

AstroVolley: Plotting & App Building

Instructor Guide

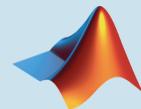


An astrodynamics game where navigators attempt to deliver payloads between star cruisers by volleying them through fantastical gravity fields.

Course Lesson Plan

- Installing and Playing AstroVolley
 - Plotting Ships (Triangles)
 - Plotting Grav. Bodies (Circles)
 - Gravity and Orbits
 - Launching Volleys
 - Creating a Random Mission Map
 - 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 Map
 - 5) Keeping Score

Course Lesson Plan



Installing and Playing AstroVolley

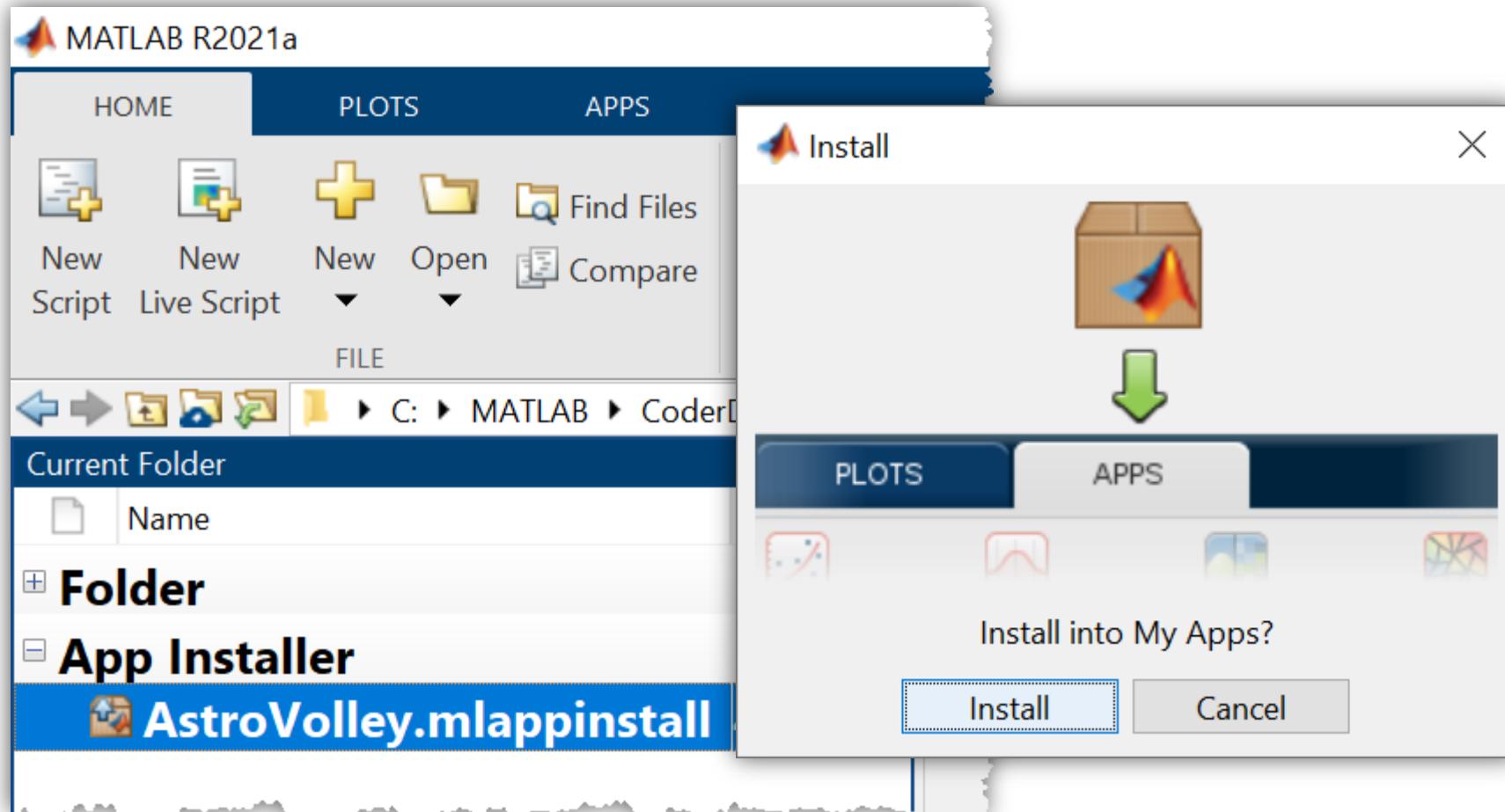
- Plotting Ships (Triangles)
- Plotting Grav. Bodies (Circles)
- Gravity and Orbits
- Launching Volleys
 - Creating a Random Mission Map
 - Plotting Velocity Vectors (quiver)
 - Animating Trajectories (comet)

App Designer

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

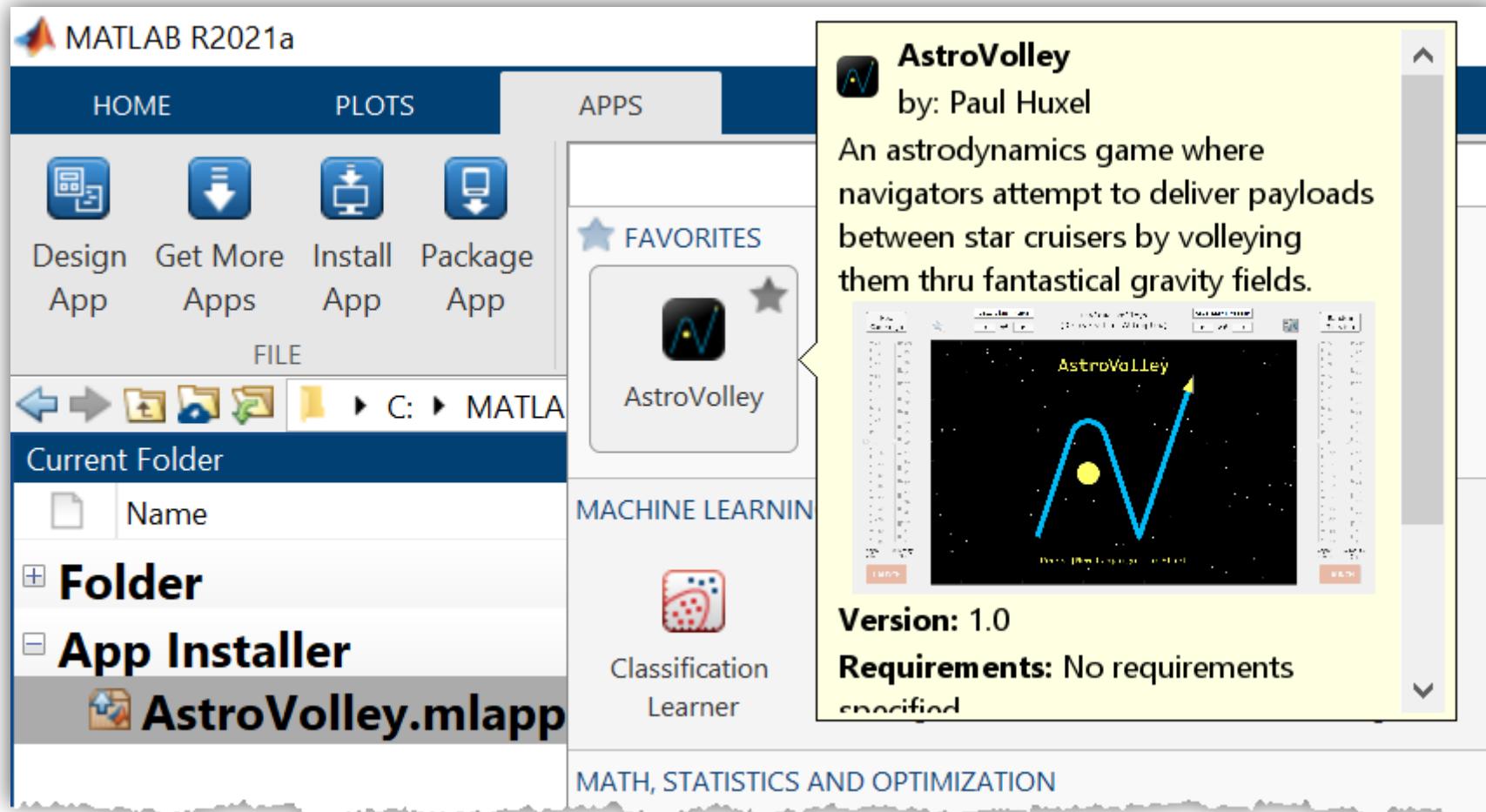
Installing AstroVolley

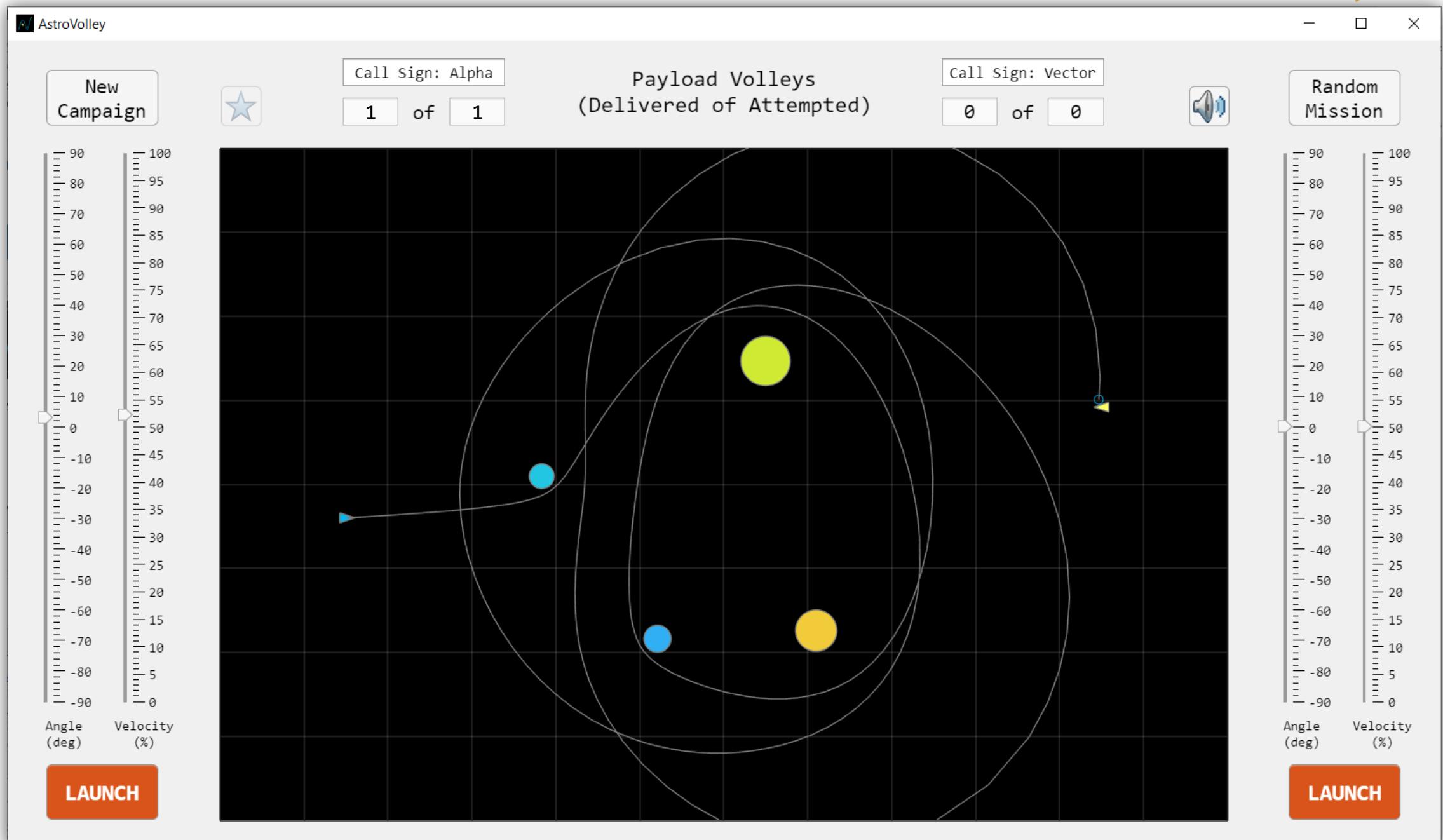
Navigate to and double-click on "AstroVolley.mlappinstall"



Playing AstroVolley

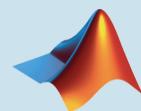
Find and select AstroVolley from the APPS tab





Course Lesson Plan

- Installing and Playing AstroVolley



Plotting Ships (Triangles)

- Plotting Grav. Bodies (Circles)
- Gravity and Orbits
- Launching Volleys
 - Creating a Random Mission Map
 - 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 Map
- 5) Keeping Score

Represent game pieces using basic shapes...



Image Credit: NASA
GSFC_20171208
Archive_e002131

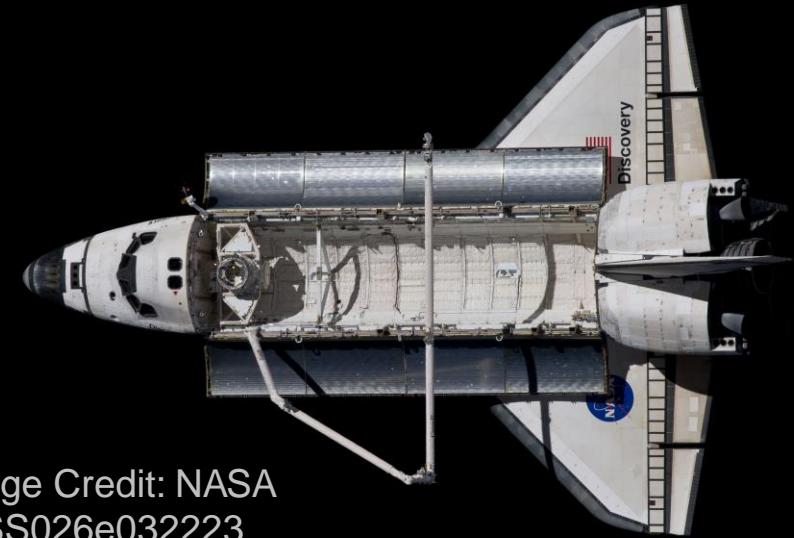


Image Credit: NASA
ISS026e032223

Represent game pieces using basic shapes...

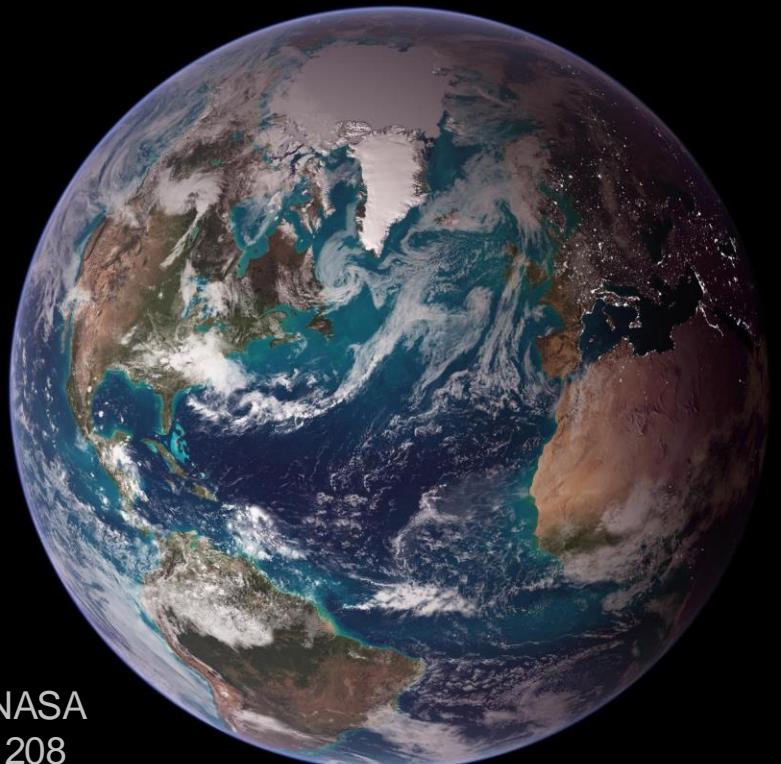


Image Credit: NASA
GSFC_20171208
Archive_e002131

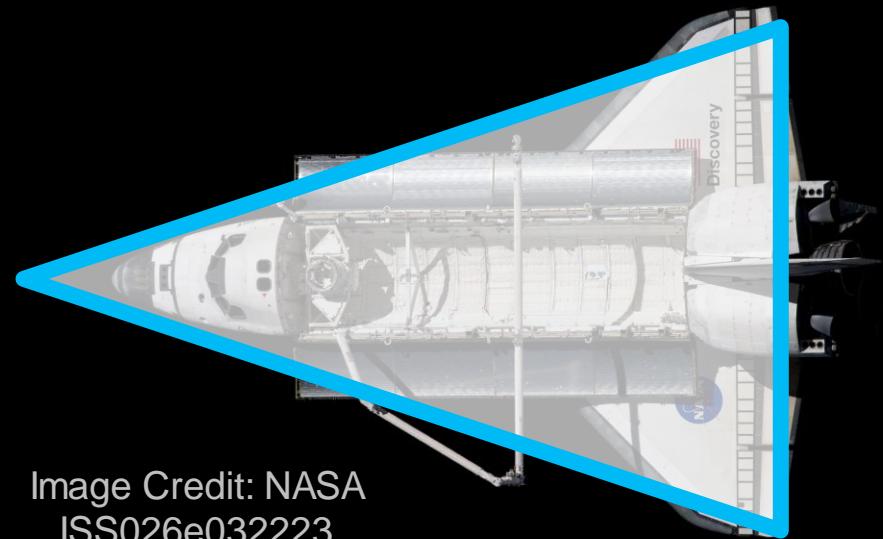
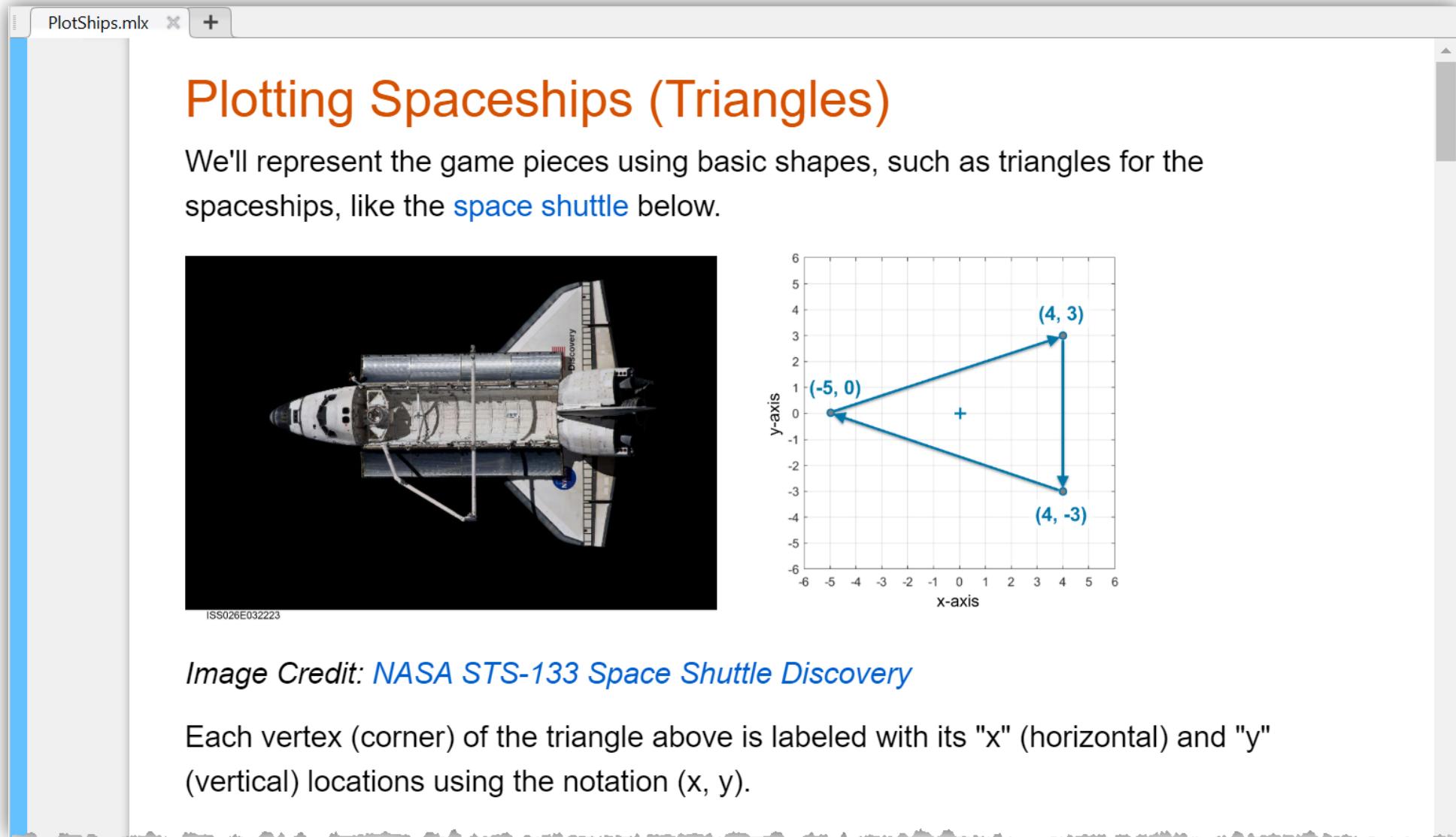


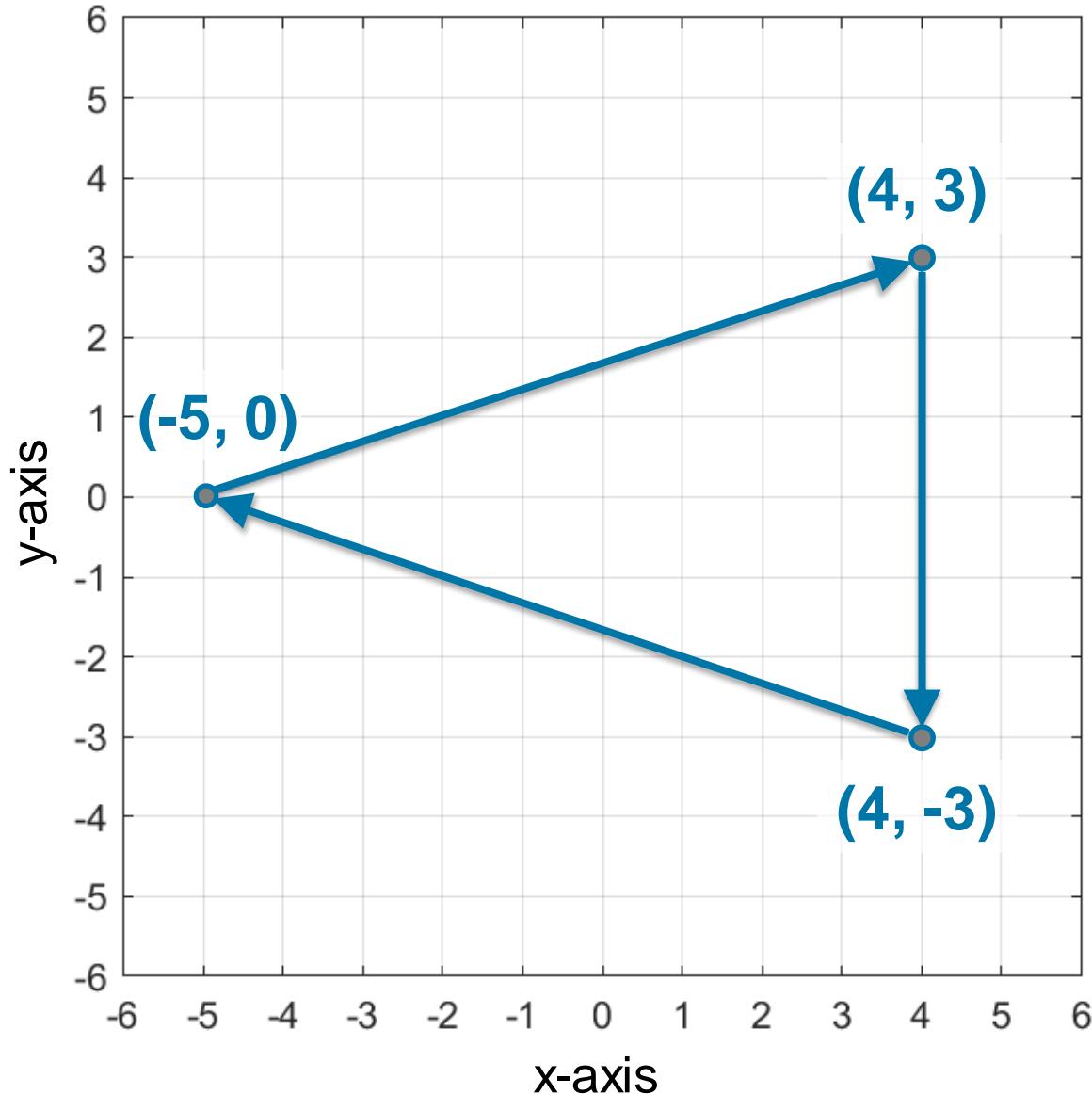
Image Credit: NASA
ISS026e032223

Triangles for the
Spaceships
(Space Shuttle)

```
>> open PlotShips mlx
```

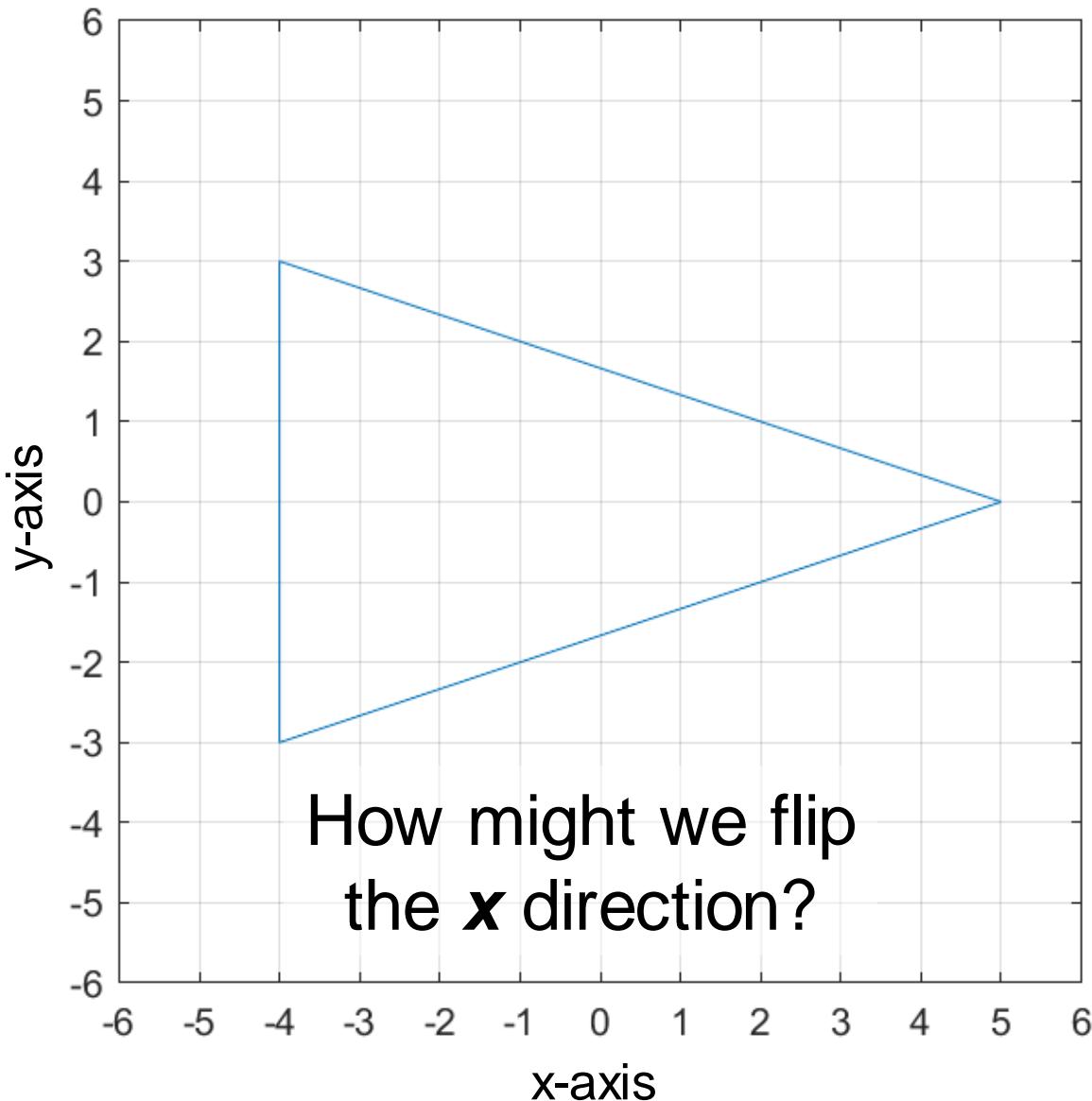


Plotting Spaceships (Triangles)



