTON

Text

WordWrap

HorizontalAlignment

VerticalAlignment

Icon

IconAlignment

FontName

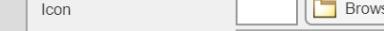
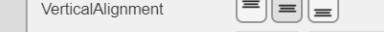
FontSize

FontWeight

FontAngle

FontColor

New,Mission



left

FontName

FontSize

FontWeight

FontAngle

FontColor

0.00,0.00,1



DESIGNER

CANVAS

VIEW



FILE

SHARE

RUN

RUN

MyAstroVolley.mlapp

Component Library

Search



COMMON



Axes



Button



Check Box



Date Picker



Drop Down



Edit Field (Numeric)



Edit Field (Text)



HTML



Hyperlink



Image



Label



List Box



Radio Button Group



Slider



Spinner



State Button



Table



Text Area



Toggle Button Group



Tree



Tree (Check Box)

CONTAINERS

Right-Click

New Mission



Cut

Ctrl+X



Copy

Ctrl+C



Paste

Ctrl+V



Duplicate

Ctrl+D



Delete

Delete



Zoom

&gt;



Align

&gt;



Same Size

&gt;



Grouping

&gt;



Reorder

&gt;

Callbacks &gt; Add ButtonPushedFcn callback



Context Menu

&gt;

Select existing callback...



Help on Selection

&gt;



Angle (deg)

-70



Velocity (%)

10

-

-80

-

-90

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

The screenshot shows the MATLAB App Designer Editor interface. The top menu bar includes DESIGNER, EDITOR, and VIEW. The toolbar contains various icons for file operations (Save, Print, Go To, etc.) and code navigation (Comment, Indent, Step, Stop). A yellow box highlights the 'Run' button in the toolbar.

The main workspace displays the MATLAB code for the application. A callout bubble highlights the `NewMissionButtonPushed` function:

```

19
20
21 properties (Access = public)
22     xs % ship x-positions
23     ys % ship y-positions
24     Rs % ship size
25     xb % body x-positions
26     yb % body y-positions
27     Rb % body sizes
28     hq % quiver handle
29 end
30
31
32 % Callbacks that handle component events
33 methods (Access = private)
34
35 % Code that executes after component creation
36 function startupFcn(app)
37     fixedMission(app);
38 end
39
40 % Button pushed function: NewMissionButton
41 function NewMissionButtonPushed(app, event)
42     randomMission(app);
43 end
44
45 % Value changing function: AngleSlider1
46 function AngleSlider1ValueChanging(app, event)
47     ang = event.Value;
48     vel = app.VelocitySlider1.Value;
49     vx = vel*cosd(ang);
50     vy = vel*sind(ang);
51     delete(app.hq)
52     app.hq = quiver(app.UIAxes, app.xs(1), app.ys(1), vx, vy, "Color", "red");
53 end
54

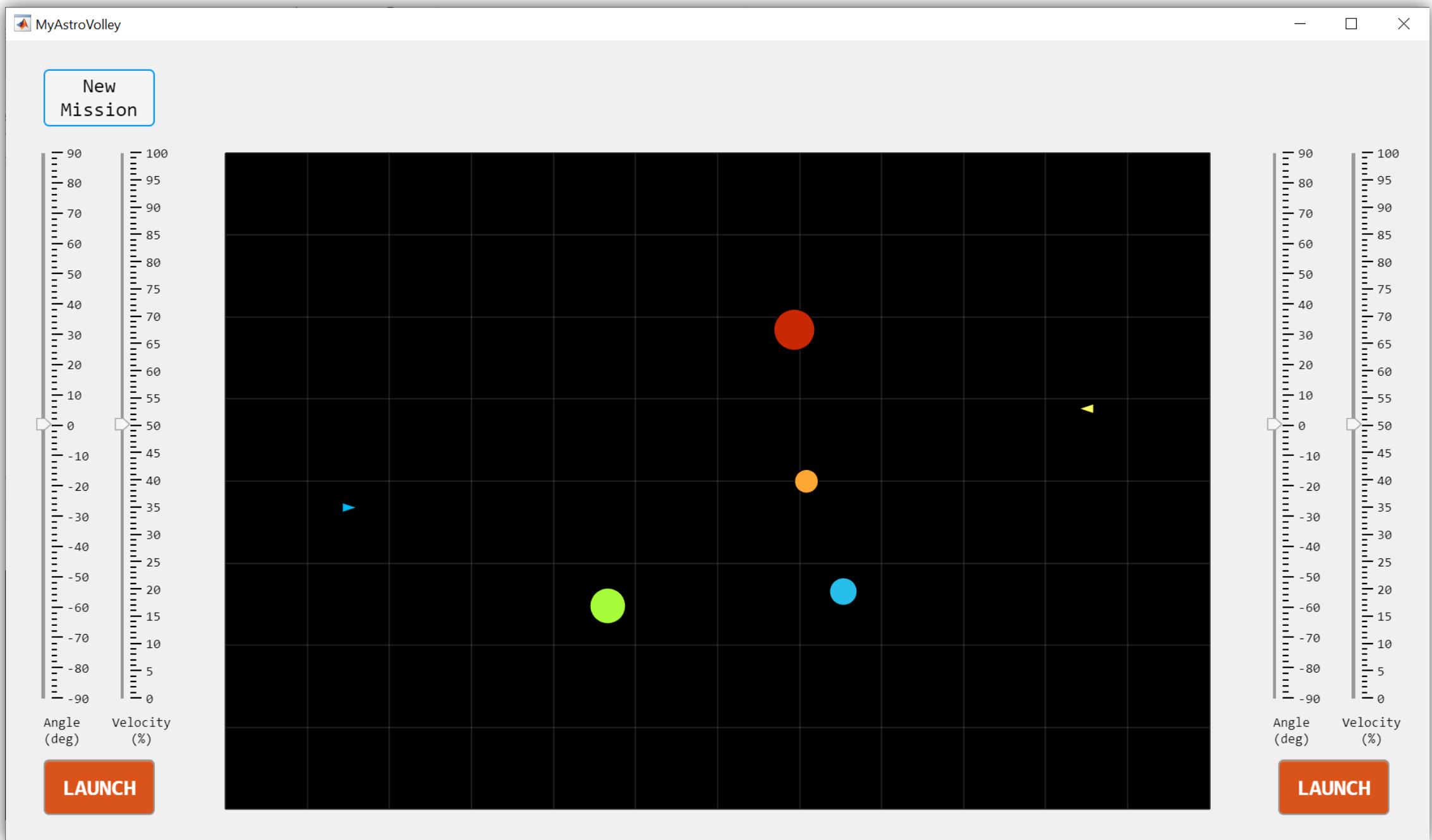
```

The left sidebar shows the 'Code Browser' with 'Callbacks', 'Functions', and 'Properties' sections. The 'App Layout' section shows a preview of the application window with a grid and two buttons: 'New Mission' and 'LAUNCH'.

The right sidebar shows the 'Component Browser' with a tree view of components under 'MyAstroVolley' and 'app.UIFigure'. It also includes 'Button' and 'Callbacks' settings for the 'NewMissionButton' component.

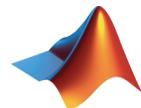
A callout bubble points to the `NewMissionButtonPushed` function in the code editor, stating:

The `randomMission` function is basically a clean version of our `mapMission` function that is also compatible with our App Designer "UIAxes" stored in the "app" input variable.



# Course Lesson Plan

- Installing and Playing AstroVolley
- Plotting Ships (Triangles)
- Plotting Grav. Bodies (Circles)
- Gravity and Orbits
- Launching Volleys
  - Creating a Mission Map Function
  - Plotting Velocity Vectors (quiver)
  - Animating Trajectories (comet)



## App Designer

- 1) UIAxes, Properties, & Functions
- 2) Sliders & Value Changing Callback
- 3) Push Button Callback
- 4) Generating Random Maps
- 5) Keeping Score

>> open MyAstroVolley4.mlapp

DESIGNER

CANVAS

VIEW



Save Convert



ALIGN



Same Size

Grouping

Reorder



Evenly

Apply Horizontally

Apply Vertically

SPACE

Run

Step

Stop

RUN

MyAstroVolley.mlapp

Component Library

Search



COMMON



Axes



Push Button



Check Box



Drop Down



Edit Field (Numeric)



HTML



Hyperlink



Edit Field (Text)



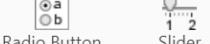
Image



Label



List Box



Radio Button Group



Slider



Spinner



State Button



Table



Text Area



Toggle Button Group



Tree



Tree (Check Box)