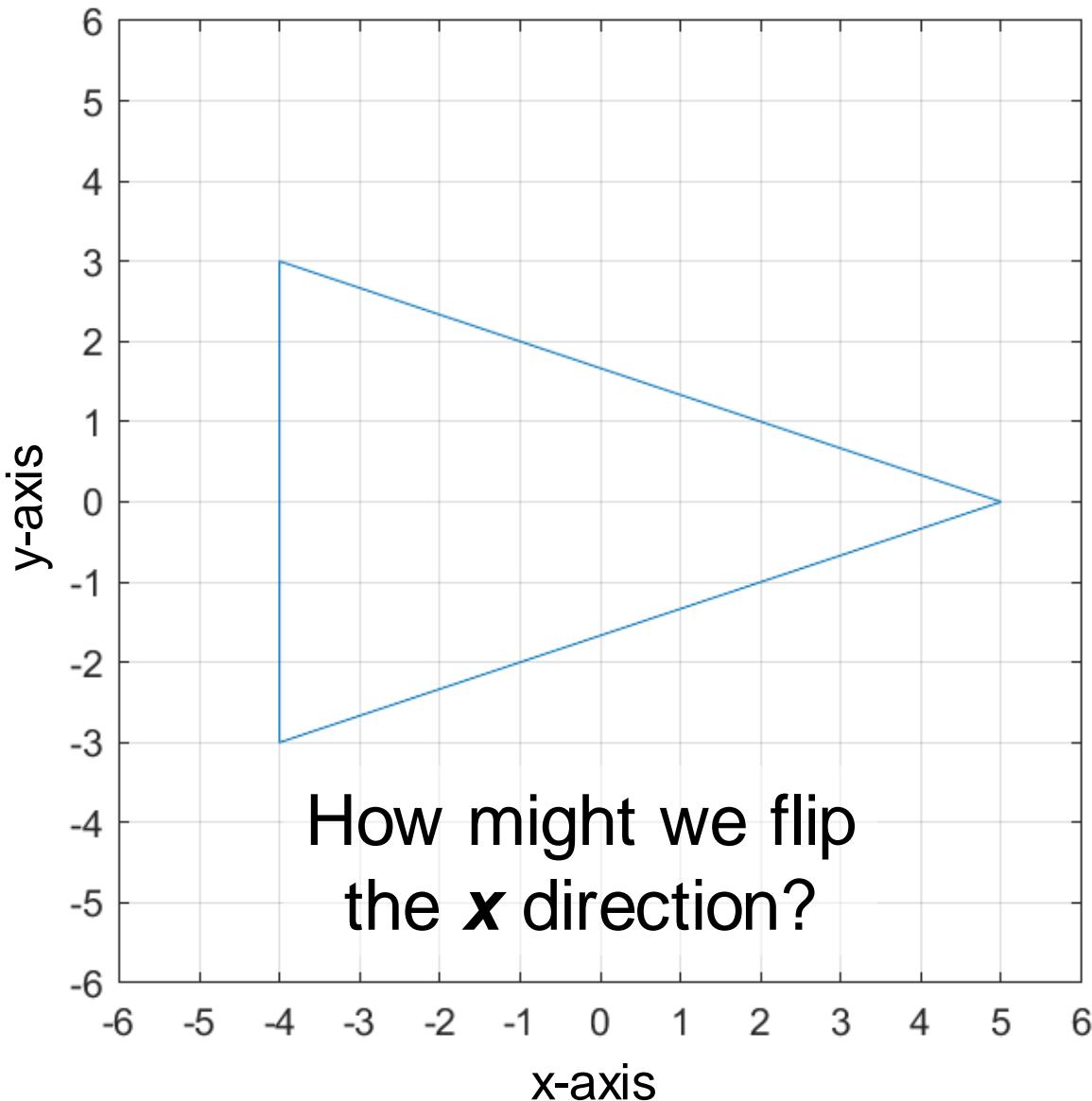
```
>> xt = [-5 4 4 -5]; % tri x-pts  
  
>> yt = [ 0 3 -3 0]; % tri y-pts  
  
>> ax = gca; % get current axes  
  
>> plot(ax,xt,yt)
```

Plotting Spaceships (Triangles)



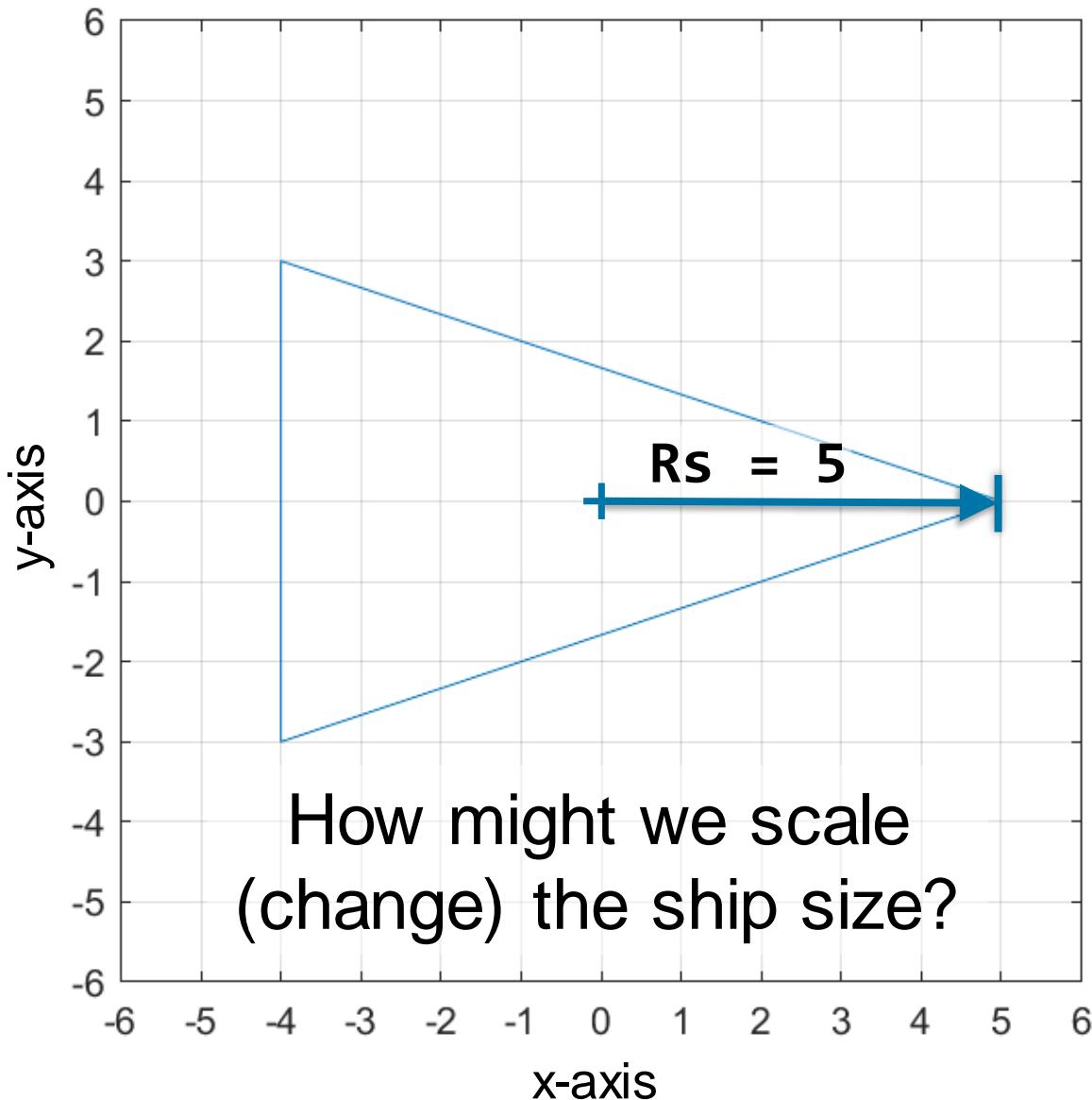
```
>> xt = [-5 4 4 -5]; % tri x-pts  
  
>> yt = [ 0 3 -3 0]; % tri y-pts  
  
>> ax = gca; % get current axes  
  
>> plot(ax, xt,yt) % point < left
```

Plotting Spaceships (Triangles)



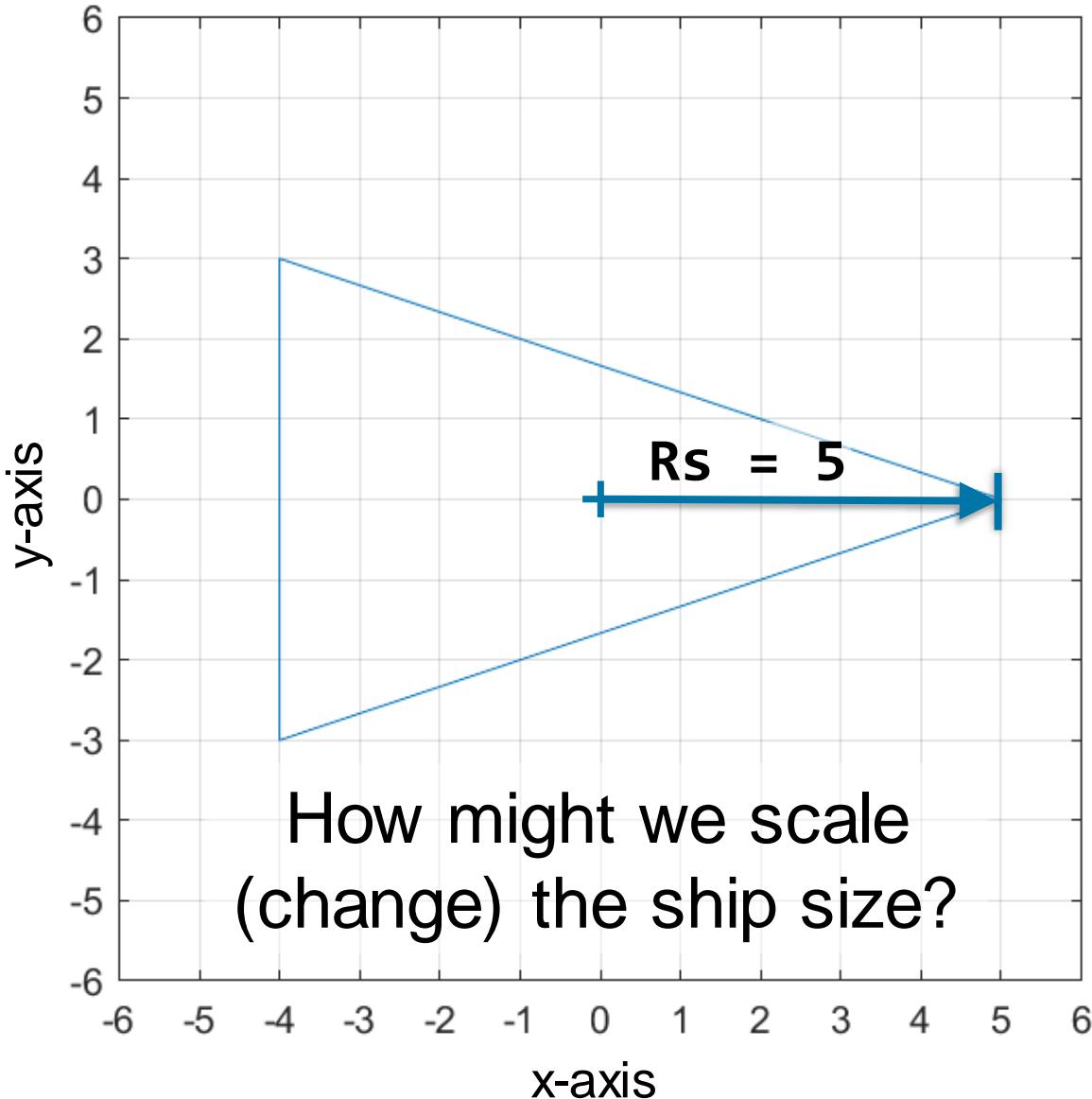
```
>> xt = [-5 4 4 -5]; % tri x-pts  
  
>> yt = [ 0 3 -3 0]; % tri y-pts  
  
>> ax = gca; % get current axes  
  
>> plot(ax, xt,yt) % point < left  
  
>> plot(ax, -xt,yt) % point > right
```

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
```

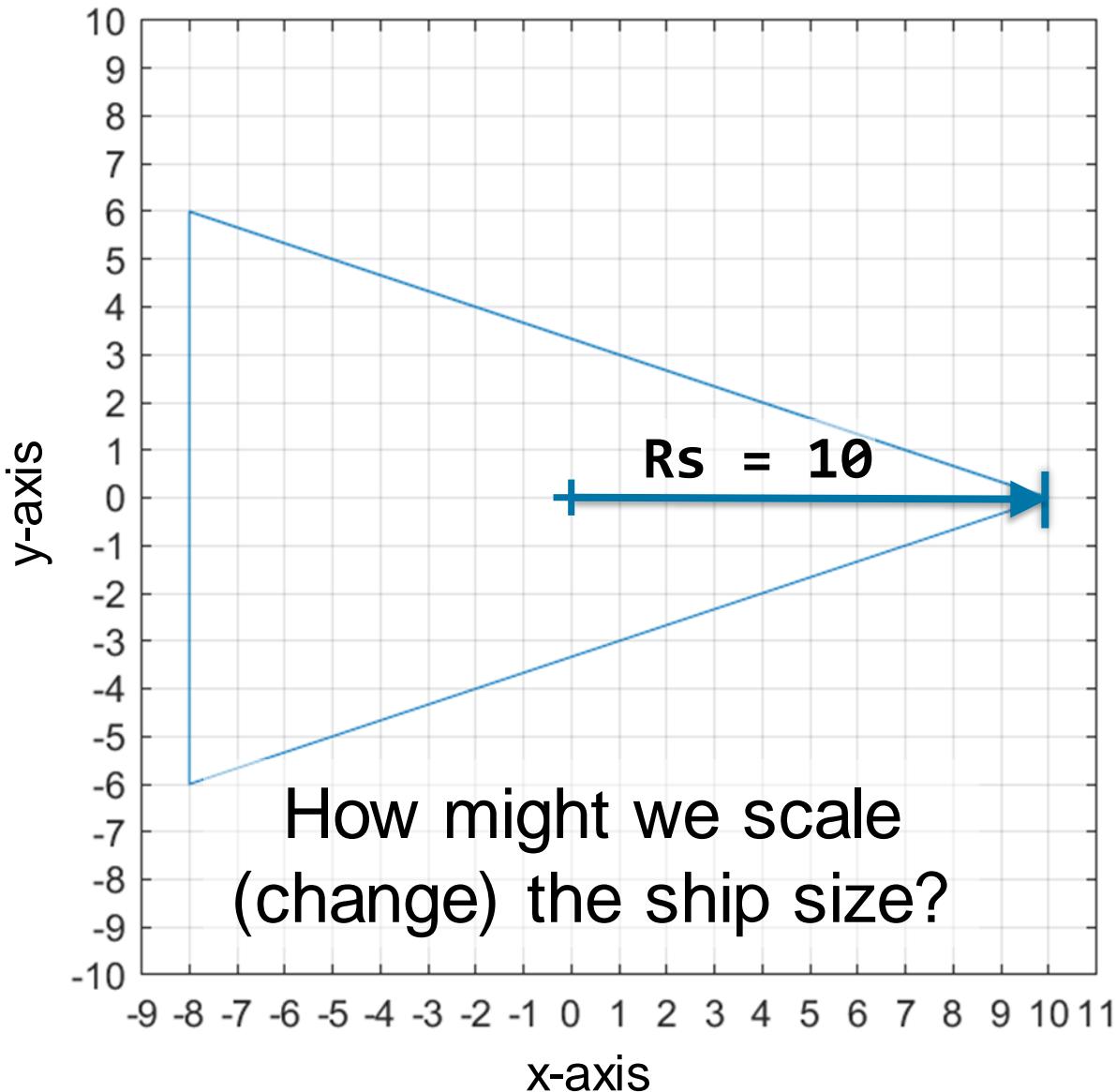
```
>> ax = gca; % get current axes
```

```
>> 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
```

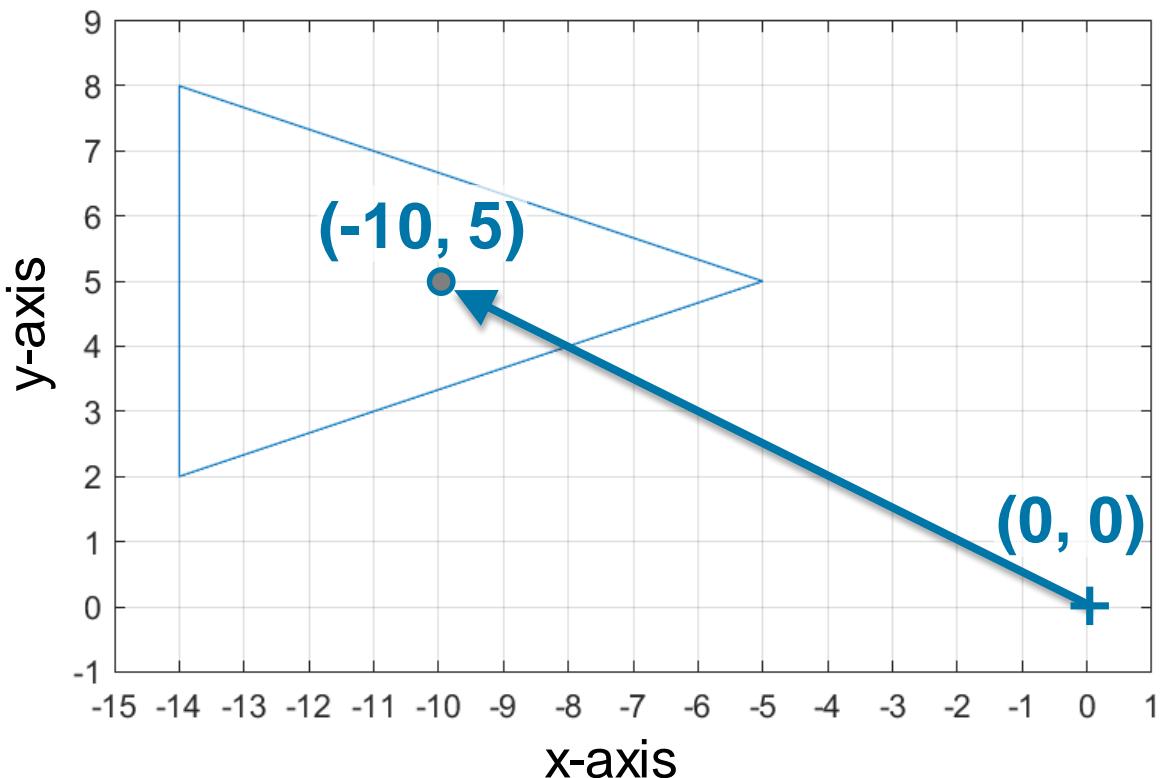
```
>> ax = gca; % get current axes
```

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

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



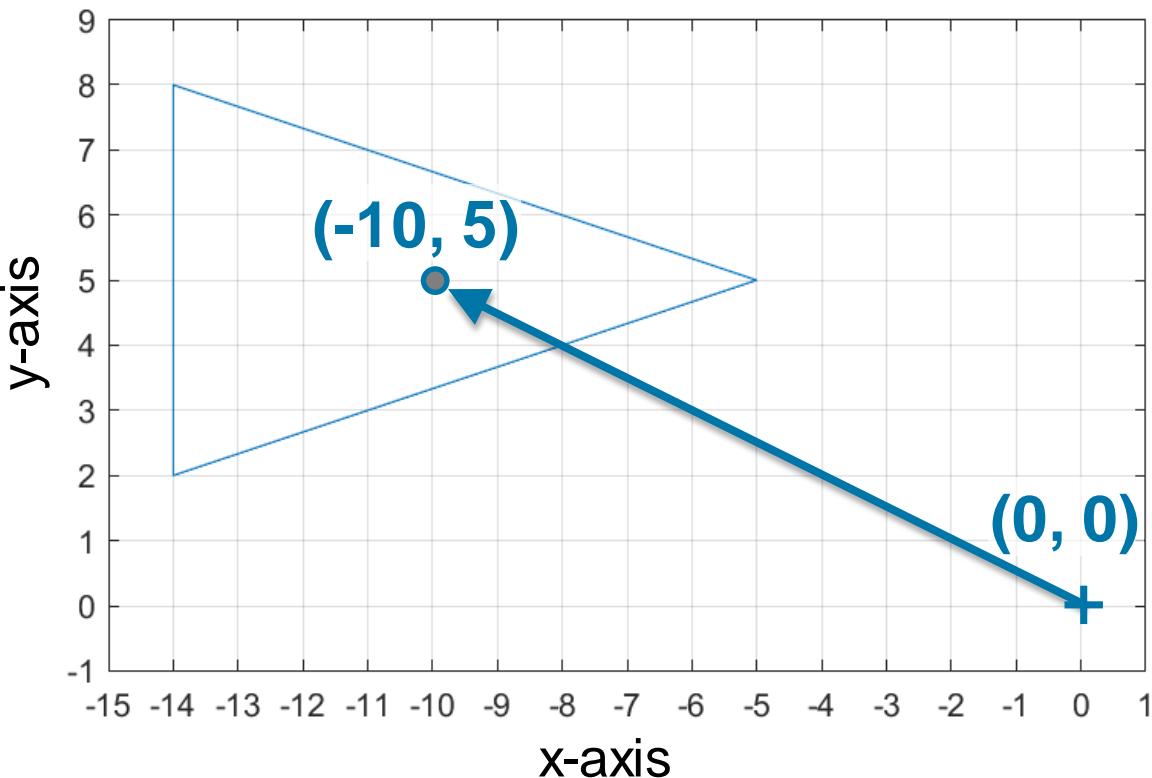
Plotting Spaceships (Triangles)



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)

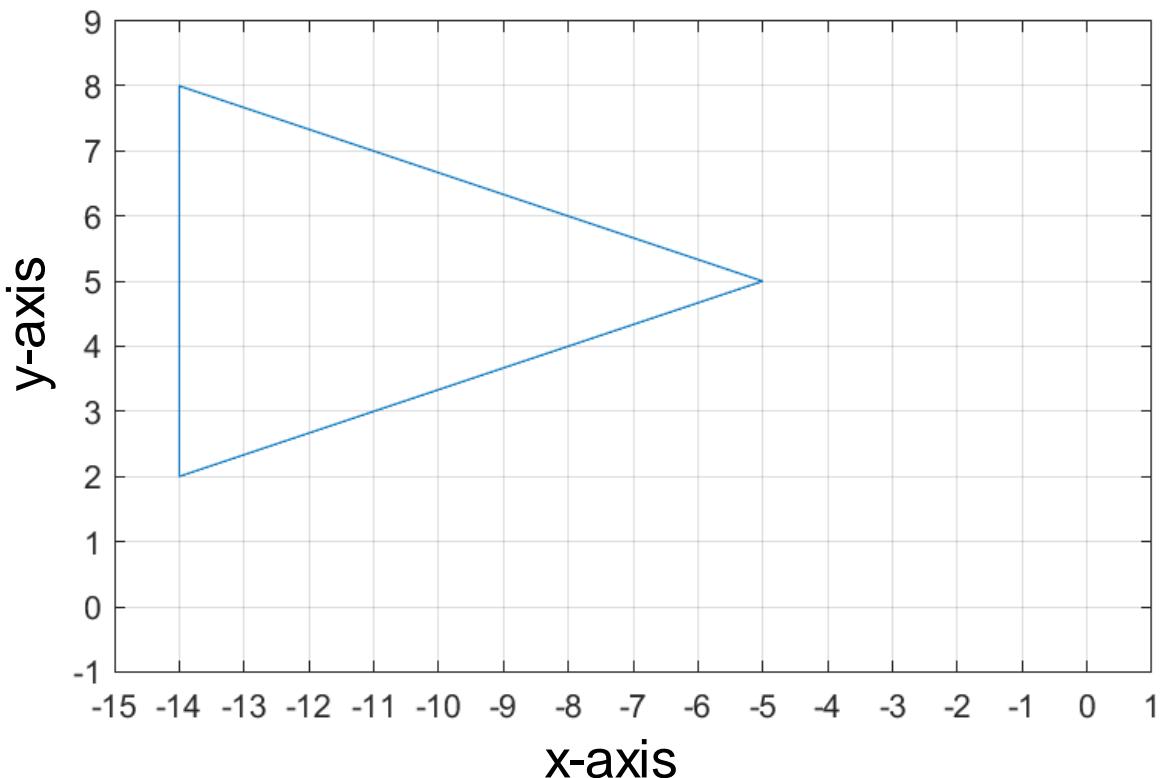


How might we reposition
the spaceship?

```
>> 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)
```



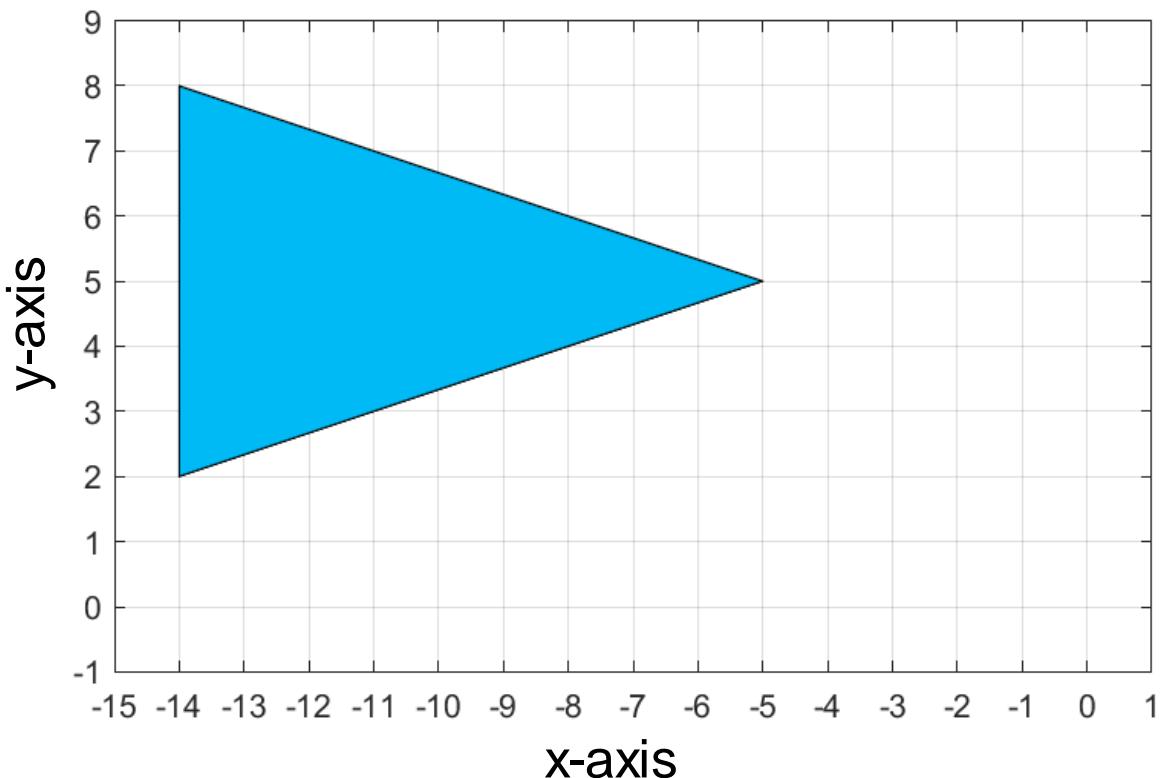
Plotting Spaceships (Triangles)



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)
```

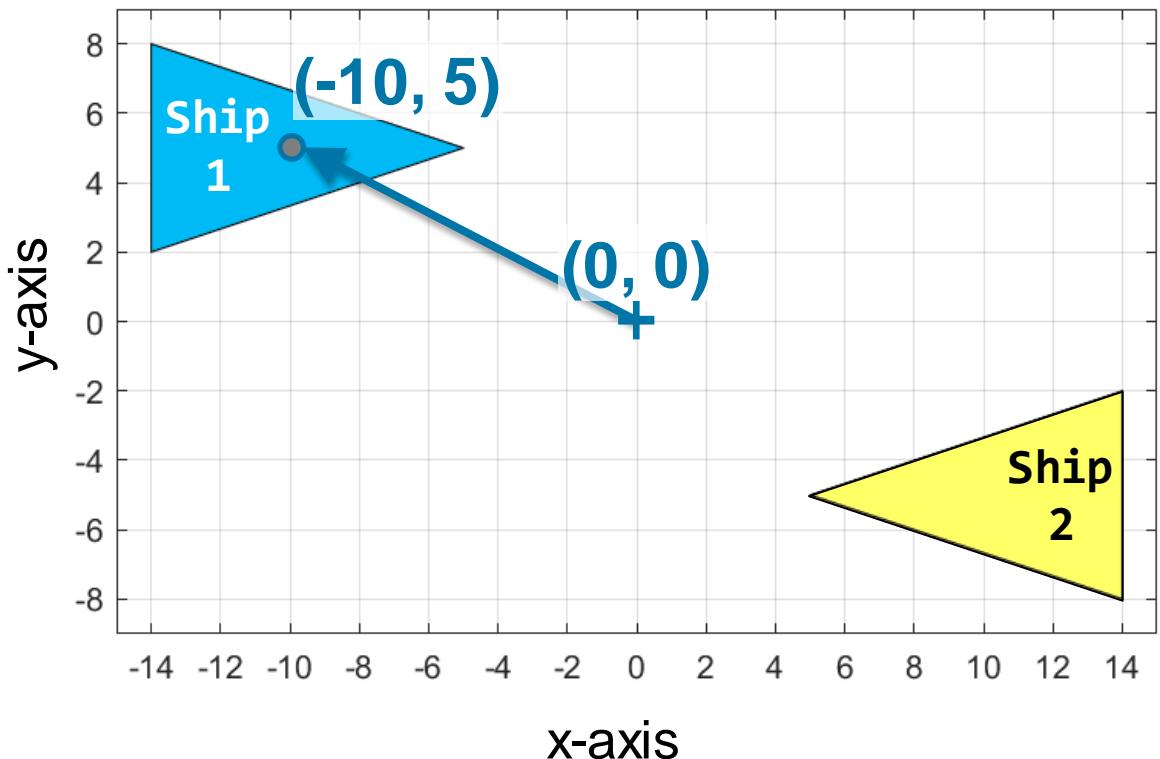
Plotting Spaceships (Triangles)



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)

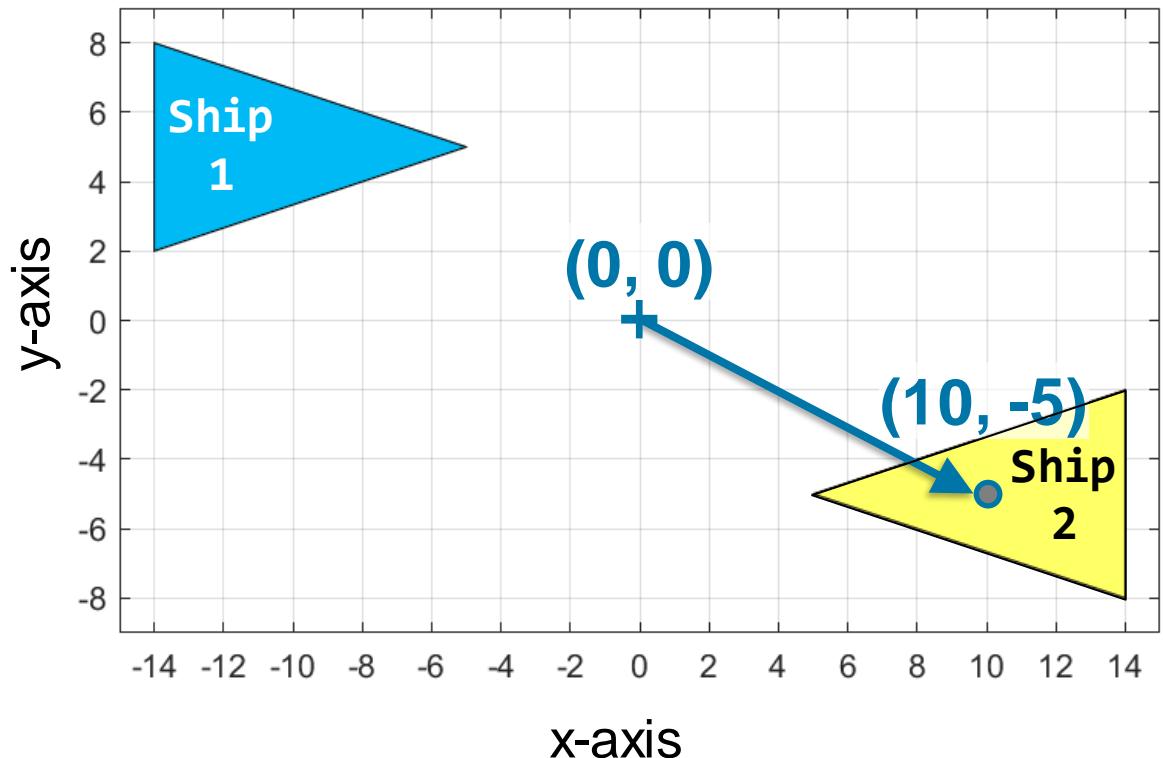


Ship 1

```
>> 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
```

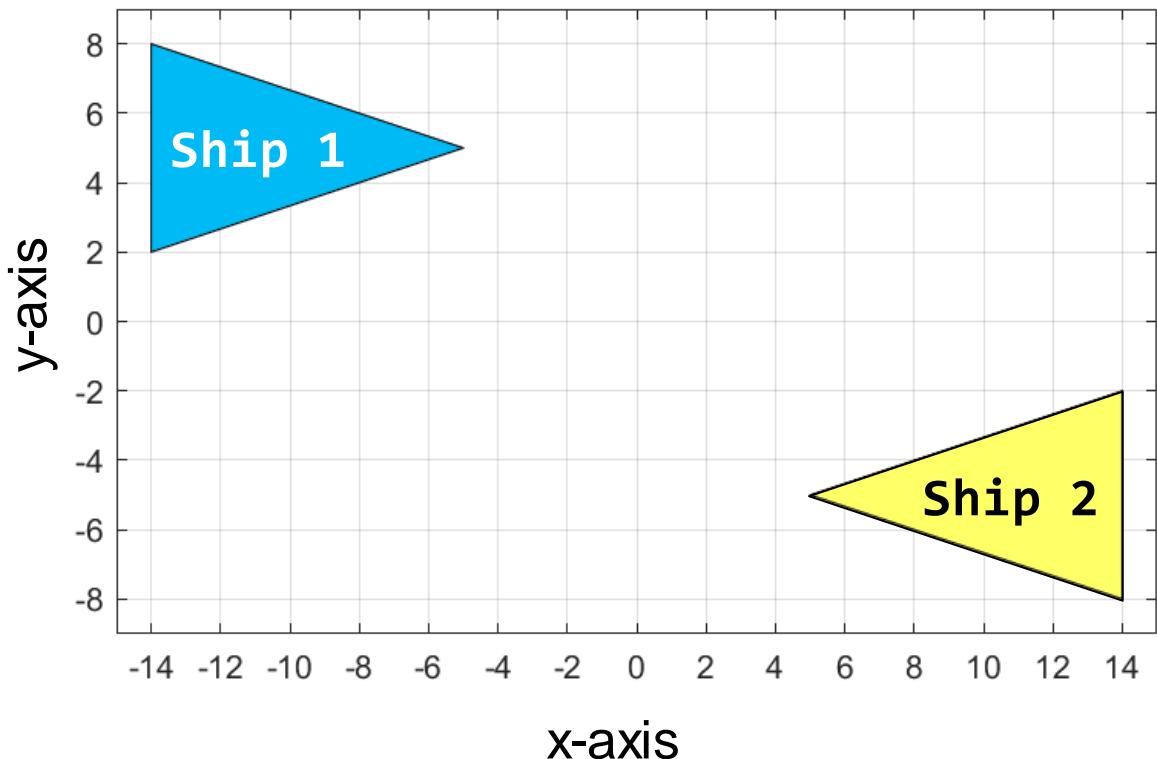
```
>> Rs = [ 5 5]; % ship size
```

```
>> 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

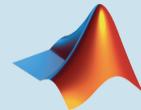
>> hold(ax, "on")

>> 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 Random Mission Map
 - 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 Map
- 5) Keeping Score

Represent game pieces using basic shapes...

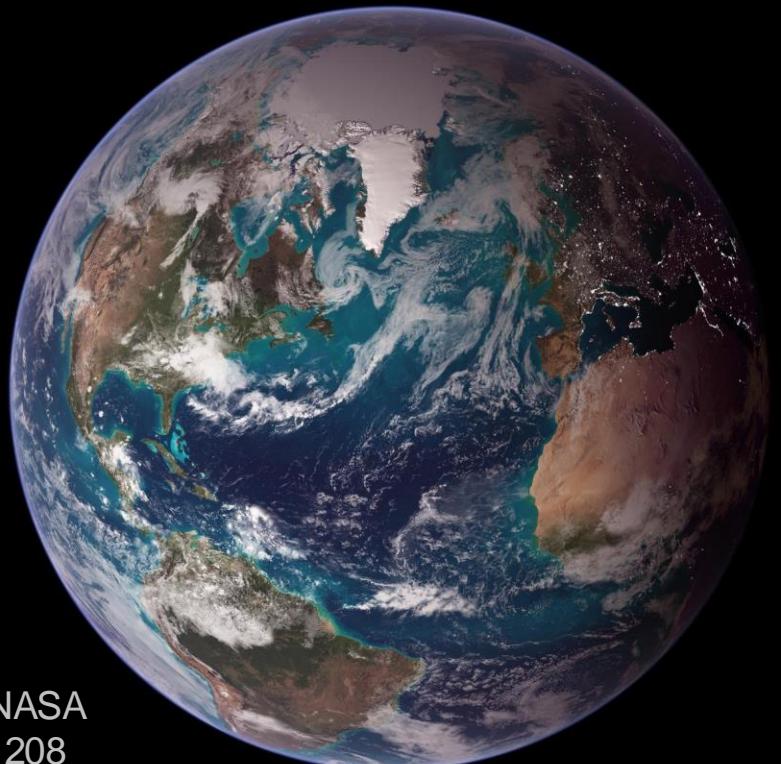


Image Credit: NASA
GSFC_20171208
Archive_e002131

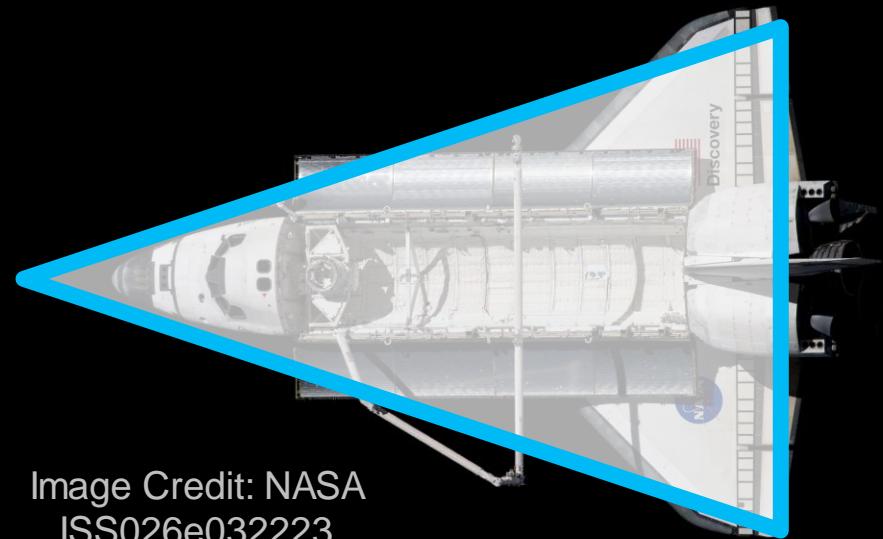


Image Credit: NASA
ISS026e032223

Triangles for the
Spaceships
(Space Shuttle)

Represent game pieces using basic shapes...

Circles for the
Gravitational Bodies
(Planets)

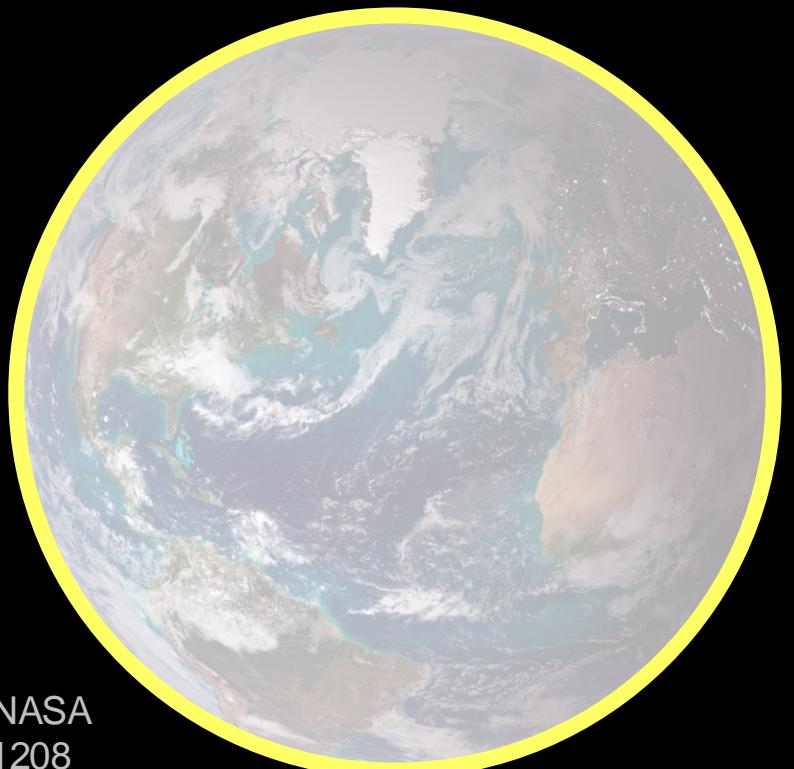


Image Credit: NASA
GSFC_20171208
Archive_e002131

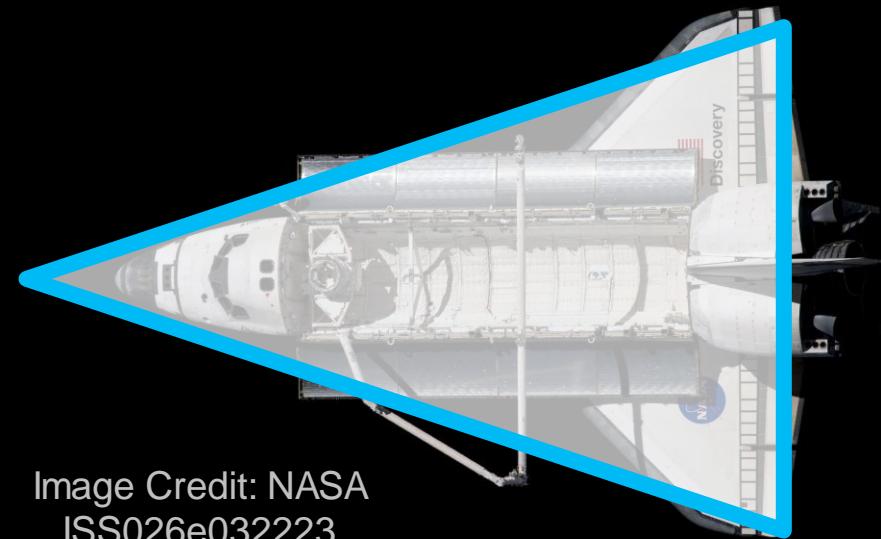
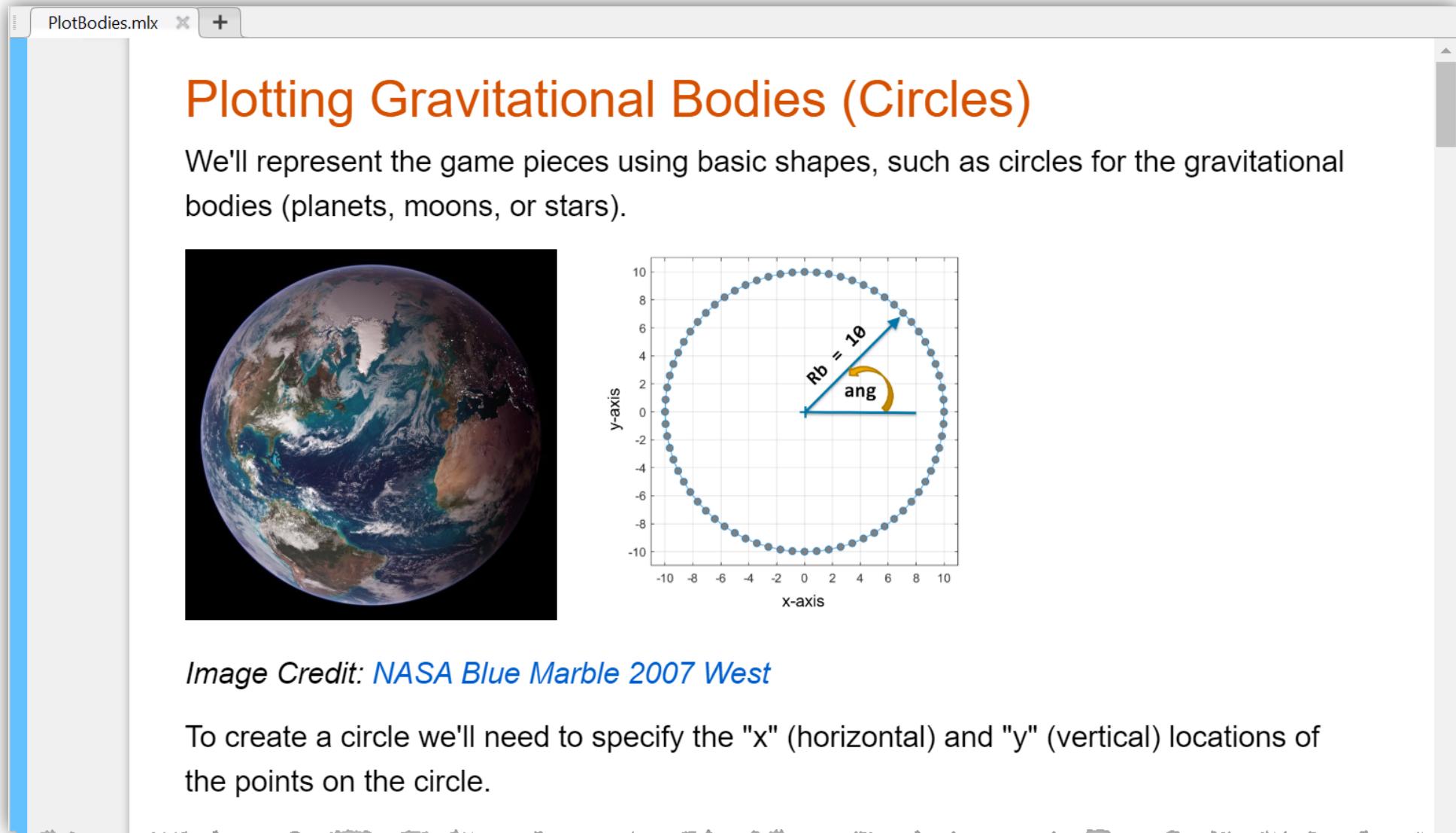


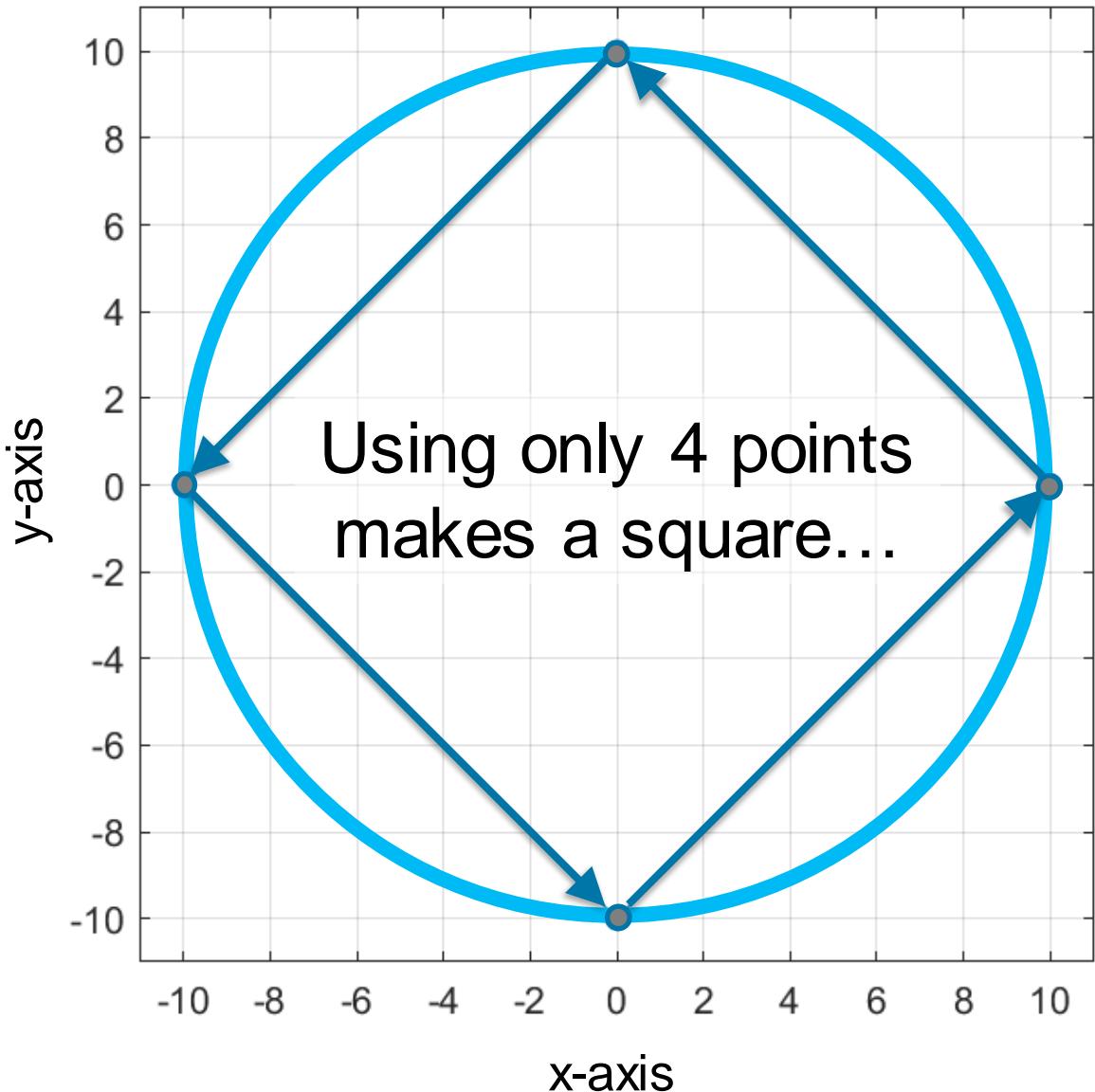
Image Credit: NASA
ISS026e032223

Triangles for the
Spaceships
(Space Shuttle)