CONTAINERS



DESIGNER

CANVAS

VIEW



ALIGN

ARRANGE

SPACE

RUN

MyAstroVolley.mlapp

Component Library

Search



COMMON



Axes



Button



Check Box



30



a - b



23



Edit Field (Numeric)



abc



ML



@



Hyperlink



Image



A



Label



a b



1 2



0



Spinner



STATE



1 2 3



Table



Text



Area



a b c



Tree

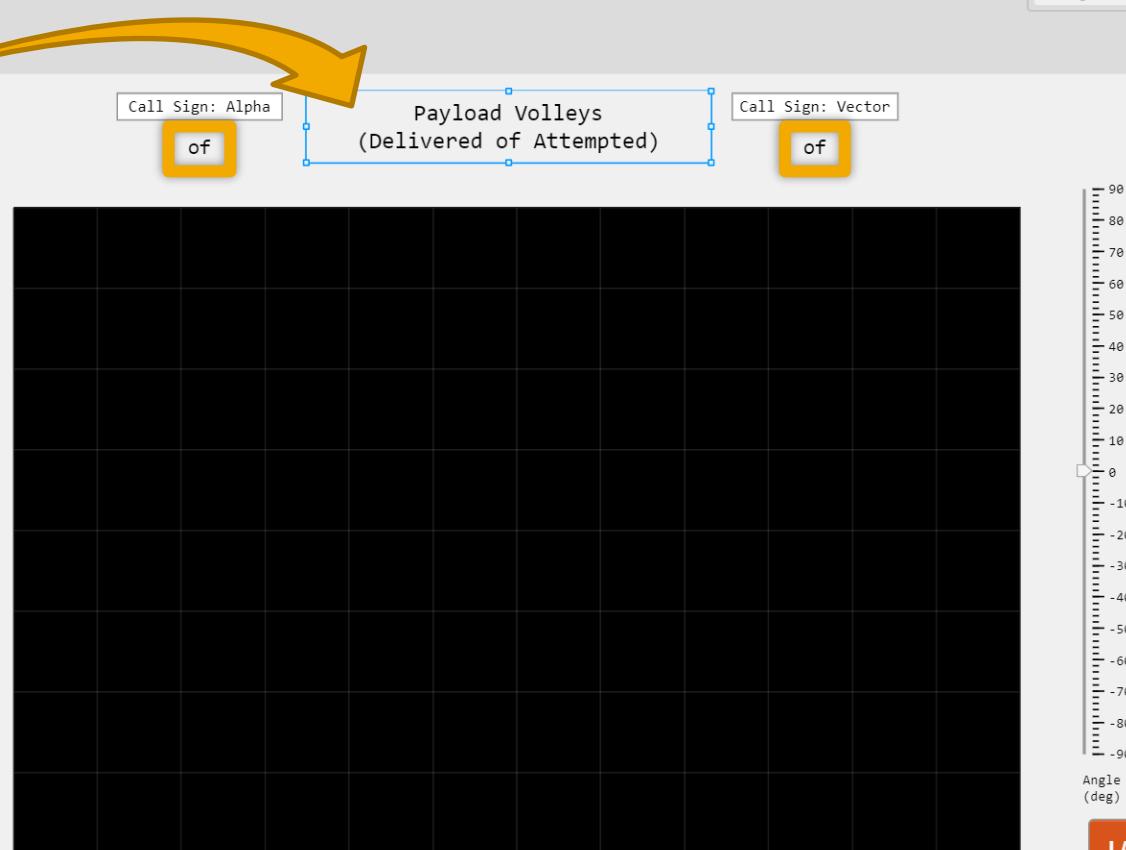


a b c



Tree (Check Box)

CONTAINERS



Design View Code View

Component Browser

Search

MyAstroVolley

app.UIFigure

app.VolleysLabel

app.ofLabel2

app.ofLabel1

app.CallSignField2

app.CallSignField1

app.NewMissionButton

app.LaunchButton2

app.LaunchButton1

app.VelocitySlider2

app.AngleSlider2

Label | Callbacks

Search

TEXT

Text Interpreter WordWrap HorizontalAlignment VerticalAlignment

FontName FontSize FontWeight FontAngle FontColor BackgroundColor

**NOTE: Name components as indicated in the Component Browser for consistency with code later.**

The screenshot shows the MATLAB App Designer interface with the following components and annotations:

- Component Library (Left):** Shows various UI components like Axes, Button, Check Box, Date Picker, Drop Down, Edit Field (Text), HTML, Hyperlink, Image, Label, List Box, Radio Button Group, Slider, Spinner, State Button, Table, Text Area, Toggle Button Group, Tree, and Tree (Check Box).
- Canvas (Center):** Displays a mission control interface with a grid background. It includes sections for "New Mission" (Call Sign: Alpha, Edit Field (Numeric)), "Payload Volleys (Delivered of Attempted)" (Call Sign: Vector, Edit Field (Numeric)), and a "LAUNCH" button.
- Component Browser (Right):** Shows the component hierarchy and properties for "MyAstroVolley". The "Edit Field (Numeric)" component is highlighted with a blue selection bar. Its properties are displayed in the right panel, including:
  - Value:** 0
  - Limits:** 0, Inf
  - RoundFractionalValues:** unchecked
  - ValueDisplayFormat:** %11.4g
  - HorizontalAlignment:** Left
- Interactivity Properties:** The "Enable" checkbox is checked (indicated by a yellow circle).

DESIGNER

CANVAS

VIEW



Save



Convert



ALIGN



Same Size



Grouping Reorder



Evenly

Apply Horizontally



Apply Vertically



Run

Step



Stop



SPACE

MyAstroVolley.mlapp

Design View

Code View

Component Library

Search

COMMON



Axes



Button



Check Box



Date Picker



Drop Down



Edit Field (Numeric)



Edit Field (Text)



HTML



Hyperlink



Image



Label



List Box



Radio Button Group



Slider



Spinner



State Button



Table



Text Area



Toggle Button Group



Tree



Tree (Check Box)