```
>> open PlotBodies mlx
```

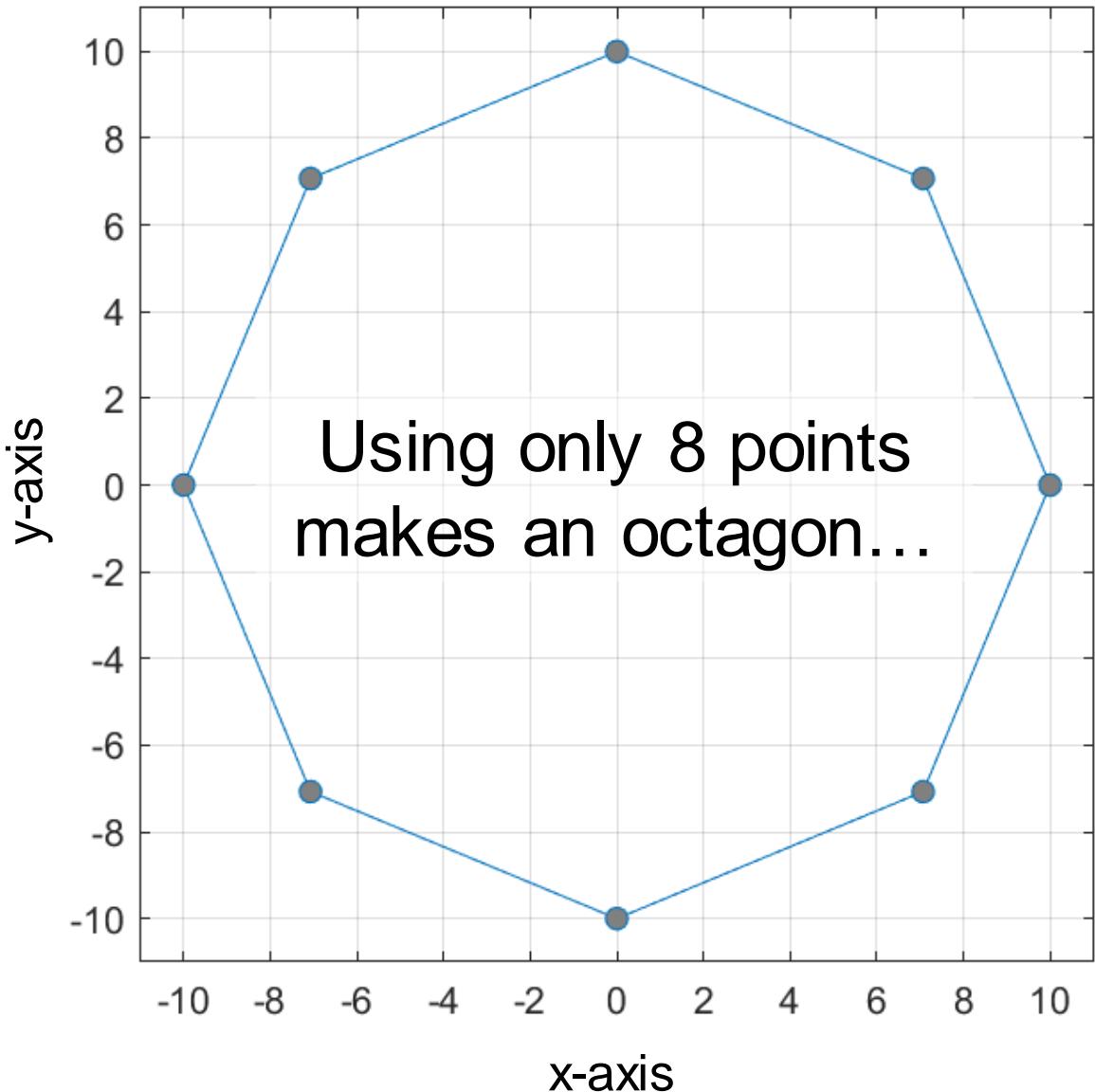


Plotting Grav. Bodies (Circles)



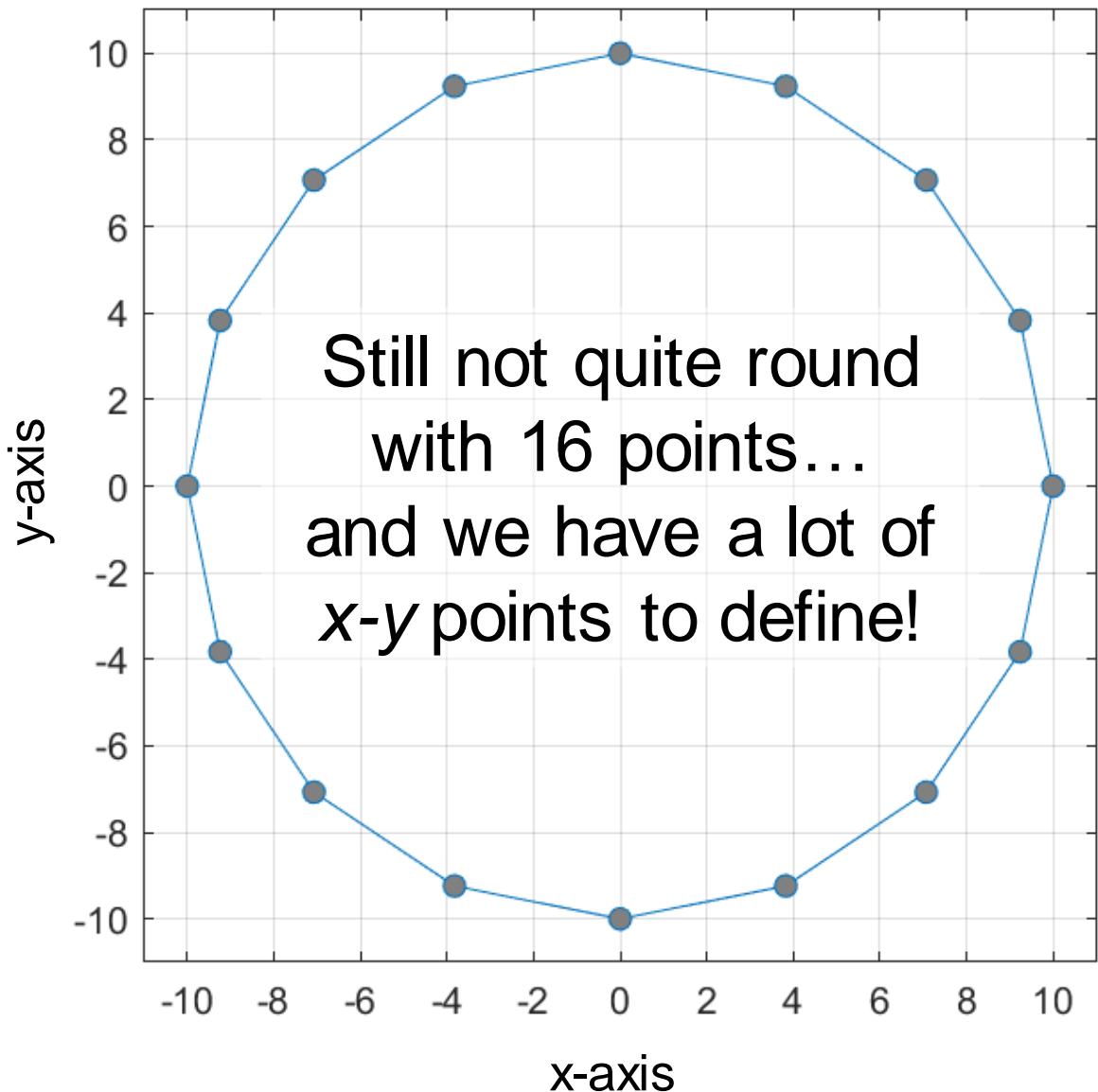
```
>> ax = gca; % get current axes  
  
>> xc = [10 0 -10 0 10];  
  
>> yc = [0 10 0 -10 0];  
  
>> plot(ax,xc,yc)
```

Plotting Grav. Bodies (Circles)



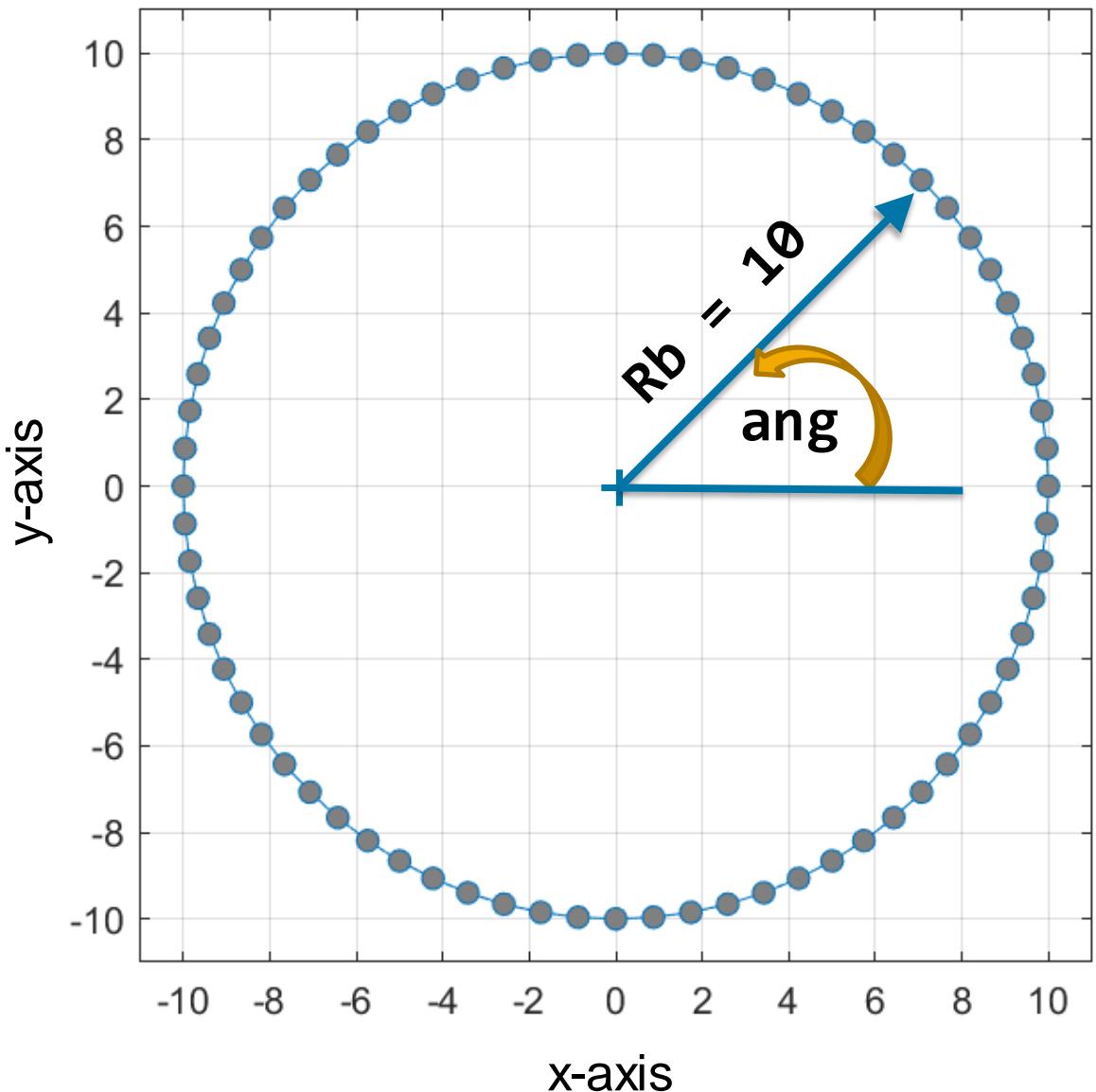
```
>> ax = gca; % get current axes  
  
>> xc = [10 7.07 0 -7.07 ...  
         -10 -7.07 0 -7.07 10];  
  
>> yc = [0 7.07 10 7.07 ...  
         0 -7.07 -10 -7.07 0];  
  
>> plot(ax,xc,yc)
```

Plotting Grav. Bodies (Circles)



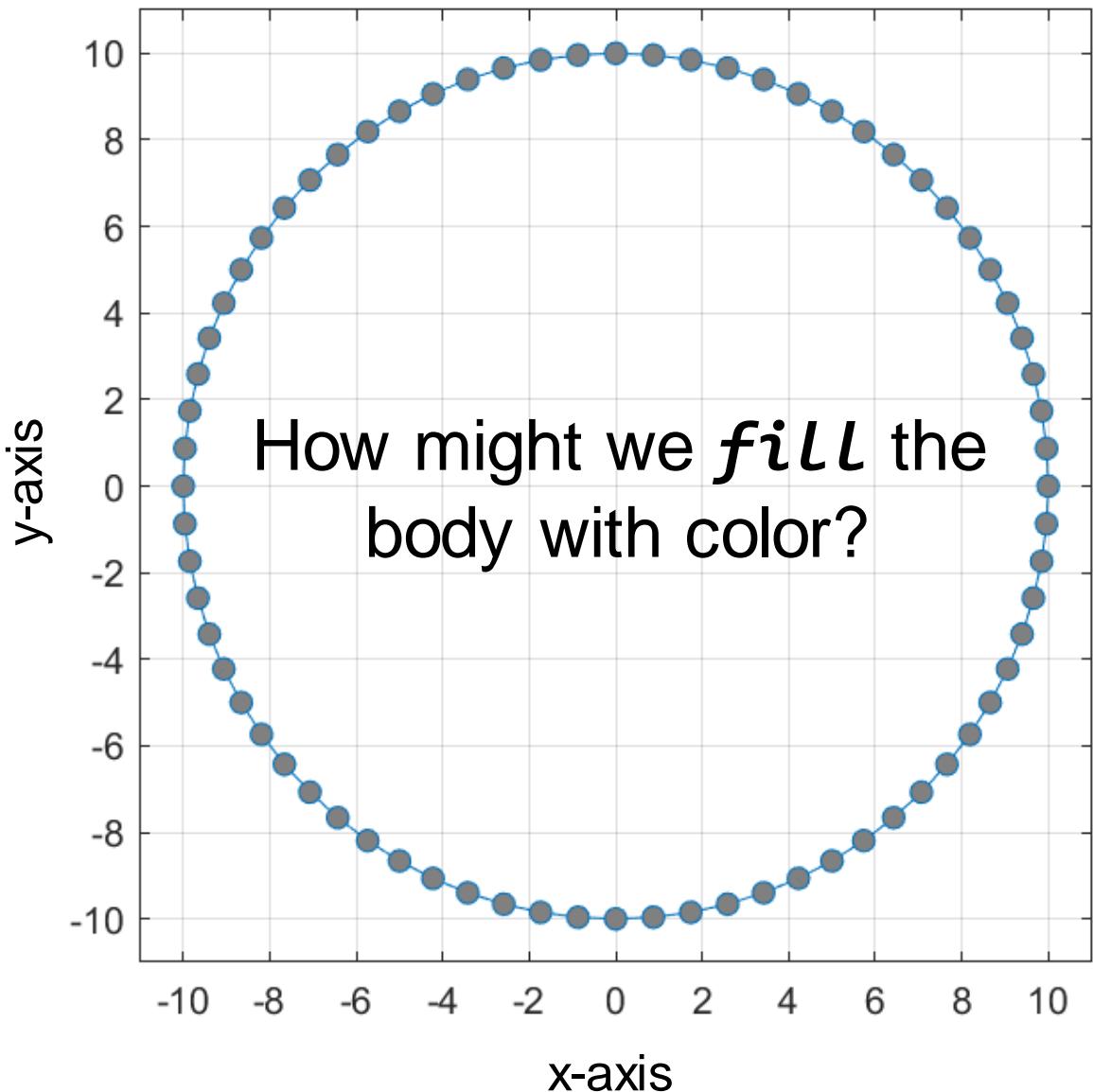
```
>> ax = gca; % get current axes  
  
>> xc = [10 9.24 7.07 3.83 ...  
          0 -3.83 -7.07 -9.24 ...  
         -10 -9.24 -7.07 -3.83 ...  
          0 3.83 7.07 9.24 10];  
  
>> yc = [ 0 3.83 7.07 9.24 ...  
          10 9.24 7.07 3.83 ...  
          0 -3.83 -7.07 -9.24 ...  
         -10 -9.24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



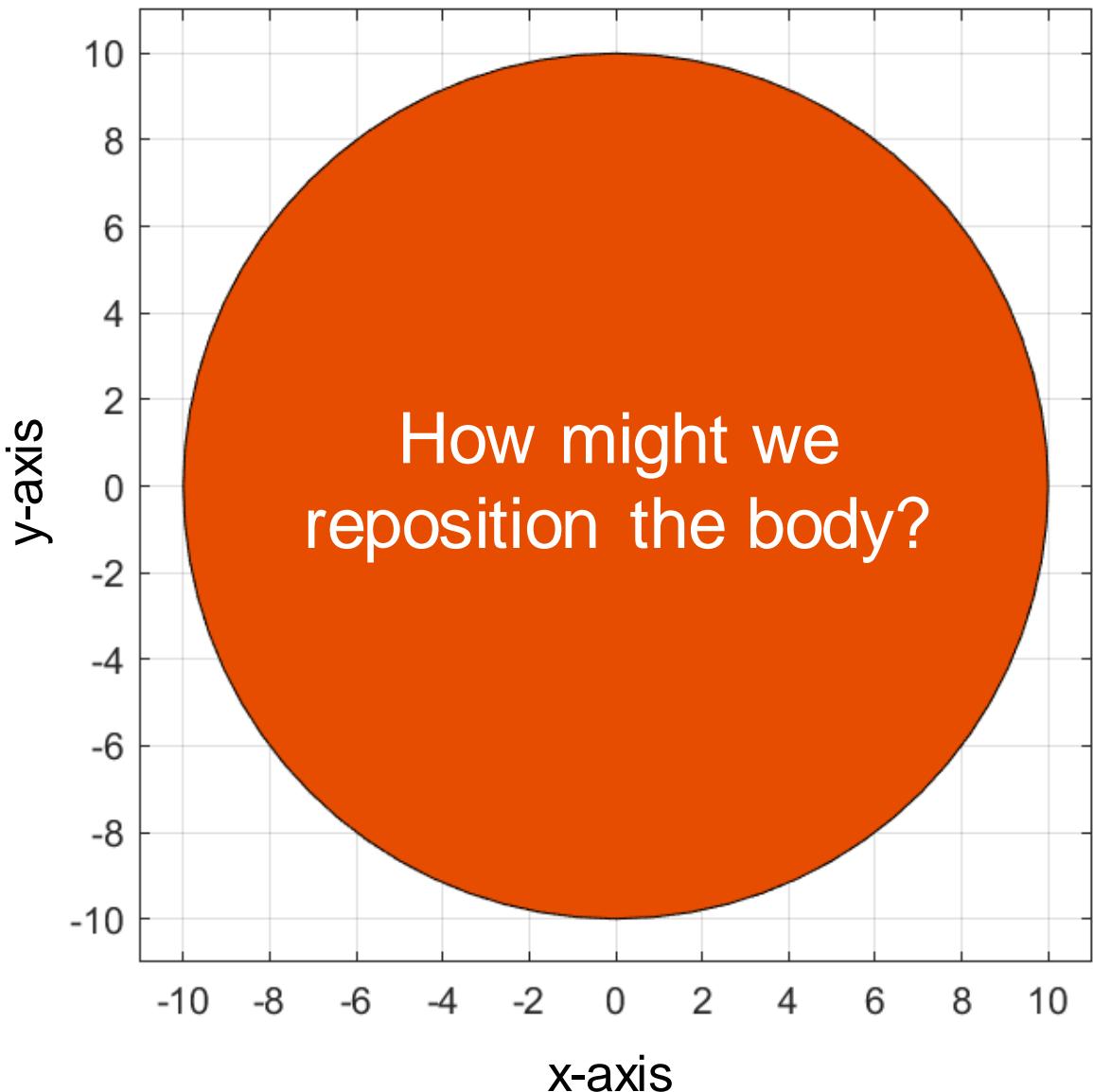
```
>> Rb = 10; % body size  
>> ang = 0:5:360; % degrees  
>> xc = cosd(ang); % circle x-pts  
>> yc = sind(ang); % circle y-pts  
>> plot(ax,Rb*xc,Rb*yc)  
>> axis(ax,"equal")
```

Plotting Grav. Bodies (Circles)



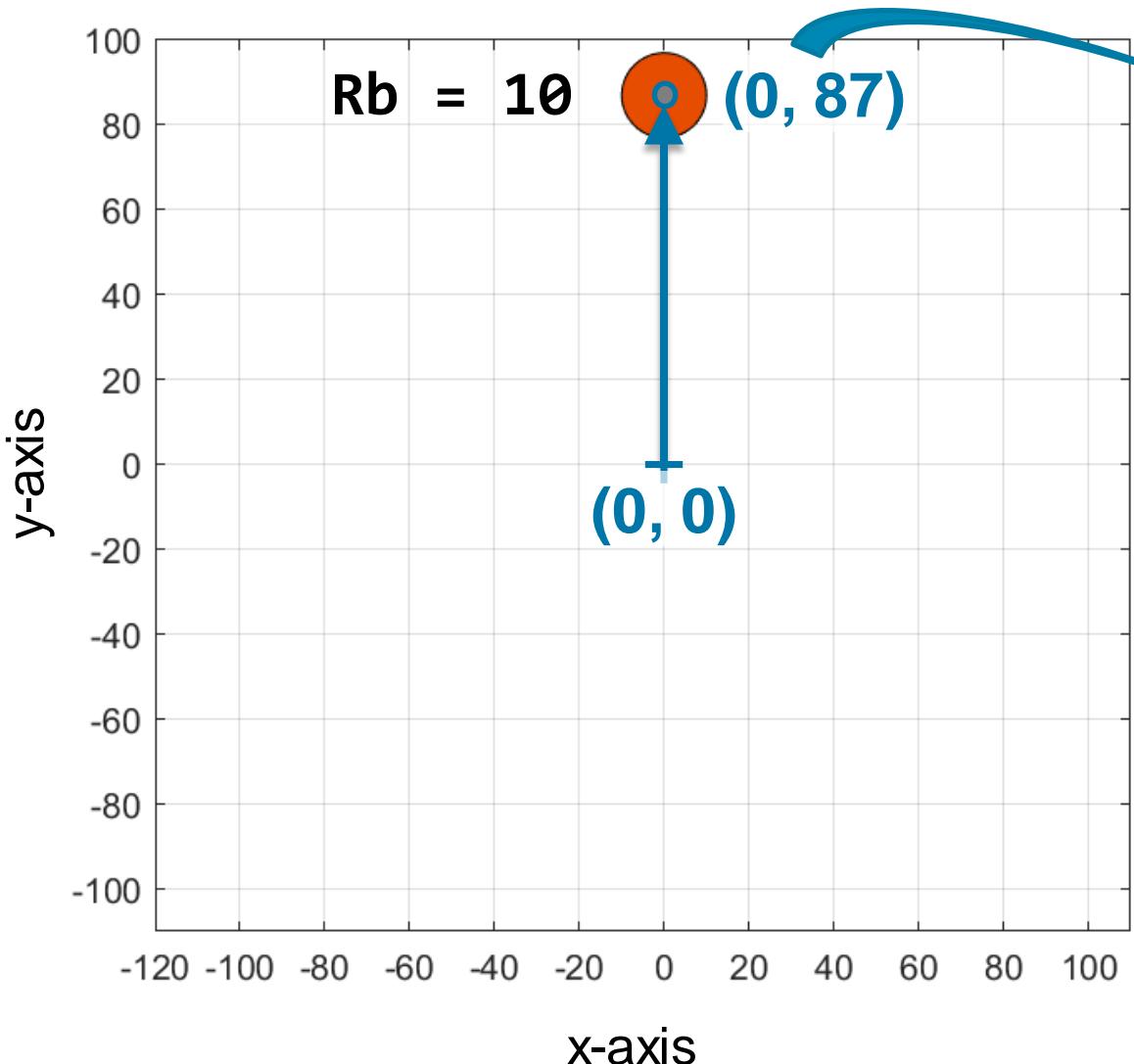
```
>> Rb = 10; % body size  
>> ang = 0:5:360; % degrees  
>> xc = cosd(ang); % circle x-pts  
>> yc = sind(ang); % circle y-pts  
>> fill(ax,Rb*xc,Rb*yc, ...  
      [.9 .3 0]) % RGB Color  
>> axis(ax,"equal")
```

Plotting Grav. Bodies (Circles)



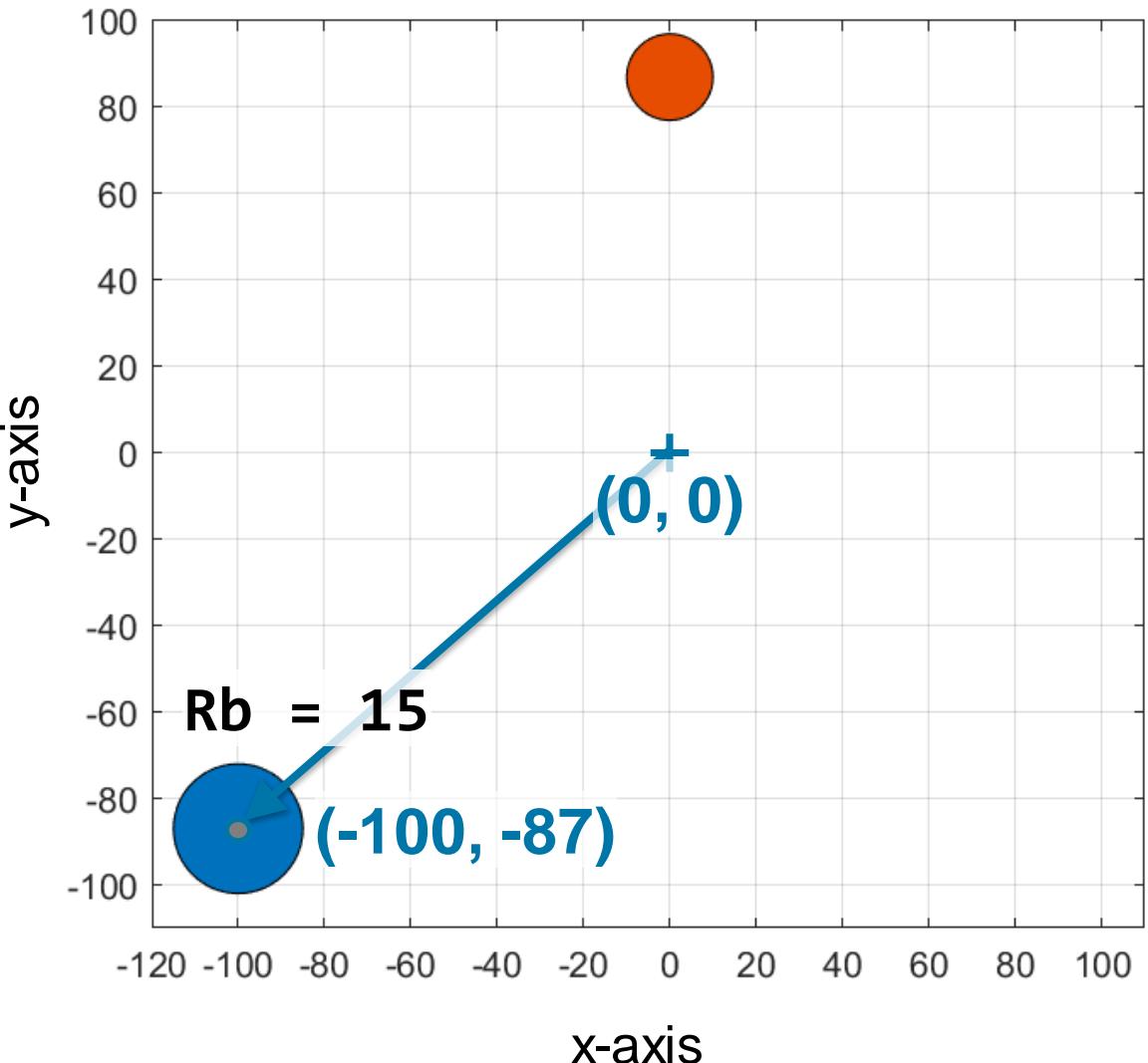
```
>> Rb = 10; % body size  
  
>> ang = 0:5:360; % degrees  
  
>> xc = cosd(ang); % circle x-pts  
  
>> yc = sind(ang); % circle y-pts  
  
>> fill(ax,Rb*xc,Rb*yc, ...  
      [.9 .3 0]) % RGB Color  
  
>> axis(ax,"equal")
```