CONTAINERS



Component Browser

Search

MyAstroVolley

app.UIFigure

app.VolleysLabel

app.ofLabel2

app.ofLabel1

app.CallSignField2

app.CallSignField1

app.ClearScoresButton

app.DeliveredField2

app.AttemptedField2

app.DeliveredField1

app.AttemptedField1

Button | Callbacks

Search

▼ BUTTON

Text



WordWrap

HorizontalAlignment



VerticalAlignment

Icon



IconAlignment



left

▼ FONT AND COLOR

FontName



FontSize



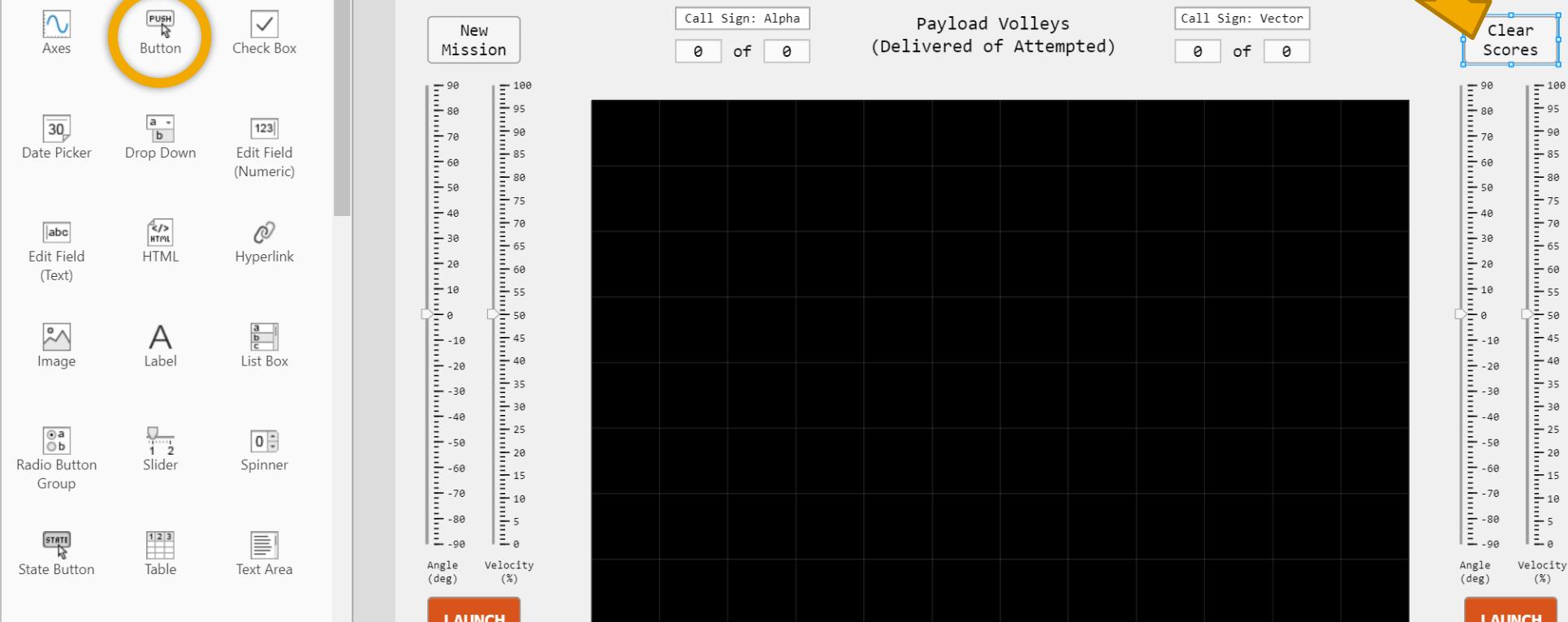
FontWeight



FontAngle



FontColor





**NOTE: Make sure component names used in the code match those in the Component Browser.**

The screenshot shows the MATLAB App Designer interface with the following components:

- DESIGNER** tab selected in the top menu bar.
- FILE** menu open, showing options like Save, Print, Go To, and NAVIGATE.
- EDITOR** tab in the top menu bar.
- VIEW** tab in the top menu bar.
- INSERT** toolbar with icons for Callback, Function, Property, App Input Arguments, App Help Text, Comment, Indent, Run, Step, and Stop.
- CODE** toolbar with icons for Comment, Indent, Run, Step, and Stop.
- Code Browser** pane on the left showing callbacks, functions, and properties. It includes a search bar and a list of events like startupFcn, NewMissionButtonPushed, AngleSlider1ValueChanging, VelocitySlider1ValueChanging, AngleSlider2ValueChanging, VelocitySlider2ValueChanging, LaunchButton1Pushed, LaunchButton2Pushed, and ClearScoresButtonPushed.
- Component Browser** pane on the right showing the component tree under MyAstroVolley. A yellow box highlights the app.DeliveredField2 component. Below it, another yellow box highlights the function definition for ClearScoresButtonPushed.
- Design View** and **Code View** tabs at the top of the main workspace.
- Main Workspace** containing the code editor with MATLAB code for button pushed functions and component initialization.

```

% Button pushed function: LaunchButton1
function LaunchButton1Pushed(app, event)
    [traj,iEvent] = launchPayload(app,1); % ship = 1
    comet(app.UIAxes,traj(:,1),traj(:,2))

    app.AttemptedField1.Value = app.AttemptedField1.Value + 1;
    if (iEvent == 2) % event: reached ship 2
        app.DeliveredField1.Value = app.DeliveredField1.Value + 1;
    end

end

% Button pushed function: LaunchButton2
function LaunchButton2Pushed(app, event)
    [traj,iEvent] = launchPayload(app,2); % ship = 2
    comet(app.UIAxes,traj(:,1),traj(:,2))

    app.AttemptedField2.Value = app.AttemptedField2.Value + 1;
    if (iEvent == 1) % event: reached ship 1
        app.DeliveredField2.Value = app.DeliveredField2.Value + 1;
    end

end

% Button pushed function: ClearScoresButton
function ClearScoresButtonPushed(app, event)
    app.AttemptedField1.Value = 0;
    app.DeliveredField1.Value = 0;
    app.AttemptedField2.Value = 0;
    app.DeliveredField2.Value = 0;

end

% Component initialization
methods (Access = private)

% Create UIFigure and components
function createComponents(app)