Plotting Gravitational Bodies (Circles)



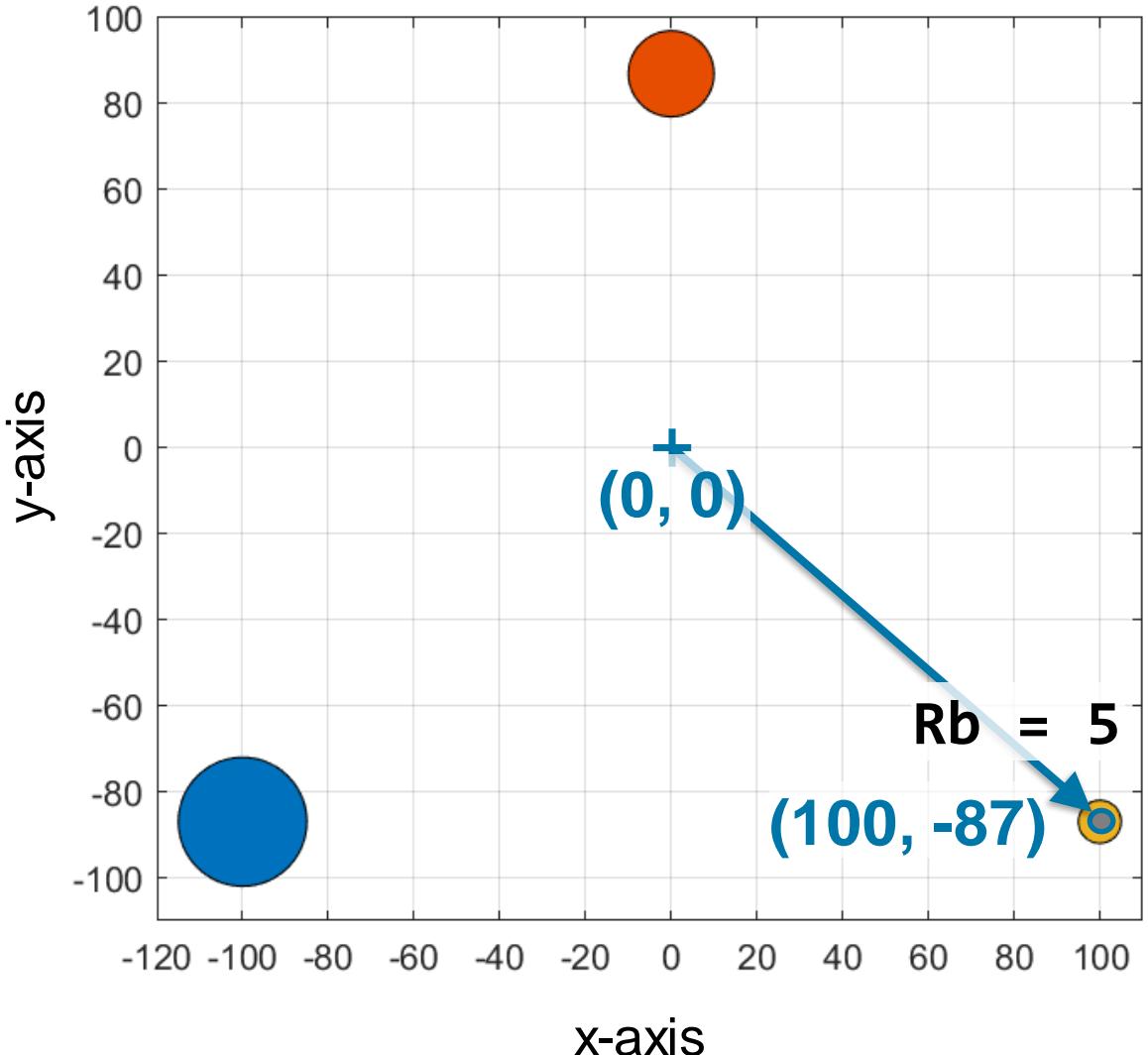
```
>> fill(ax, 0+10*xc, ...
          87+10*yc, ...
          [.9 .3 0]) % RGB Color
```

Plotting Gravitational Bodies (Circles)



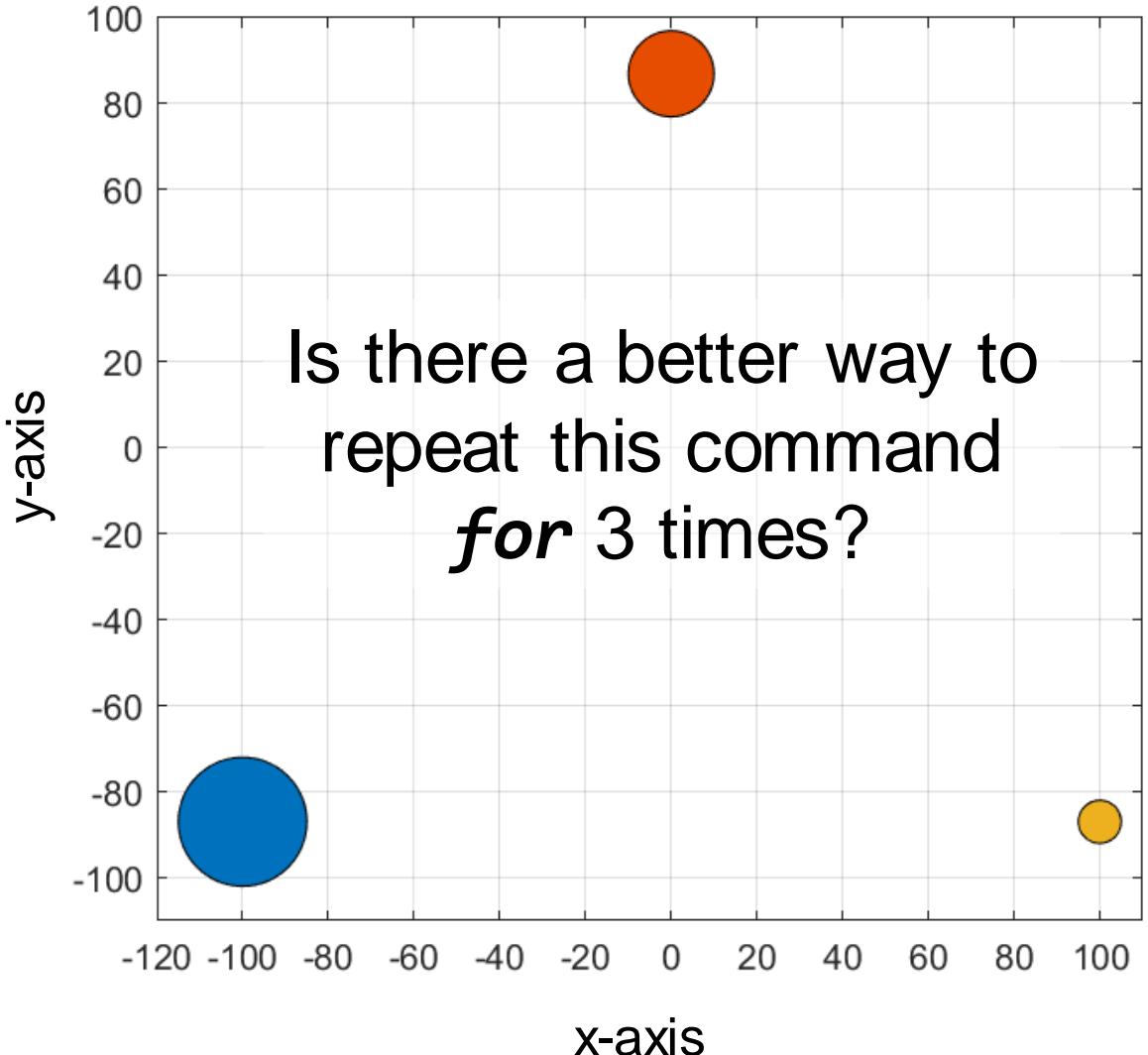
```
>> fill(ax, -100+15*xc, ...  
      -87+15*yc, ...  
      [0 .4 .7]) % RGB Color  
  
>> fill(ax, 0+10*xc, ...  
      87+10*yc, ...  
      [.9 .3 0]) % RGB Color
```

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
```

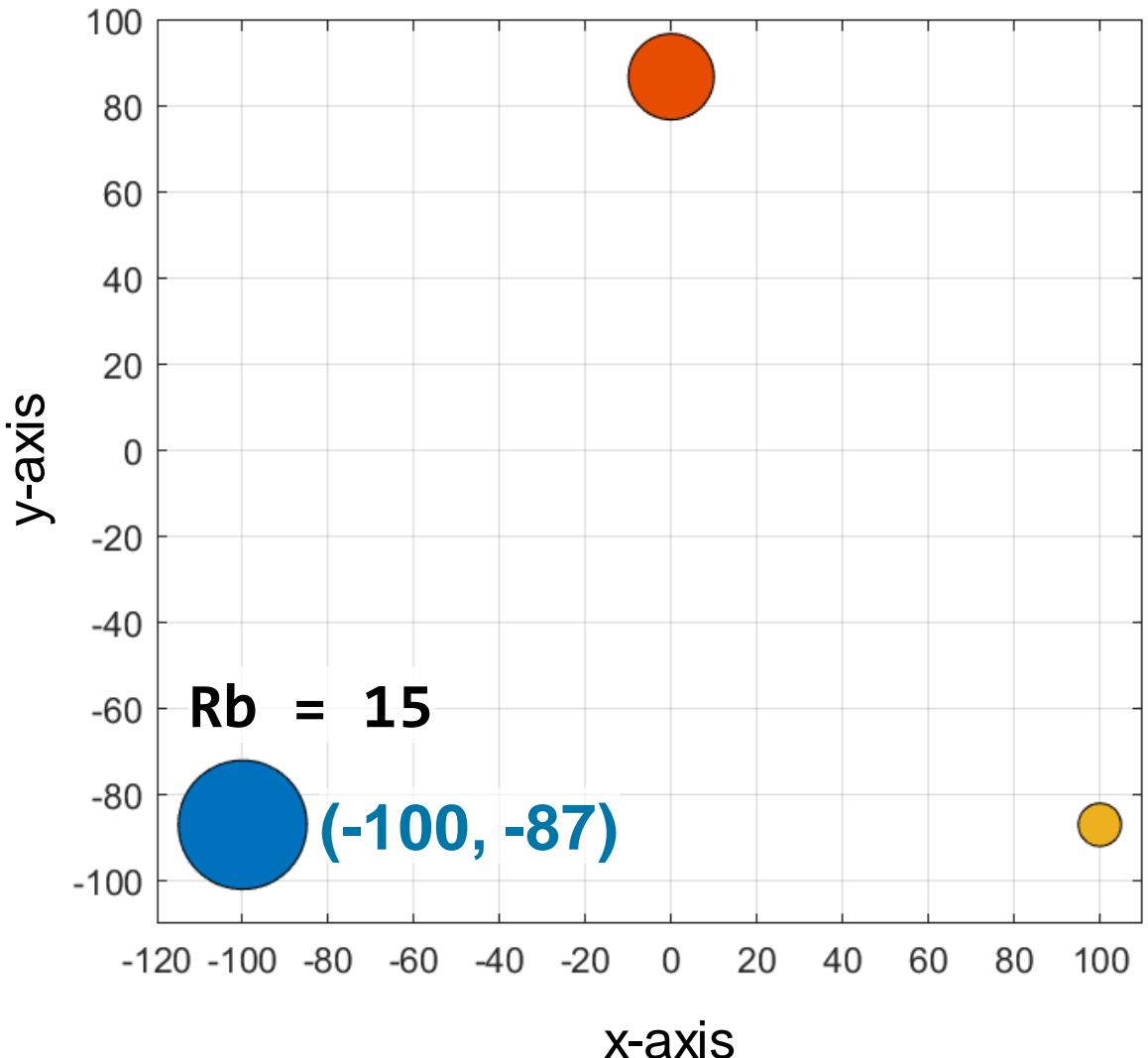


```
>> fill(ax, 0+10*xc, ...
         87+10*yc, ...
         [.9 .3 0]) % RGB Color
```



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

Plotting Gravitational Bodies (Circles)



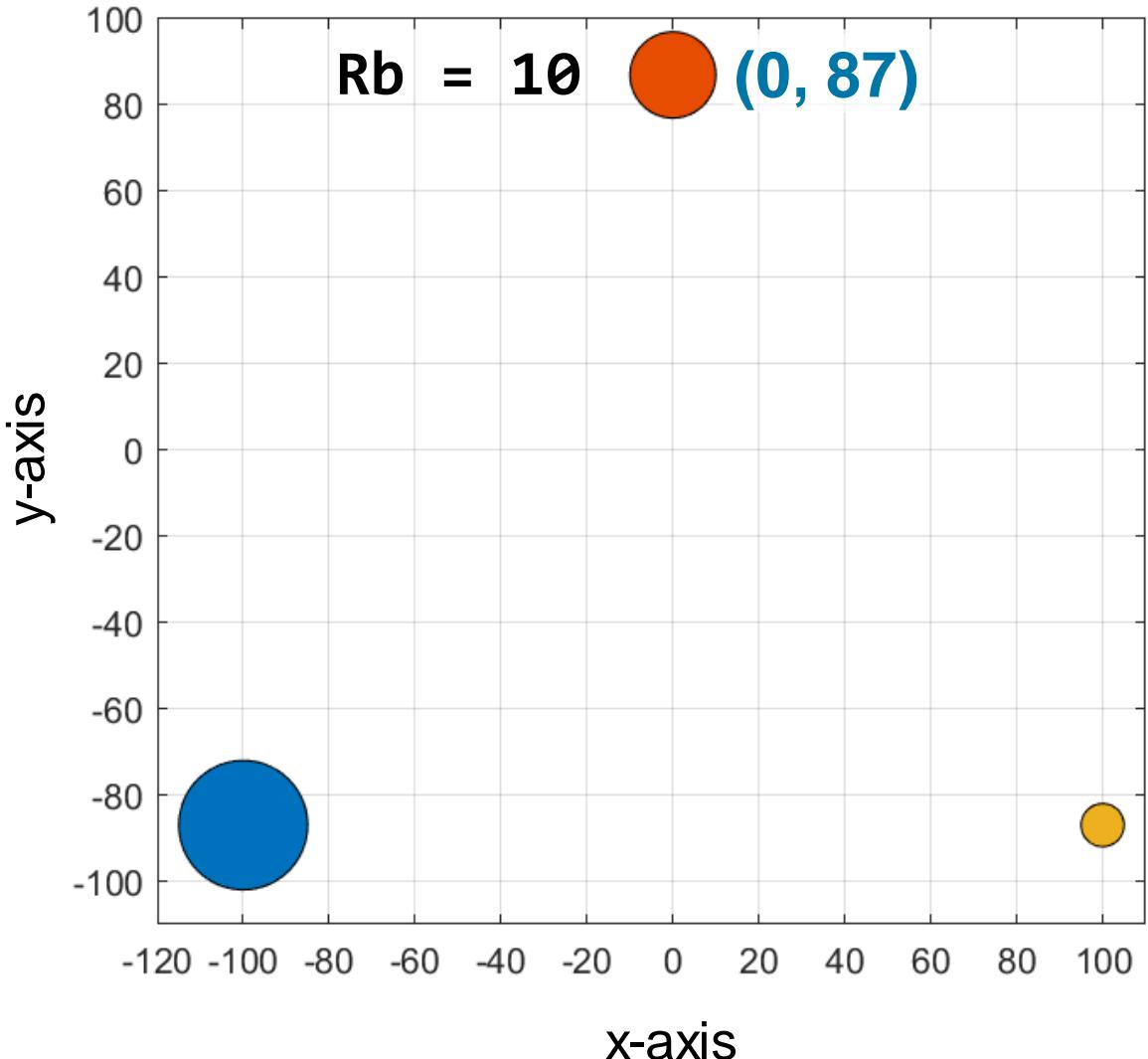
Body 1
(k = 1)

```
>> 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)