```

**NOTE: Make sure component names used in the code match those in the Component Browser.**

The screenshot shows the MATLAB App Designer interface. The top menu bar includes DESIGNER, EDITOR, and VIEW. The toolbar contains various icons for file operations (Save, Print, Compare, Go To, Bookmark) and code navigation (Find, Insert, Comment, Indent, Run, Step, Stop). The main workspace displays the MATLAB code for the application. Two sections of code are highlighted with orange boxes:

```

% Button pushed function: LaunchButton1
function LaunchButton1Pushed(app, event)
    [traj,iEvent] = launchPayload(app,1); % ship = 1
    comet(app.UIAxes,traj(:,1),traj(:,2))

    app.AttemptedField1.Value = app.AttemptedField1.Value + 1;
    if (iEvent == 2) % event: reached ship 2
        app.DeliveredField1.Value = app.DeliveredField1.Value + 1;
    end

% Button pushed function: LaunchButton2
function LaunchButton2Pushed(app, event)
    [traj,iEvent] = launchPayload(app,2); % ship = 2
    comet(app.UIAxes,traj(:,1),traj(:,2))

    app.AttemptedField2.Value = app.AttemptedField2.Value + 1;
    if (iEvent == 1) % event: reached ship 1
        app.DeliveredField2.Value = app.DeliveredField2.Value + 1;
    end

% Button pushed function: ClearScoresButton
function ClearScoresButtonPushed(app, event)
    app.AttemptedField1.Value = 0;
    app.DeliveredField1.Value = 0;
    app.AttemptedField2.Value = 0;
    app.DeliveredField2.Value = 0;

    end
end

% Component initialization
methods (Access = private)

% Create UIFigure and components
function createComponents(app)

```

The left sidebar shows the "Code Browser" with sections for Callbacks, Functions, and Properties, and a search bar. The "App Layout" section shows the user interface with a plot area, score fields, and buttons. The right sidebar is the "Component Browser" which lists components like MyAstroVolley, app.UIFigure, app.VolleysLabel, etc., with "DeliveredField2" selected. The bottom right corner of the component browser has a preview panel showing the button's appearance with the text "Clear Scores".

**DESIGNER**    **EDITOR**    **VIEW**

**FILE**    **NAVIGATE**

**INSERT**

**CODE**

**RUN**

**Comment %**    **Indent**    **Step**    **Stop**

**Run**

**Design View**    **Code View**

**Component Browser**

**Search**

- MyAstroVolley
  - app.UIFigure
  - app.VolleyesLabel
  - app.ofLabel2
  - app.ofLabel1
  - app.CallSignField2
  - app.CallSignField1
  - app.ClearScoresButton
  - app.DeliveredField2
  - app.AttemptedField2
  - app.DeliveredField1
  - app.AttemptedField1

**Button | Callbacks**

**Search**

**BUTTON**

**Text**: Clear.Scores

**WordWrap**:

**HorizontalAlignment**:

**VerticalAlignment**:

**Icon**: **Browse**

**IconAlignment**: left

**FONT AND COLOR**

**FontName**: Consolas

**FontSize**: 18

**FontWeight**: **B**

**FontAngle**: **I**

**FontColor**: 0.00,0.00,1

**Code Browser**

**Callbacks | Functions | Properties**

**Search**

startupFcn  
NewMissionButtonPushed  
AngleSlider1ValueChanging  
VelocitySlider1ValueChanging  
AngleSlider2ValueChanging  
VelocitySlider2ValueChanging  
LaunchButton1Pushed  
LaunchButton2Pushed  
ClearScoresButtonPushed

**App Layout**

New Mission Call Sign: Alpha Payload Volleys (Delivered of Attempted)