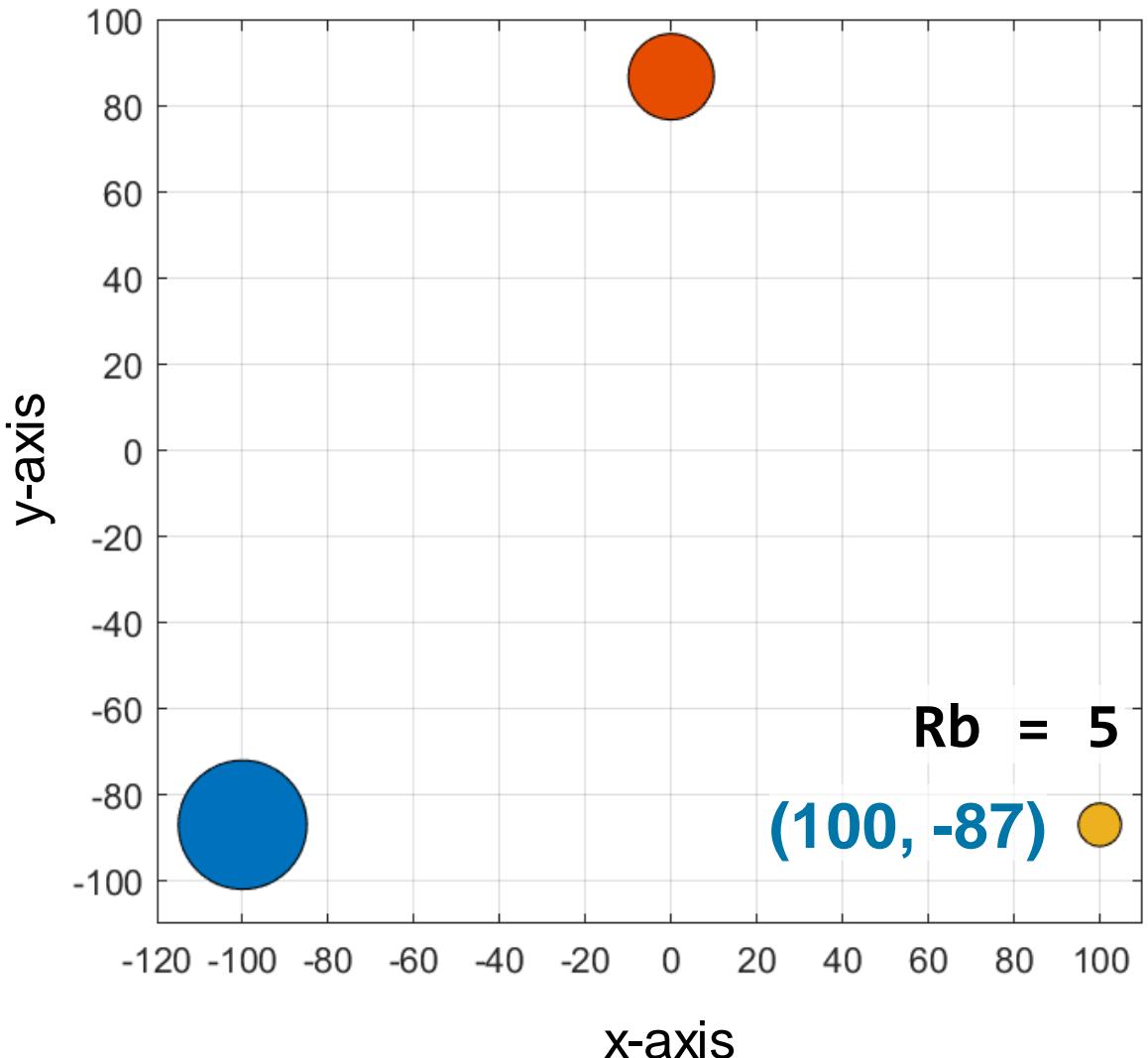
Body 2
($k = 2$)

```
>> 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

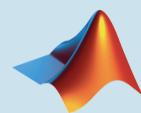
ang = 0:5:360; % angle, degrees
xc = cosd(ang); % circle x-points
yc = sind(ang); % circle y-points

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 Random Mission Map
 - 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 Map
- 5) Keeping Score

The International Space Station (ISS)

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

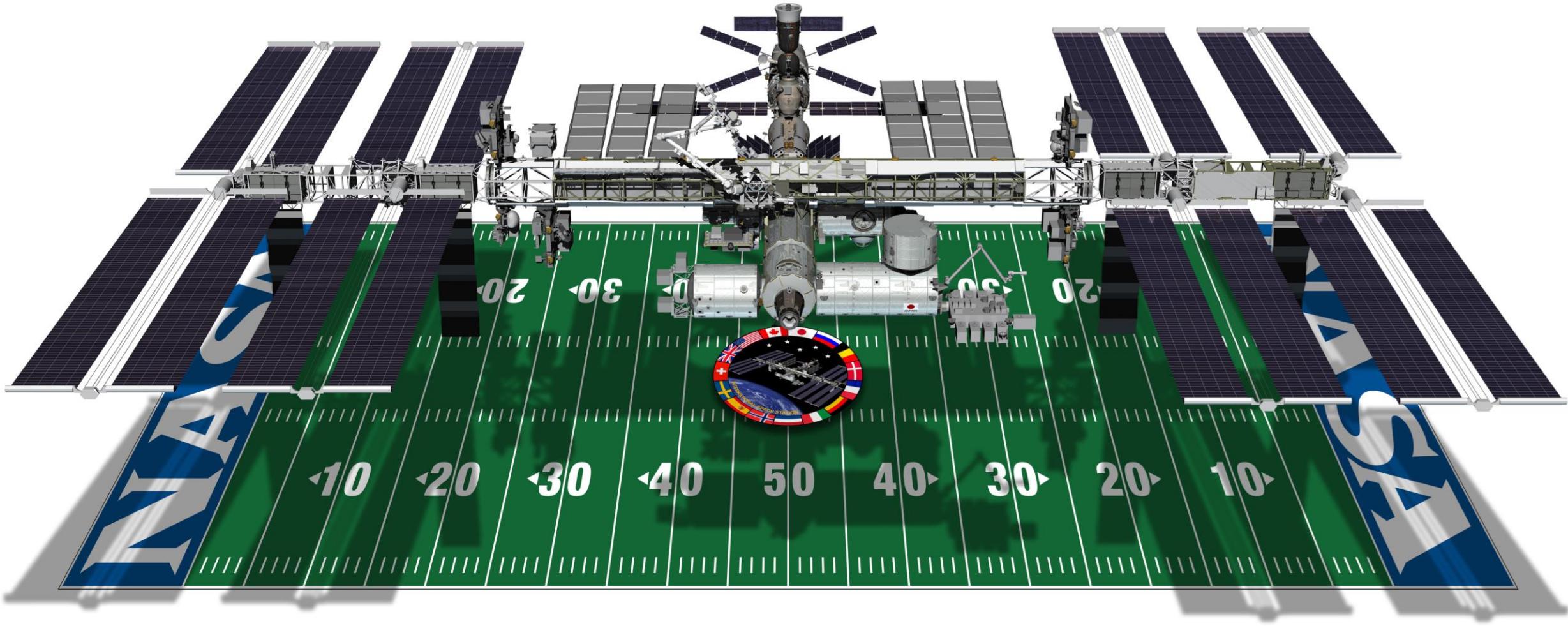


Image Credit: NASA

Spot the International Space Station from Your Town!

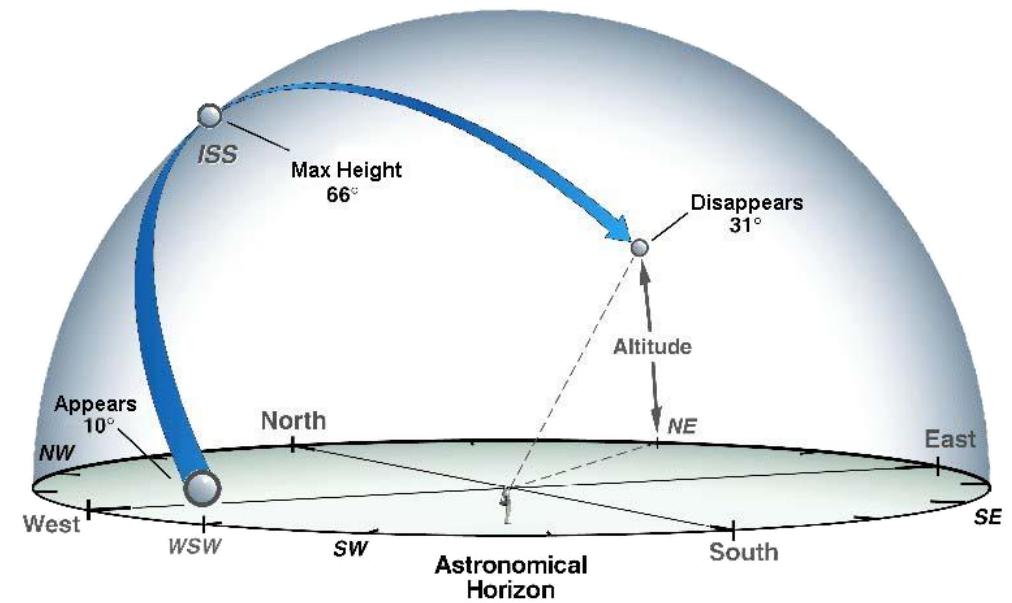
<https://spotthestation.nasa.gov>



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°
Appears: 10° above SW
Disappears: 43° 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?



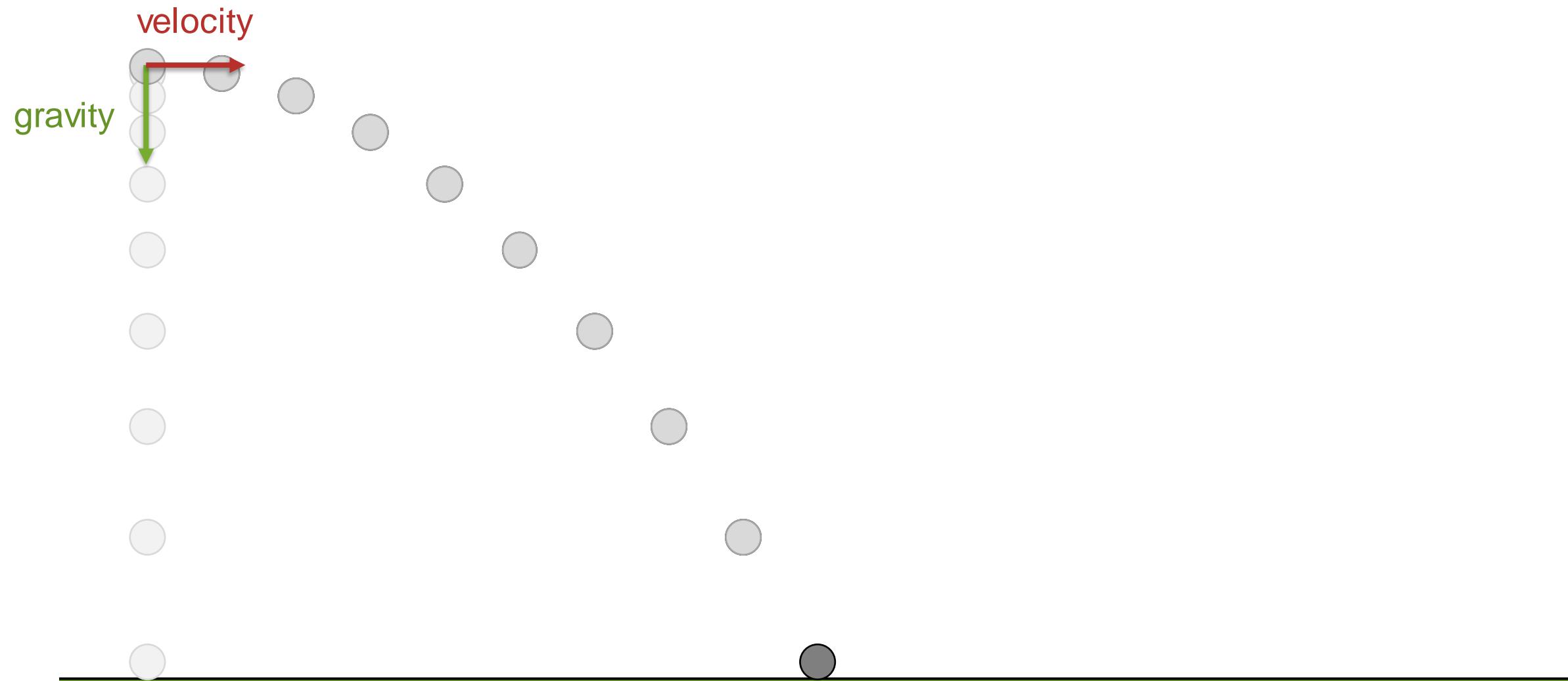
What will happen when this object is released with no velocity? Why?



What if we give the object a horizontal velocity?



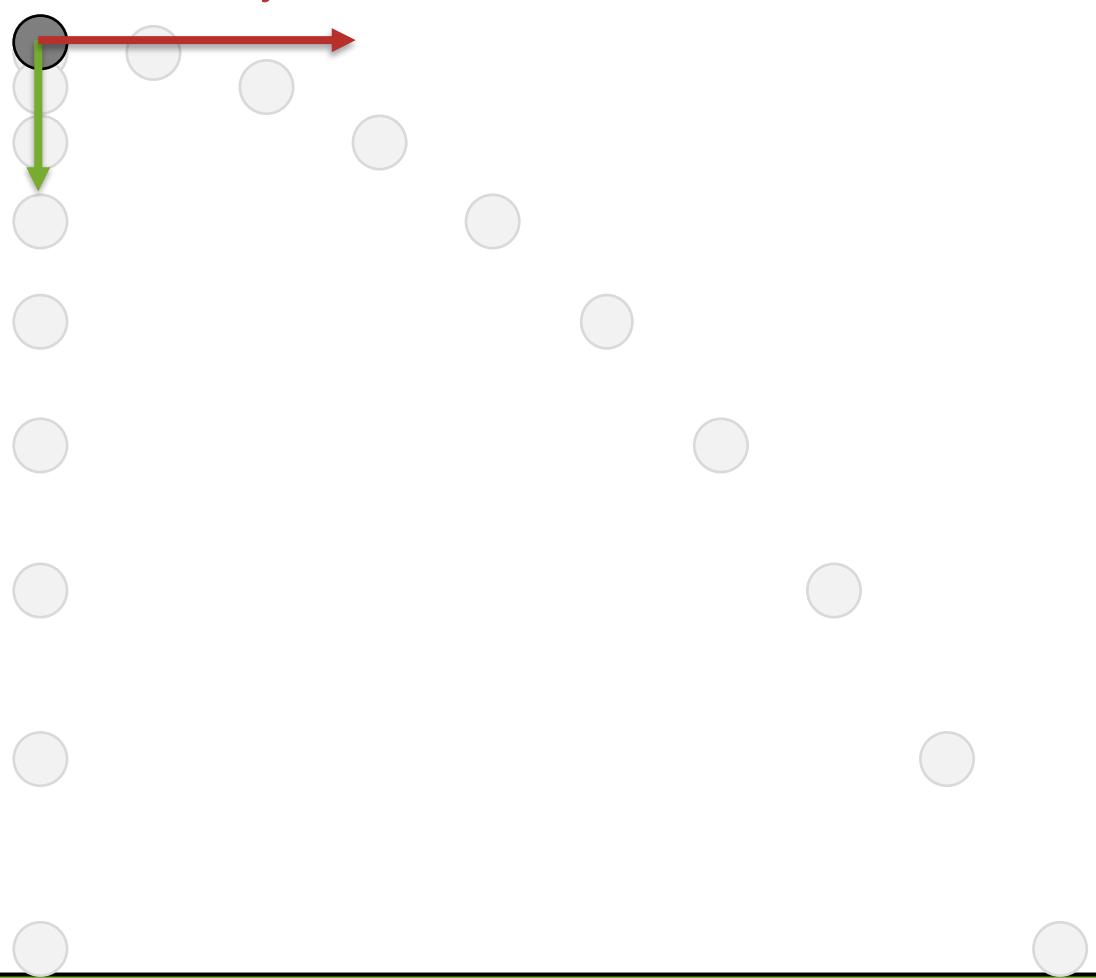
What if we give the object a horizontal velocity?



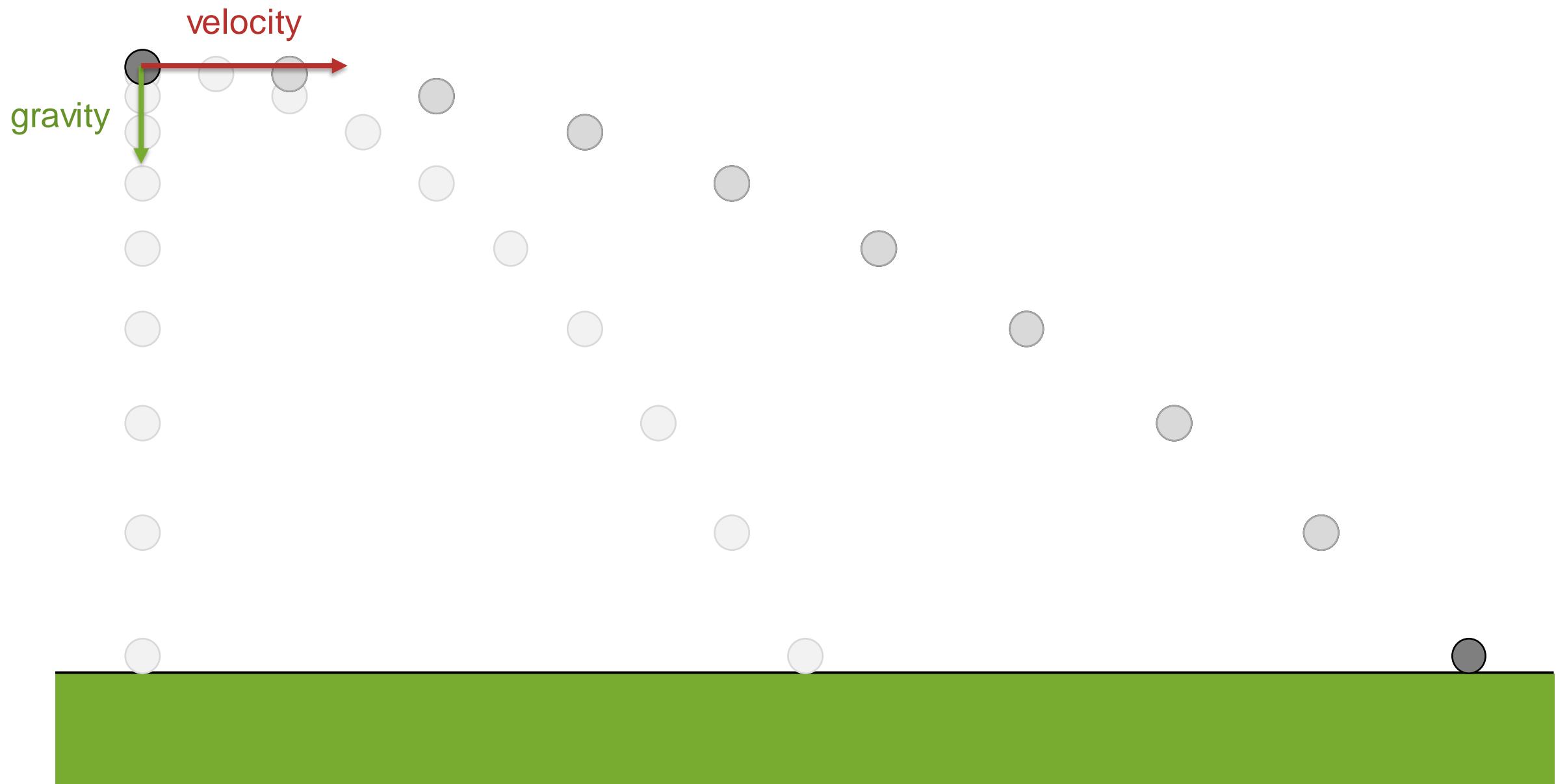
What if we give the object a bigger horizontal velocity?

velocity

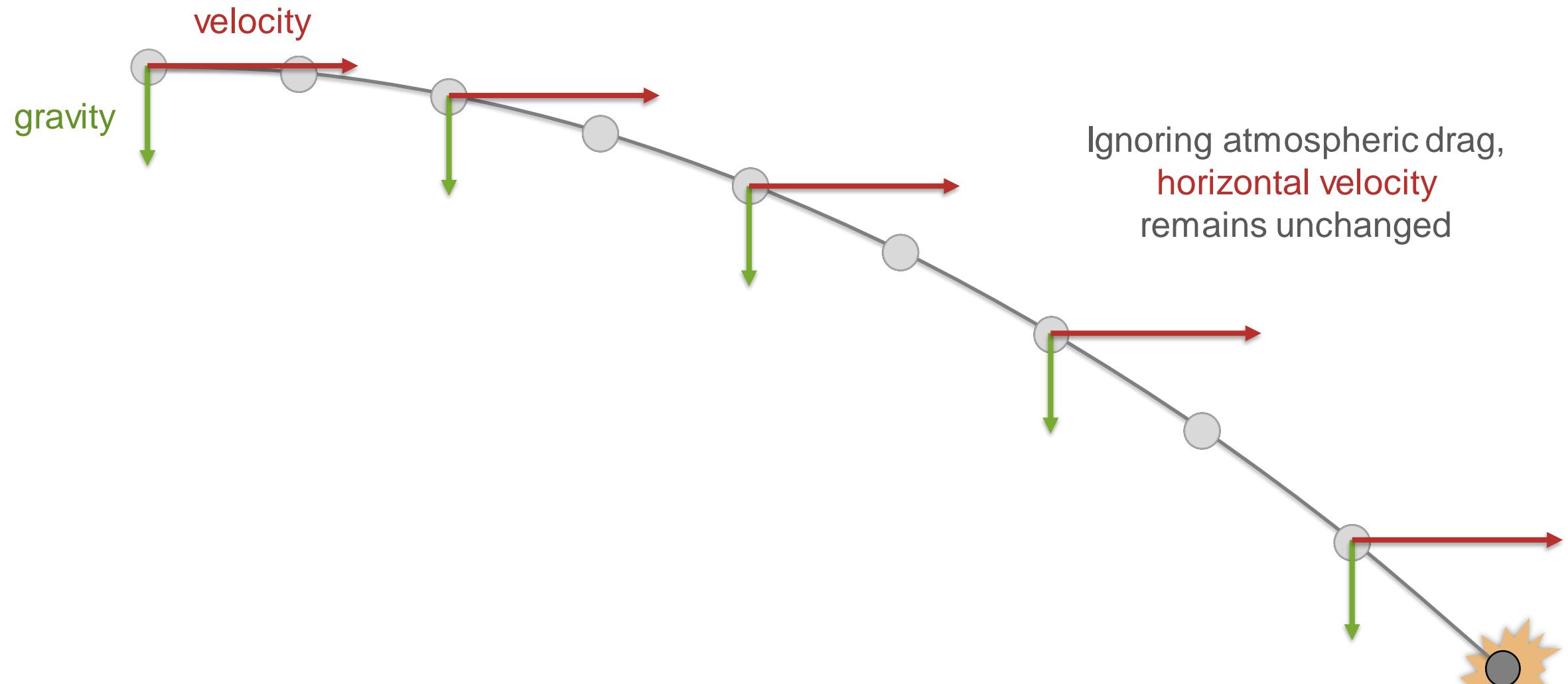
gravity



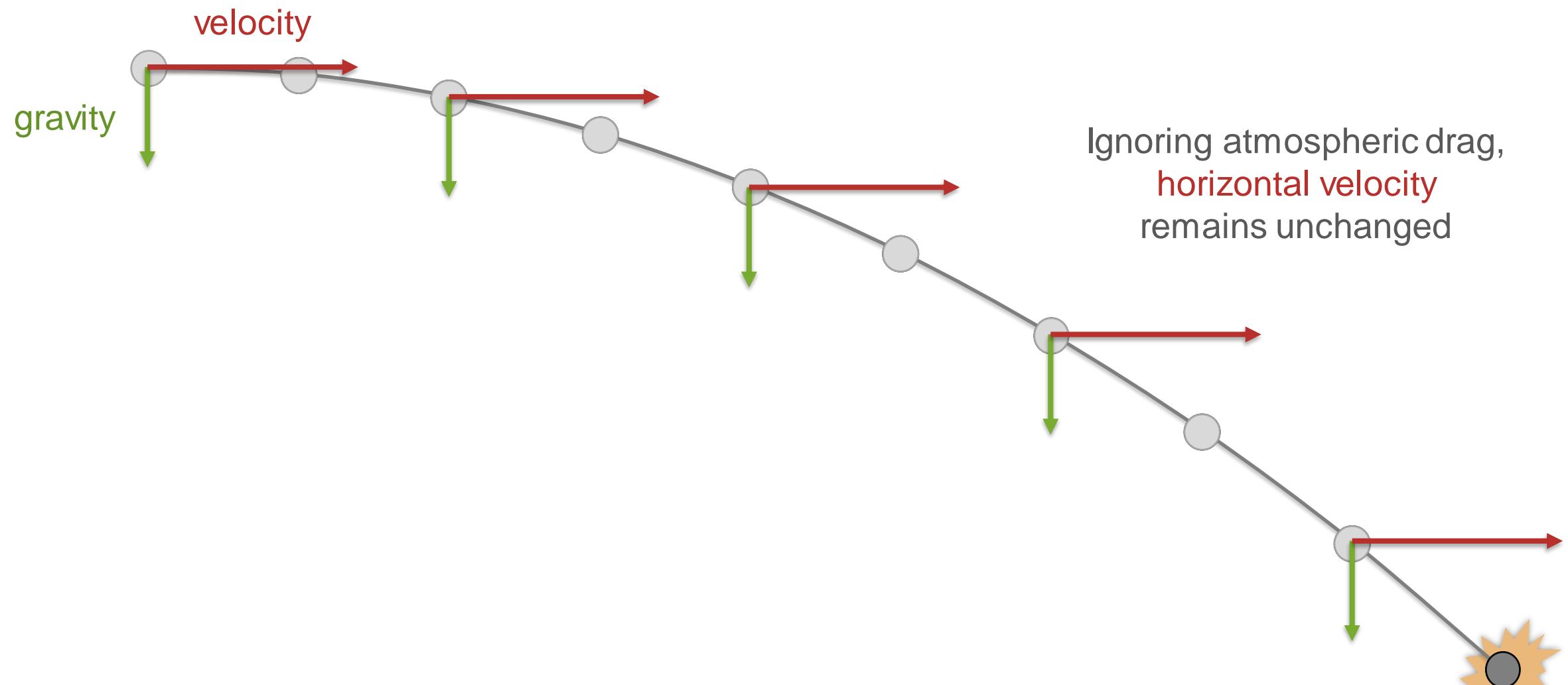
What if we give the object a bigger horizontal velocity?



Why does the object stop?

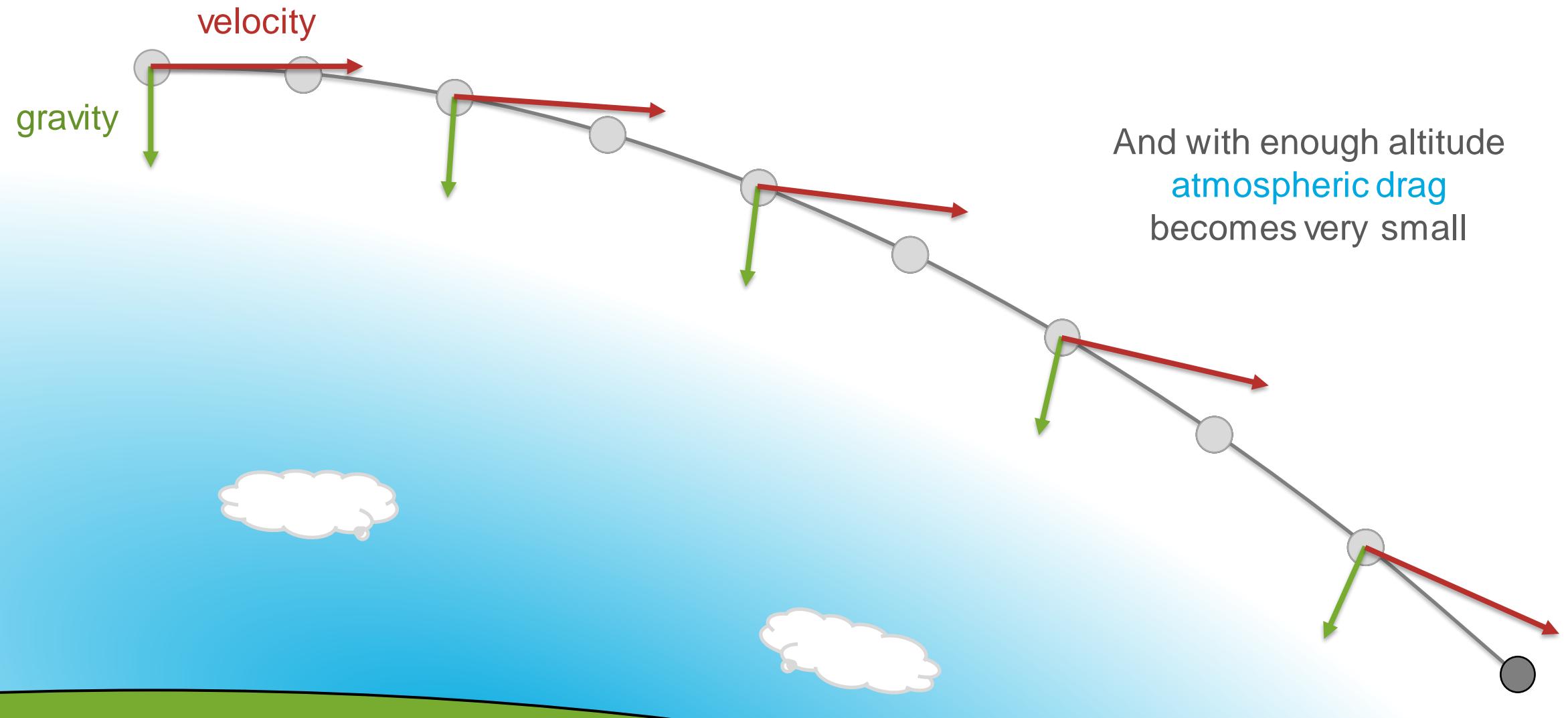


Why does the object stop? What happens to the surface further out?

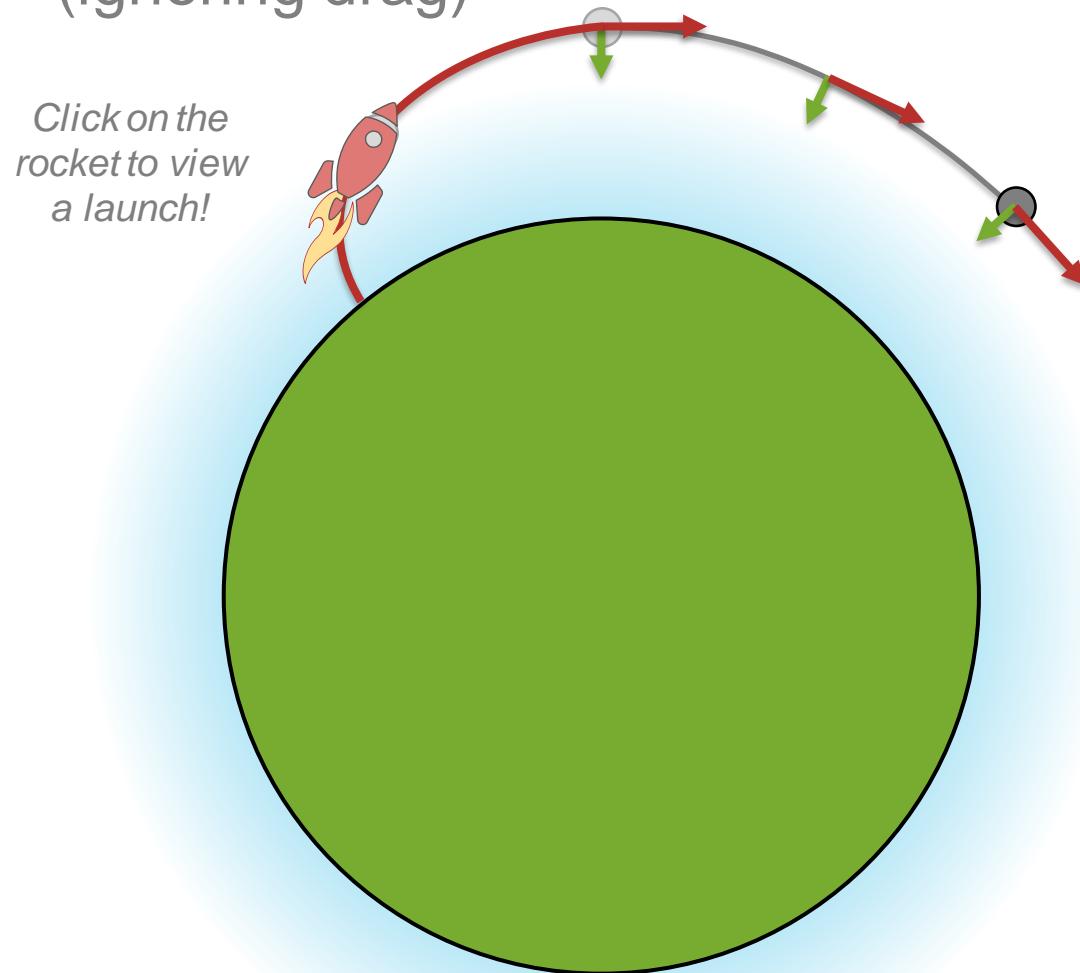


Is the surface flat?

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)

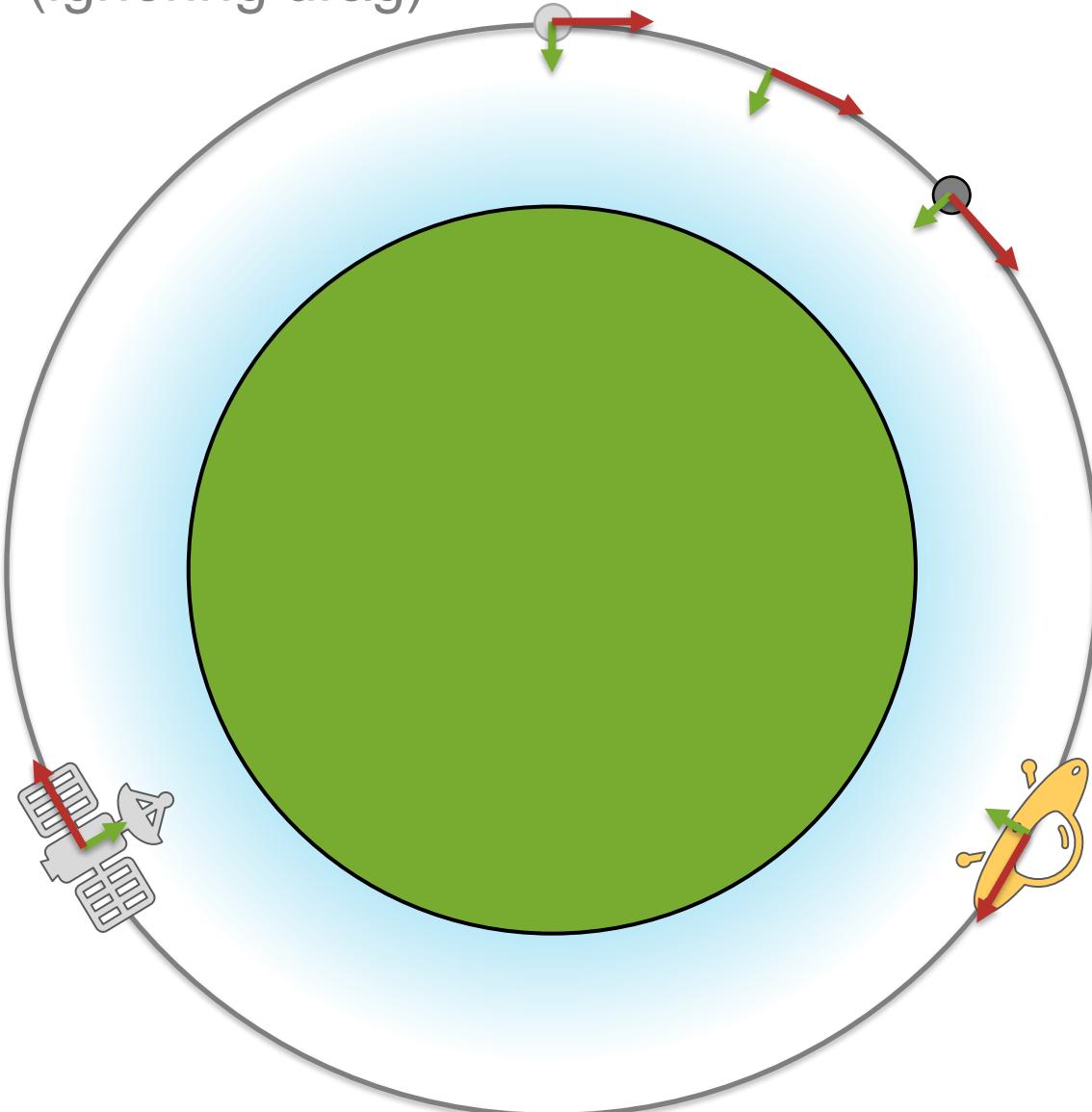


"Free fall" is the motion of an object acted upon solely by gravity (no drag)



"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)

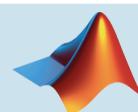


"Freeflying at Skydive Miami"
by Norcal21jg created 20 Sept 2010 (Public Domain)

>> run OrbitalVelocity

Course Lesson Plan

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



Launching Volleys

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

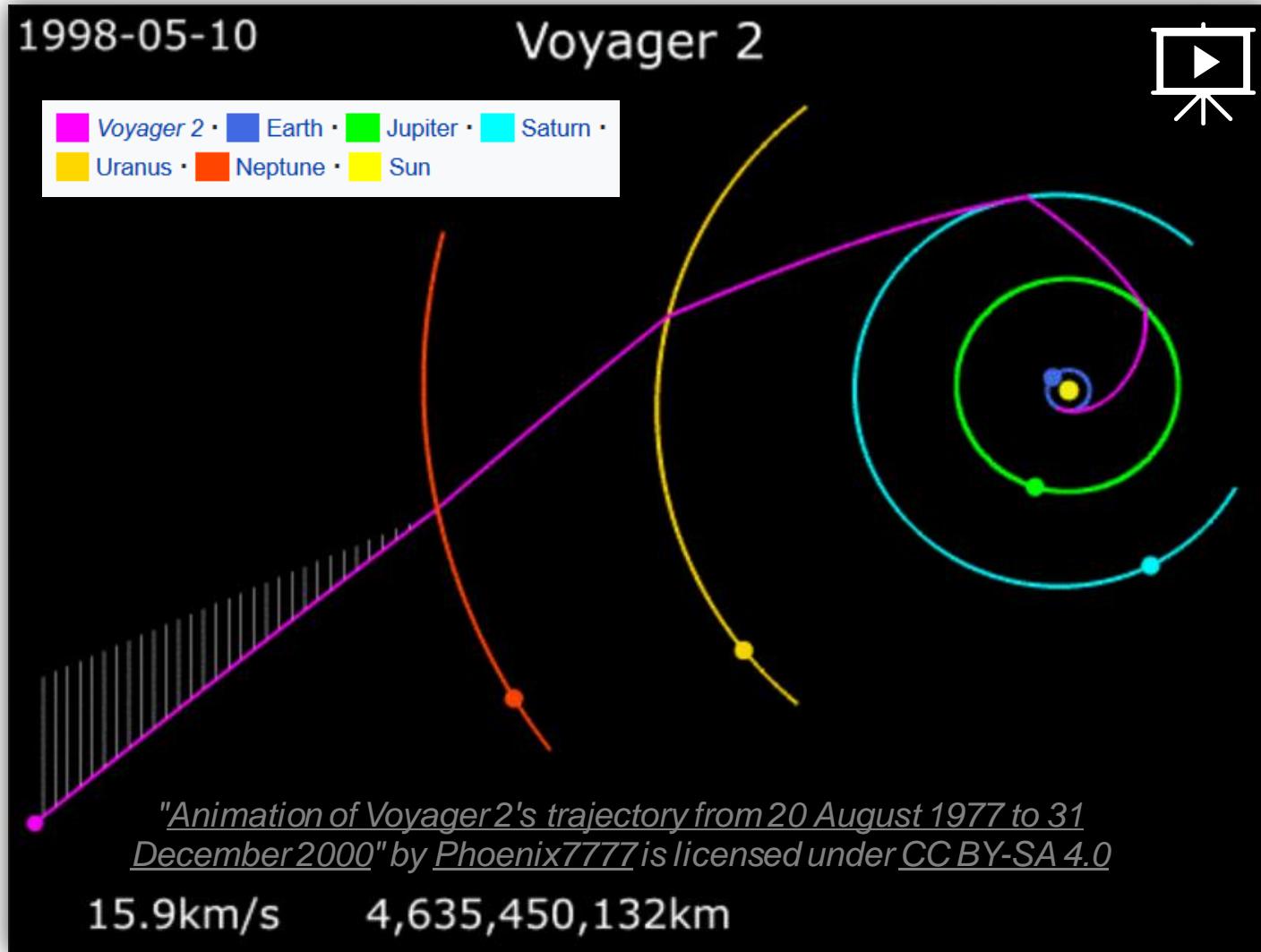
App Designer

- 1) UIAxes, Properties, & Functions
- 2) Sliders & Value Changing Callback
- 3) Push Button Callback
- 4) Generating Random Map
- 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

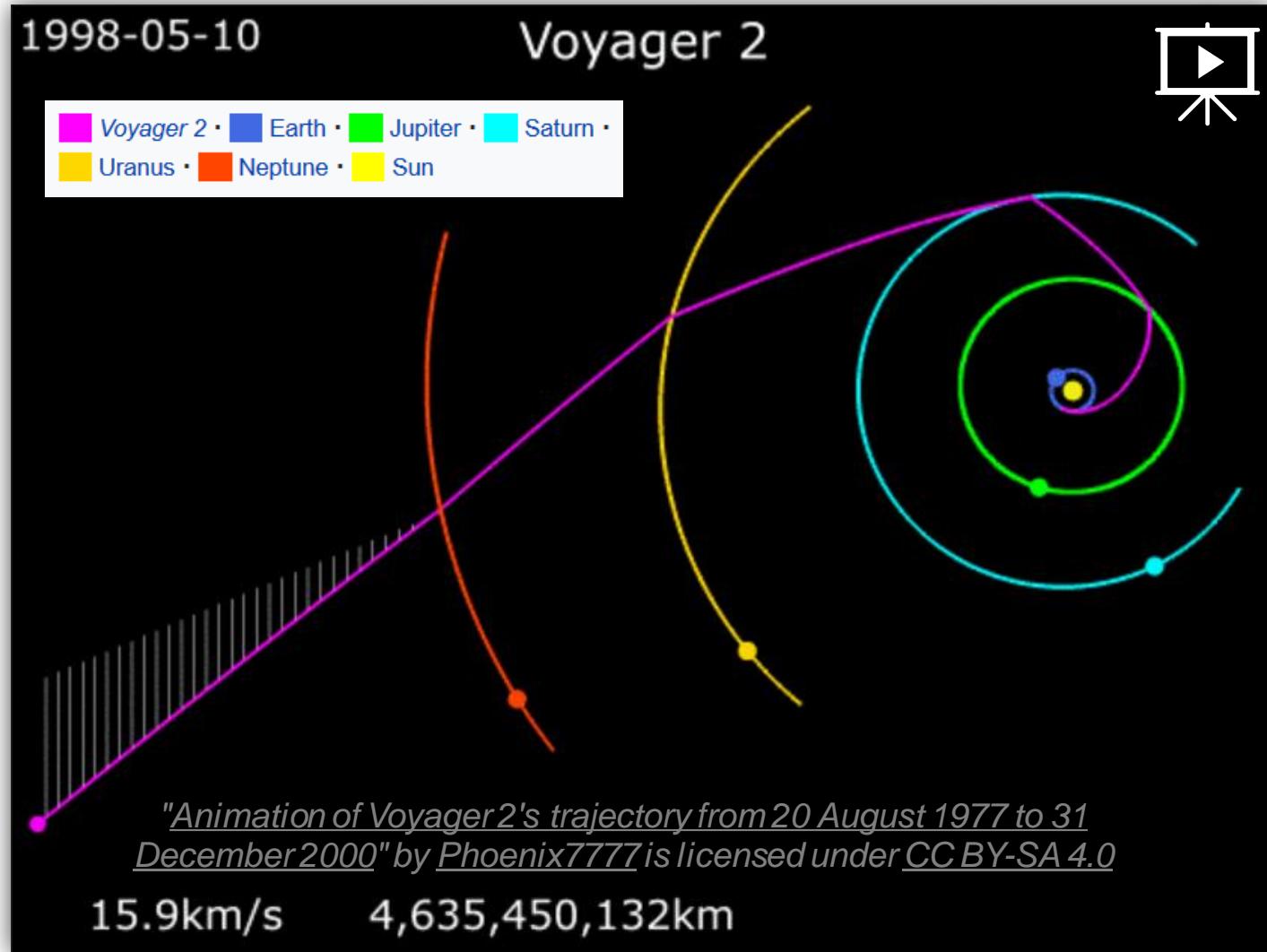


Observations...



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...

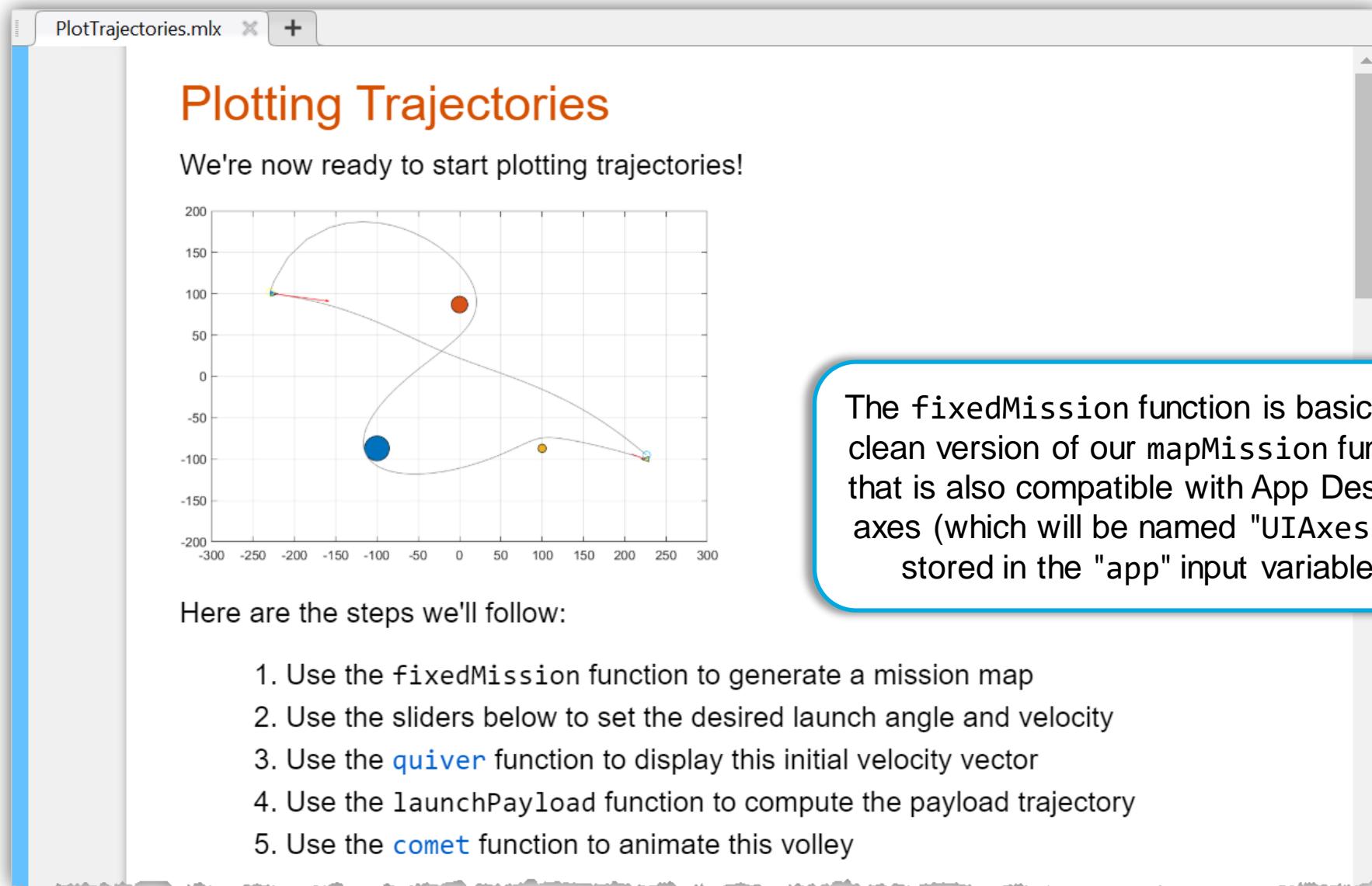
- Gravity of each body pulls on all the others (e.g. tides)
- For simplicity, we'll ignore these interactions and "lock" each body in place

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,:))
```

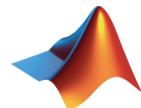
```
app.xb = [-100 0 100]; % body x-pos  
app.yb = [ -87 87 -87]; % body y-pos  
app.Rb = [ 15 10 5]; % body sizes  
cb = lines(3); % body colors  
  
ang = 0:5:360; % angle, degrees  
xc = cosd(ang); % circle x-points  
yc = sind(ang); % circle y-points  
for k = 1:3  
    fill(ax,app.xb(k)+app.Rb(k)*xc,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



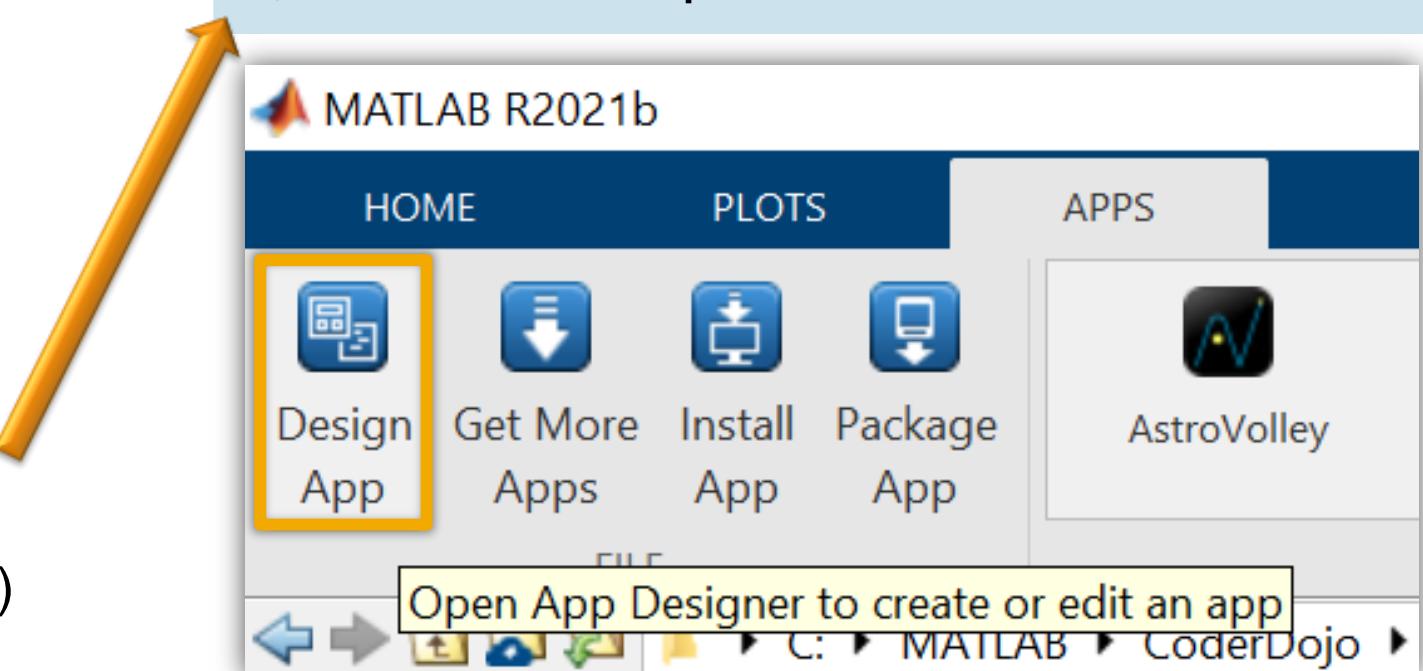
Course Lesson Plan

- Installing and Playing AstroVolley
- Plotting Ships (Triangles)
- Plotting Grav. Bodies (Circles)
- Gravity and Orbits
- Launching Volleys
 - Creating a Random Mission Map
 - 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.

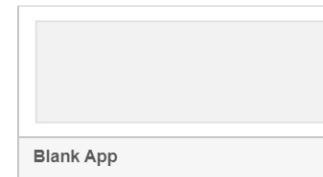
[Start Tutorial](#)[Open...](#)

Recent Apps

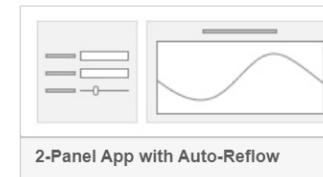
No Recent Apps

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

▼ New



Blank App

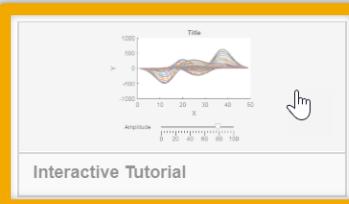


2-Panel App with Auto-Reflow

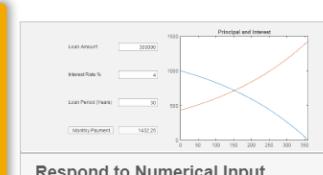


3-Panel App with Auto-Reflow

▼ Examples: General



Interactive Tutorial



Respond to Numerical Input



Respond to User Selections



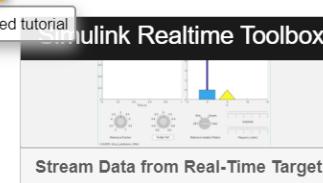
Embed HTML Content



Lay Out Controls in a Grid

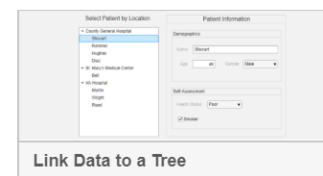


Display and Animate Flight Status



Stream Data from Real-Time Target

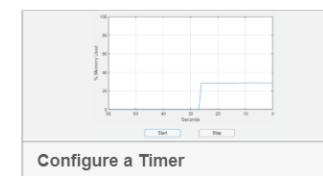
▼ Examples: Programming Tasks



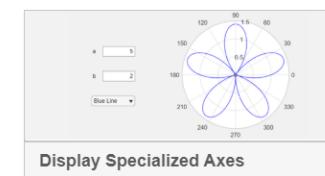
Link Data to a Tree



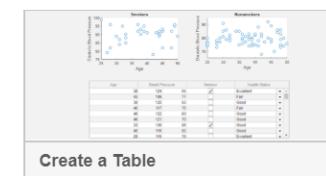
Analyze an Image



Configure a Timer



Display Specialized Axes



Create a Table

DESIGNER

CANVAS

VIEW



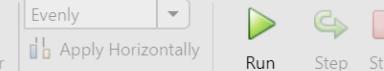
Save Convert



ALIGN



ARRANGE



SPACE



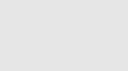
Run



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



Slider



Spinner



State Button



Table



Text Area



Toggle Button

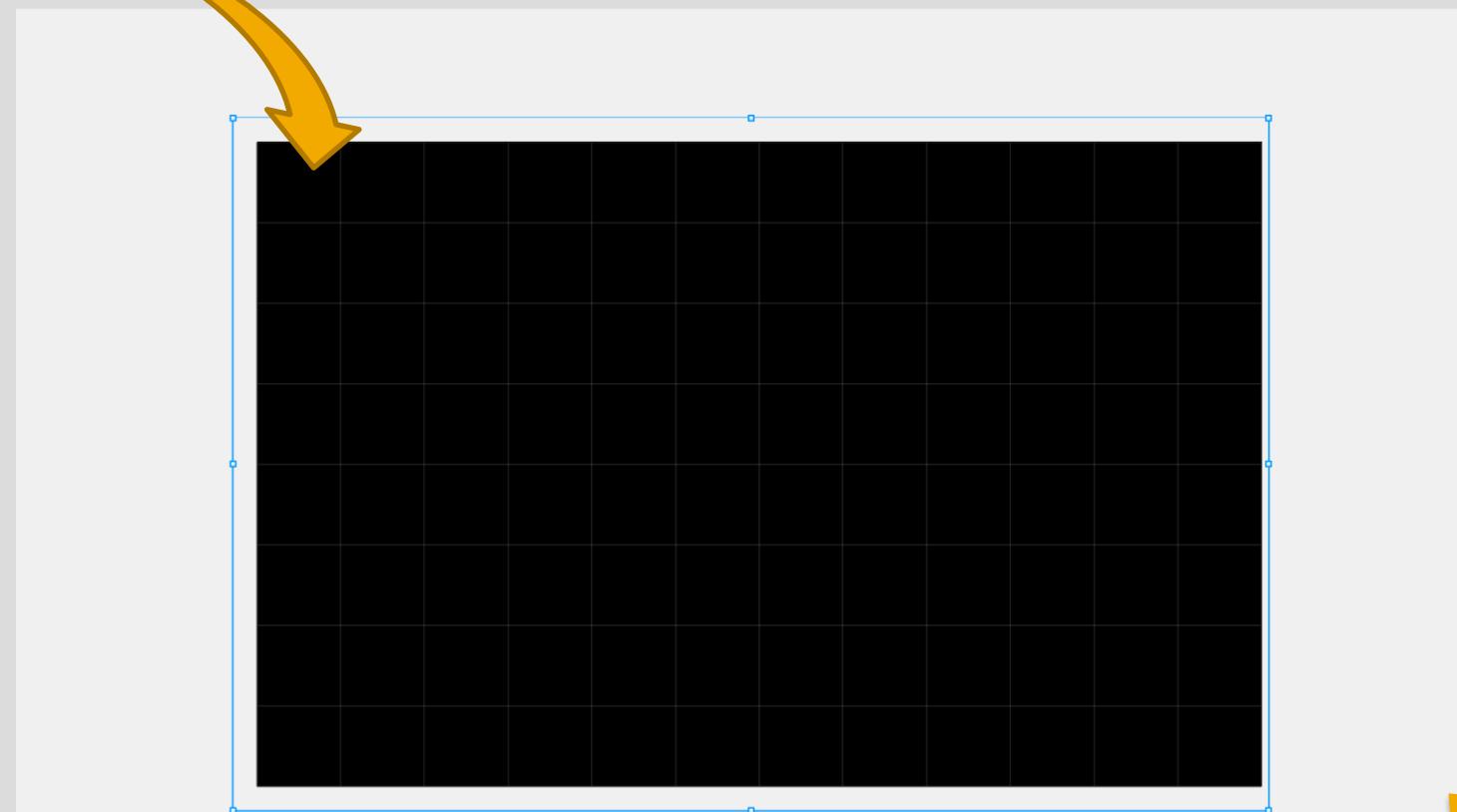


Tree



Tree (Check Box)

CONTAINERS



Design View Code View

Component Browser

Search

MyAstroVolley

app.UIFigure

app.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

1.00,1.00,

BOX STYLING

Color

0.00,0.00,



DESIGNER

CANVAS

VIEW



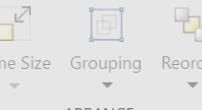
Save

Convert

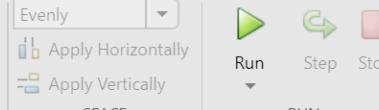
FILE



ALIGN



ARRANGE



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

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

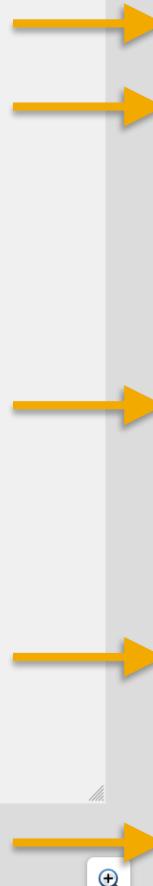
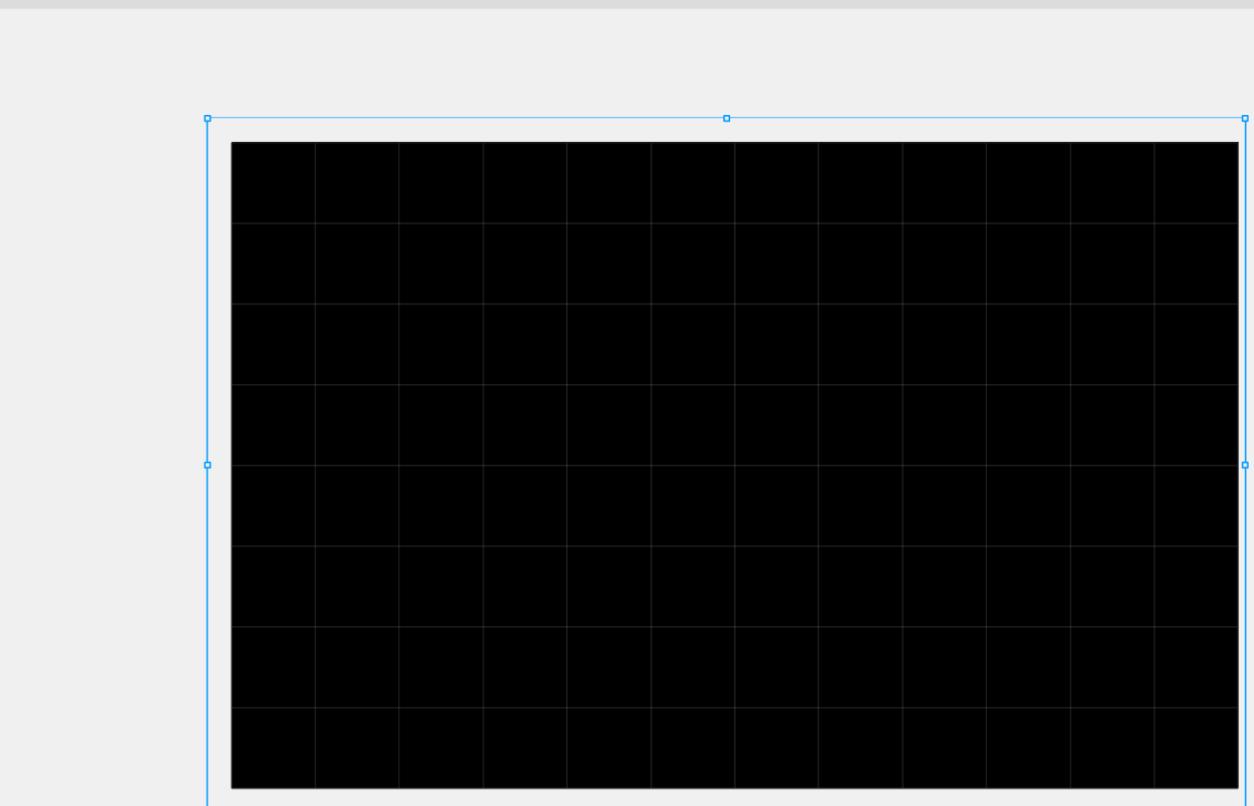
ZGrid

GridColor

Color

0.00,0.00,0.00

0.00,0.00,0.00



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

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 for a project named "MyAstroVolley.mlapp". The "Code View" tab is selected. In the center canvas, there is a large black rectangular area with a grid pattern, which appears to be a placeholder or a specific component. The left panel, "Component Library", contains a grid of common UI components: 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 right panel, "Component Browser", shows the project structure with "MyAstroVolley" expanded to show "app.UIFigure" and "app.UIAxes". The "Axes" section of the browser includes fields for Title.String, XLabel.String, YLabel.String, ZLabel.String, Subtitle.String, and TitleHorizontalAlignment (set to center). It also includes sections for FONT (FontName set to Helvetica, FontWeight set to Bold, FontSize set to 12) and TICKS (X and Y tick ranges). The "Code View" tab is highlighted with a yellow border.

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
Public Property

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');

Design View Code View

Component Browser

Search

MyAstroVolley

app.UIFigure
app.UIAxes

App | Callbacks

Search

SHARING DETAILS

CODE OPTIONS

This screenshot shows the MATLAB App Designer interface. The central area is the code editor for the file 'MyAstroVolley.mlapp'. The code defines a class 'MyAstroVolley' that inherits from 'matlab.apps.AppBase'. It includes sections for properties (both private and public), methods (private and public), and callbacks. A specific section of the code, which defines public properties for ship and body positions and sizes, is highlighted with a yellow box. To the left of the code editor is the 'App Layout' pane, which contains a black placeholder for the app's UI. On the right side, there are two panes: 'Component Browser' and 'App | Callbacks'. The 'Component Browser' pane shows the app's structure, including its main figure and axes. The 'App | Callbacks' pane provides details about the app's sharing and code options.

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