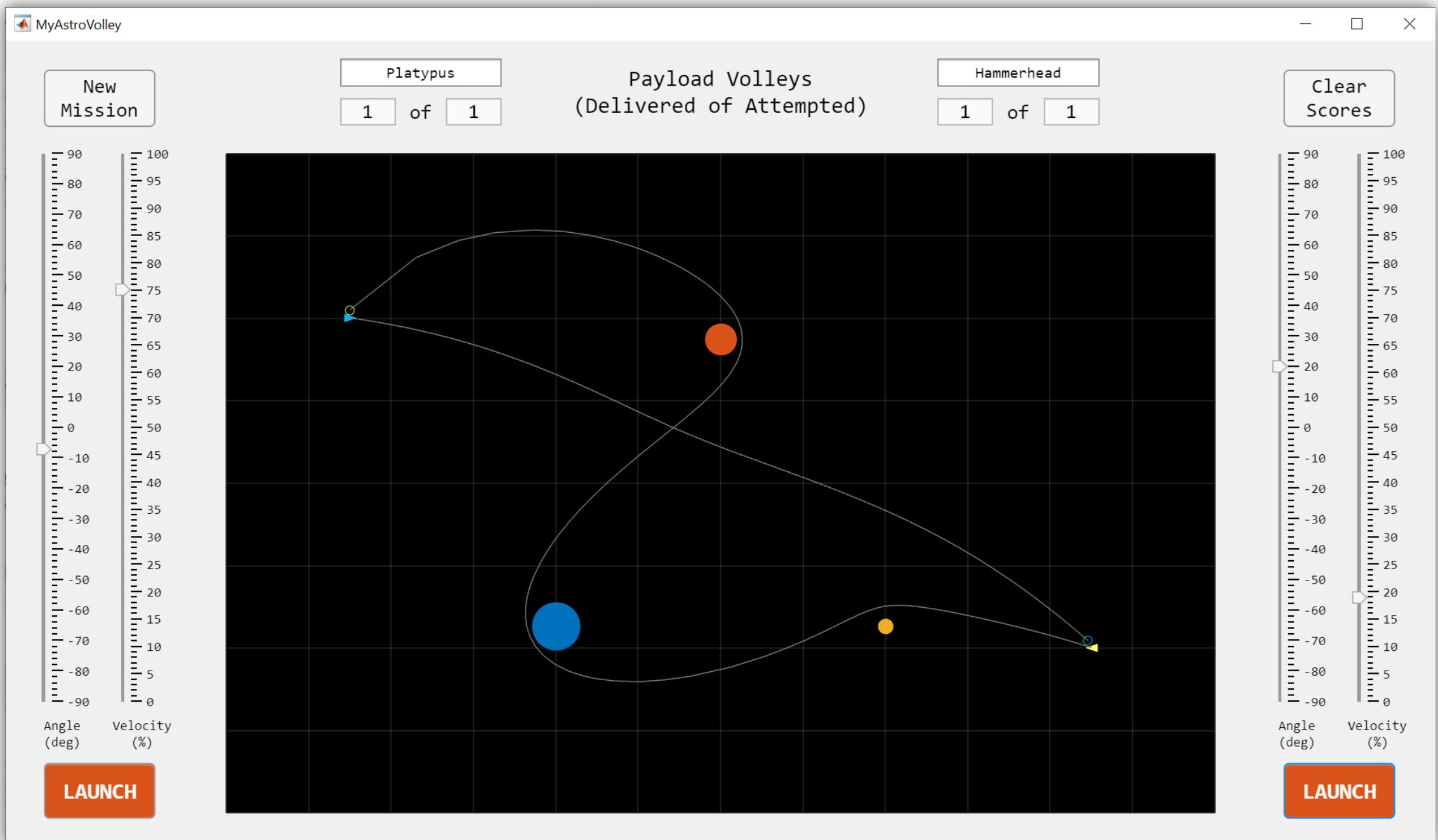
Angle (deg) Velocity (m/s)

LAUNCH

```

95 % Button pushed function: LaunchButton1
96 function LaunchButton1Pushed(app, event)
97     [traj,iEvent] = launchPayload(app,1); % ship = 1
98     comet(app.UIAxes,traj(:,1),traj(:,2))
99
100    app.AttemptedField1.Value = app.AttemptedField1.Value + 1;
101    if (iEvent == 2) % event: reached ship 2
102        app.DeliveredField1.Value = app.DeliveredField1.Value + 1;
103    end
104
105    % Button pushed function: LaunchButton2
106    function LaunchButton2Pushed(app, event)
107        [traj,iEvent] = launchPayload(app,2); % ship = 2
108        comet(app.UIAxes,traj(:,1),traj(:,2))
109
110        app.AttemptedField2.Value = app.AttemptedField2.Value + 1;
111        if (iEvent == 1) % event: reached ship 1
112            app.DeliveredField2.Value = app.DeliveredField2.Value + 1;
113        end
114
115    % Button pushed function: ClearScoresButton
116    function ClearScoresButtonPushed(app, event)
117        app.AttemptedField1.Value = 0;
118        app.DeliveredField1.Value = 0;
119        app.AttemptedField2.Value = 0;
120        app.DeliveredField2.Value = 0;
121
122        end
123    end
124
125    % Component initialization
126    methods (Access = private)
127
128        % Create UIFigure and components
129        function createComponents(app)
130
131

```



# Excellent work everyone!

Thank you for being a part of  
**MATLAB AstroVolley!**