```
classdef MyAstroVolley < matlab.apps.AppBase

    % Properties that correspond to app components
    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

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

        % 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');

            end

        end

    end

end
```

DESIGNER

EDITOR

VIEW



Save



Print



Compare



Go To



Bookmark



Callback



Function



Property



App Input



Arguments



Text



Comment



%



%



%



Indent



Step



Stop

CODE

RUN



Run

MyAstroVolley.mlapp

Code Browser

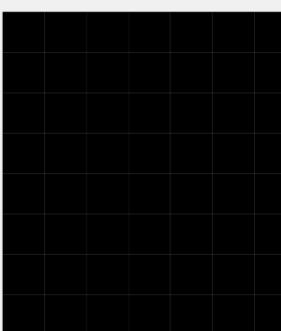
Callbacks | Functions | Properties

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

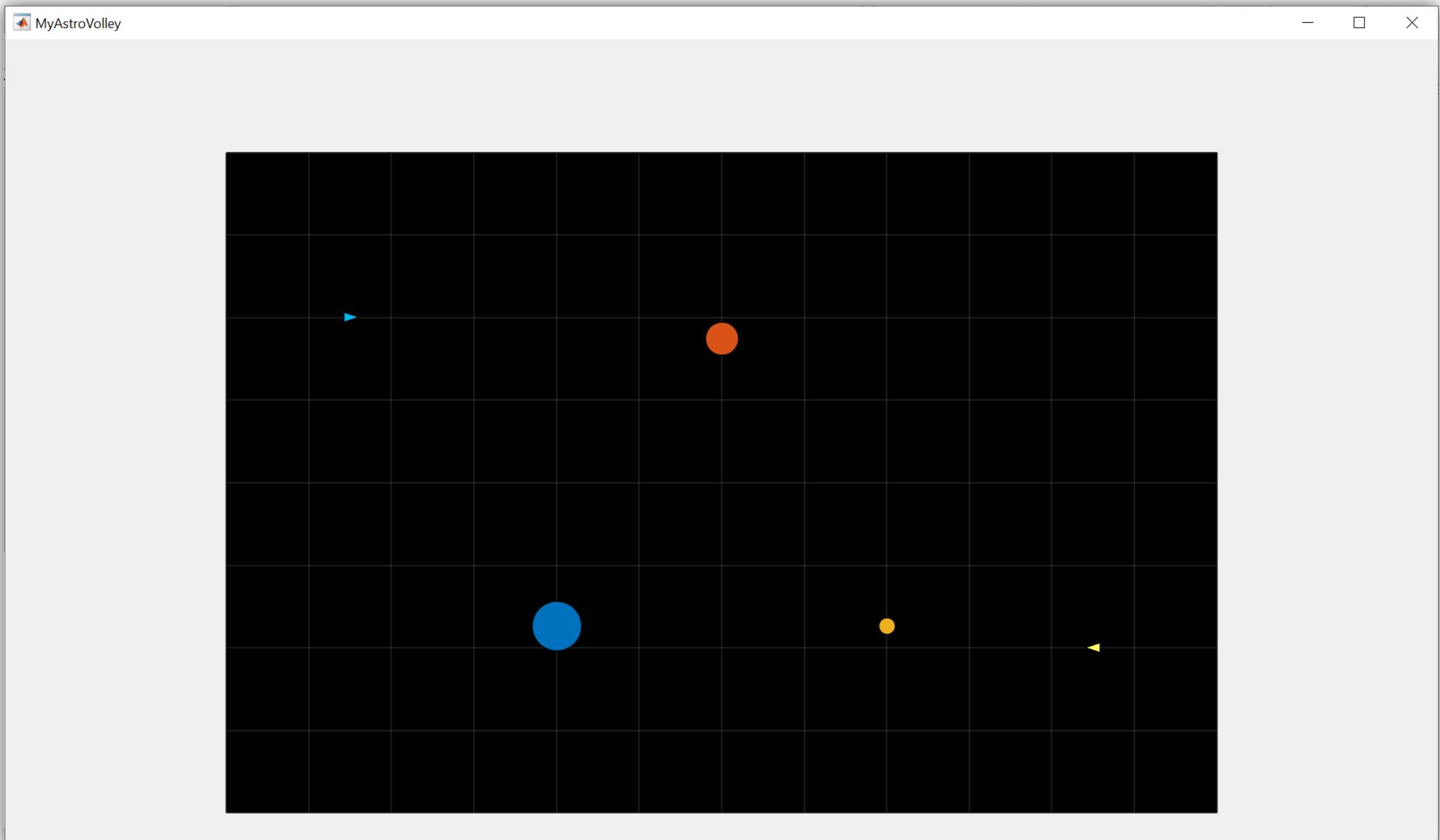
app.UIFigure
app.UIAxes

App | Callbacks

Search

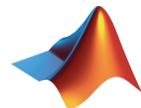
► SHARING DETAILS

► CODE OPTIONS



Course Lesson Plan

- Installing and Playing AstroVolley
- Plotting Ships (Triangles)
- Plotting Grav. Bodies (Circles)
- Gravity and Orbits
- Launching Volleys
 - Creating a Random Mission Map
 - 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 Map
- 5) Keeping Score



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

DESIGNER

CANVAS

VIEW



Save

Convert

FILE

ALIGN

ARRANGE

Evenly
Same Size
Grouping
Reorder

Run
Space
RUN



MyAstroVolley.mlapp

Component Library

Search



COMMON



Axes



Button



Check Box



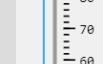
Date Picker



Drop Down



Edit Field (Numeric)



Hyperlink



Edit Field (Text)



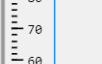
HTML



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.AngleSlider1

app.UIAxes

Slider | Callbacks

Search

Label Angle(deg)

SLIDER

Value

Limits

Orientation

TICKS

MajorTicks

MajorTickLabels

MinorTicks

FONT

FontName

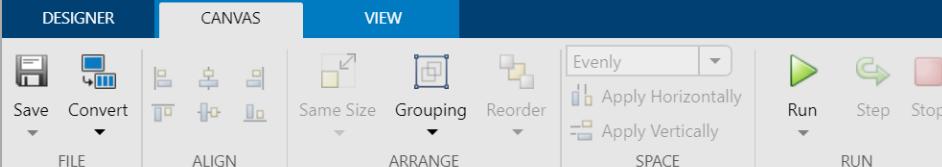
FontSize

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



Evenly

Run

Apply Horizontally

Step

Apply Vertically

Stop



SPACE



RUN

Design View Code View

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

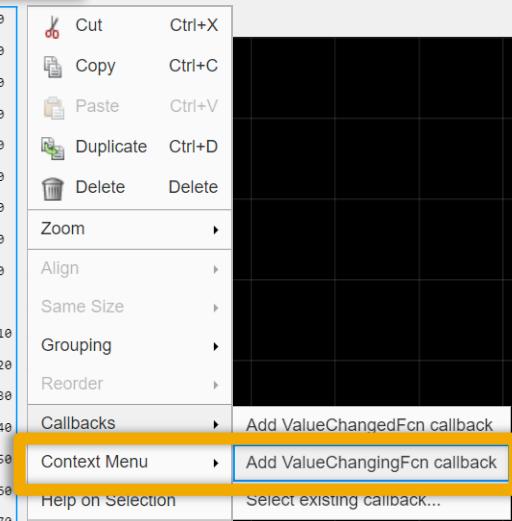


Tree



Tree (Check Box)

Right-Click



Component Browser

Search

MyAstroVolley

app.UIFigure

app.AngleSlider1

app.UIAxes

Slider | Callbacks

Search

Label Angle(deg)

SLIDER

Value

Limits

Orientation

TICKS

MajorTicks

MajorTickLabels

MinorTicks

FONT

FontName

FontSize

FontWeight

FontAngle

FontColor

INTERACTIVITY

DESIGNER EDITOR VIEW

FILE NAVIGATE INSERT CODE RUN

MyAstroVolley.mlapp

Code Browser

Callbacks | Functions | Properties

Search

startupFcn

AngleSlider1ValueChanging

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:

Design View

Code View

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 end
20
21
22 % Callbacks that handle component events
23 methods (Access = private)
24
25 % Code that executes after component creation
26 function startupFcn(app)
27 fixedMission(app);
28 end
29
30 % Value changing function: AngleSlider1
31 function AngleSlider1ValueChanging(app, event)
32 ang = event.Value;
33 vel = app.VelocitySlider1.Value;
34 vx = vel*cosd(ang);
35 vy = vel*sind(ang);
36 delete(app.hq)
37 app.hq = quiver(app.UIAxes, app.xs(1), app.ys(1), vx, vy, "Color", "red");
38 end
39
40 end
41
42 % Component initialization
43 methods (Access = private)
44
45 % Create UIFigure and components
46 function createComponents(app)

AngleSlider1 (UIAxes)

DESIGNER

CANVAS

VIEW



Save

Convert

FILE

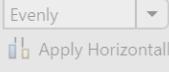


ALIGN



Grouping

Reorder



Evenly

Same Size

Grouping

Reorder



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)



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.VelocitySlider1

app.AngleSlider1

app.UIAxes

Slider | Callbacks

Search

Label Velocity,(%)

SLIDER

Value

Limits

Orientation

TICKS

MajorTicks

MajorTickLabels

MinorTicks

FONT

FontName

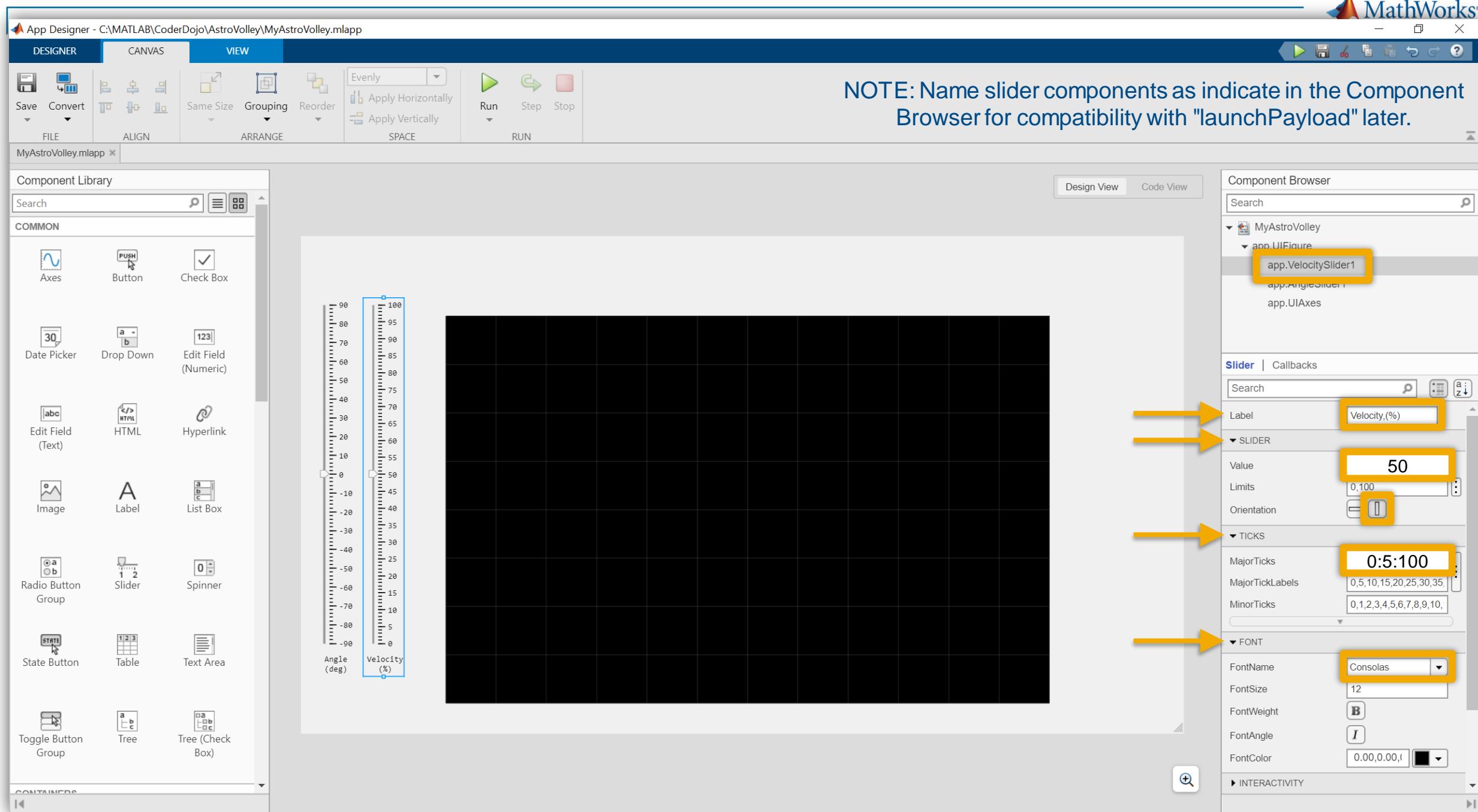
FontSize

FontWeight

FontAngle

FontColor

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



DESIGNER

CANVAS

VIEW



Save

Convert

FILE



ALIGN



Same Size

Grouping

Reorder



Run

Step

Stop

SPACE

RUN

MyAstroVolley.mlapp

Design View

Code View

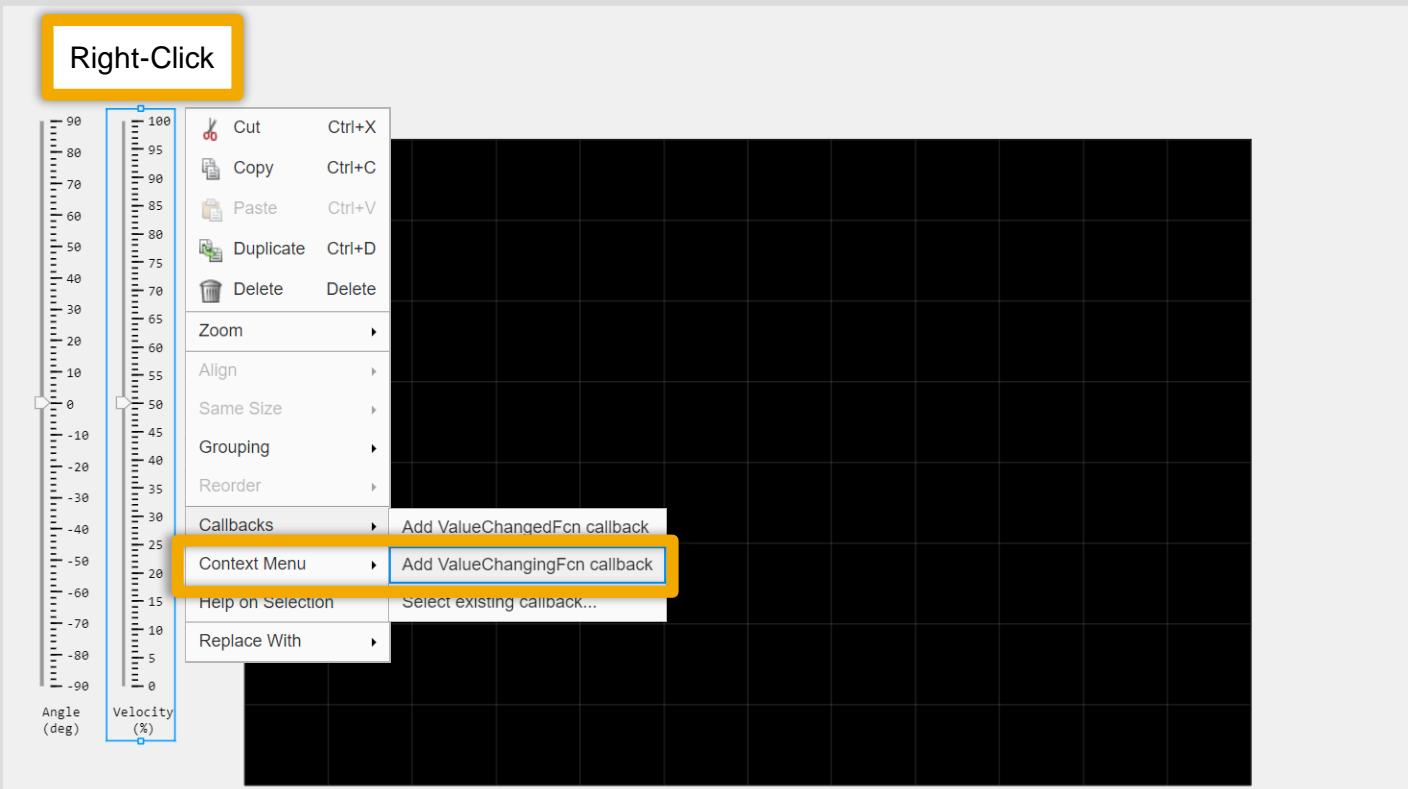
Component Library

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.VelocitySlider1
- app.AngleSlider1
- app.UIAxes

Slider | Callbacks

Search

Label Velocity,(%)

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 **B**

FontAngle *I*

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

CODE

RUN

MyAstroVolley.mlapp

Code Browser

Callbacks | Functions | Properties

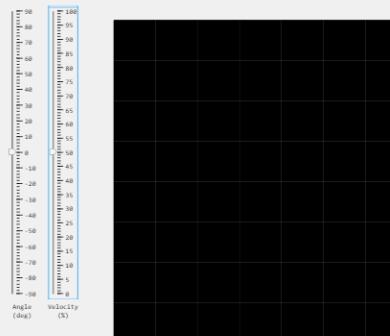
Search

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
44     % Value changing function: VelocitySlider1
45     function VelocitySlider1ValueChanging(app, event)
46         ang = app.AngleSlider1.Value;
47         vel = event.Value; ←
48         vx = vel*cosd(ang);
49         vy = vel*sind(ang);
50         delete(app.hq)
51         app.hq = quiver(app.UIAxes,app.xs(1),app.ys(1),vx,vy,"Color","red");
52     end

```

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

Horizontal

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

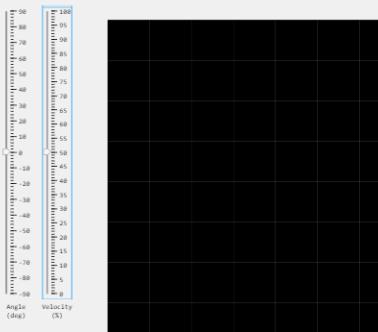


startupFcn

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

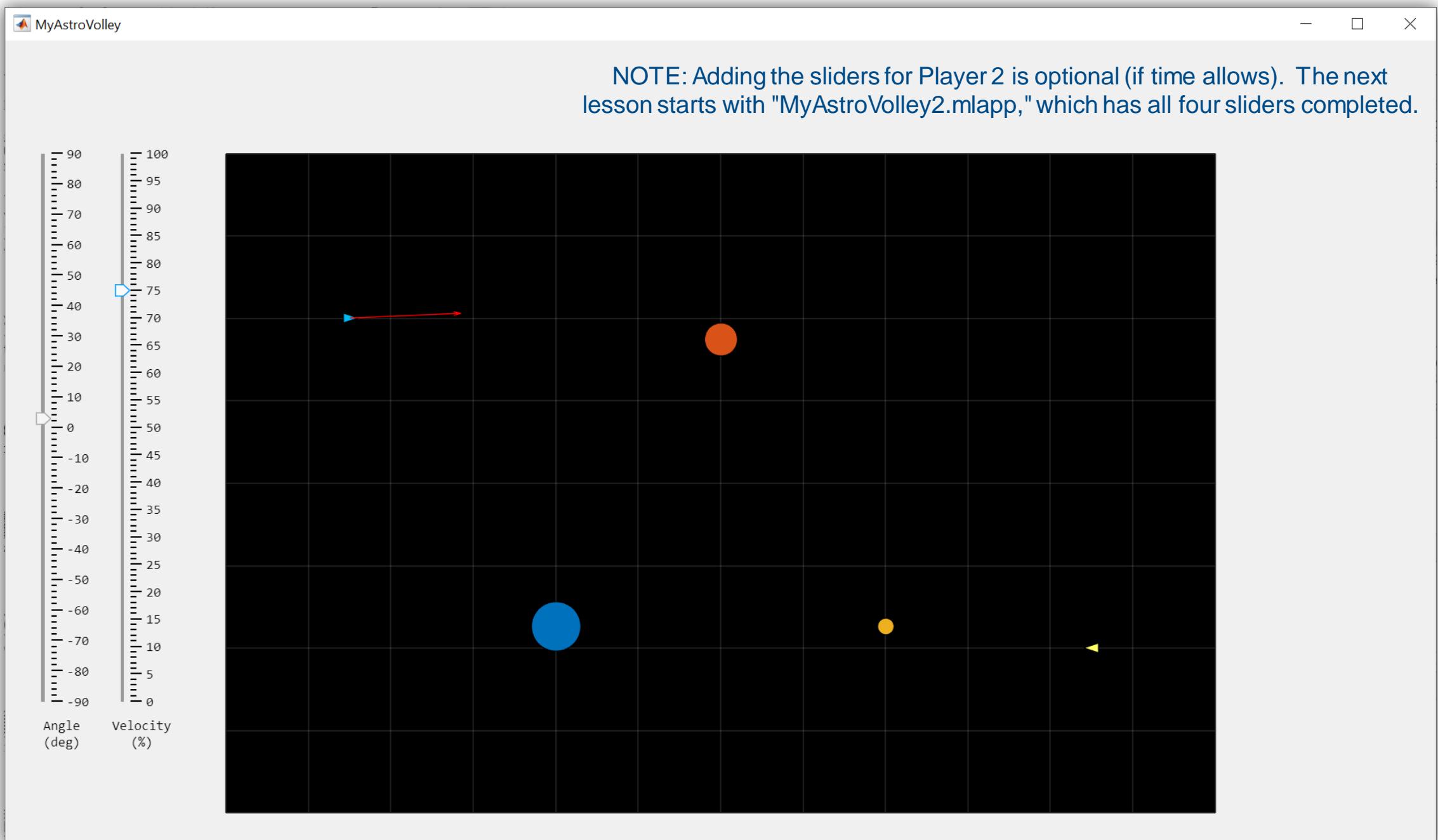
Consolas

FontSize

12

FontWeight





DESIGNER

CANVAS

VIEW



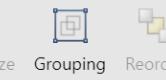
Save

Convert

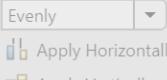
FILE



ALIGN



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



List Box



Radio Button



Slider



Spinner



State Button



Table



Text Area



Toggle Button



Tree



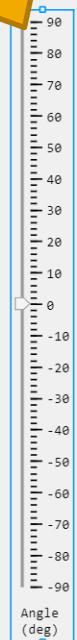
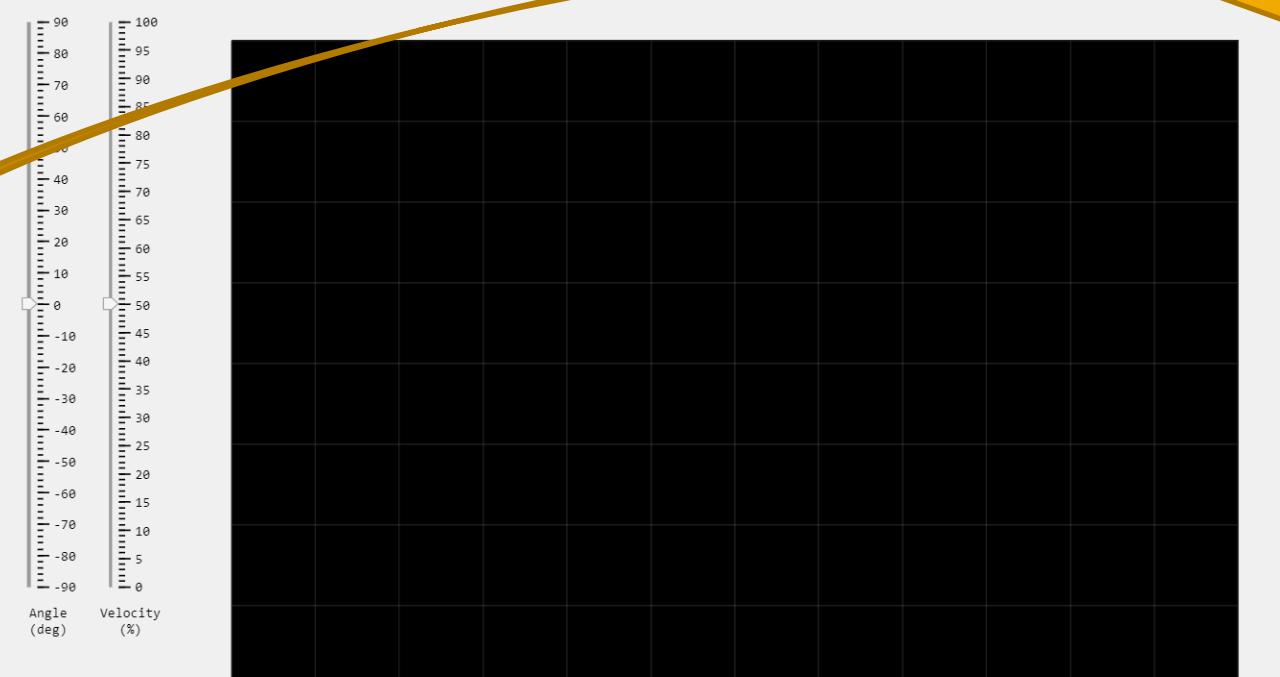
Tree (Check Box)

CONTAINERS



Design View

Code View



Component Browser

Search

MyAstroVolley

app.UIFigure

app.AngleSlider2

app.VelocitySlider1

app.AngleSlider1

app.UIAxes

Slider | Callbacks

Search

Label Angle(deg)

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.

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



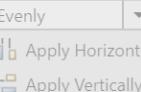
FILE



Same Size

Grouping

Reorder



Evenly

Apply Horizontally

Apply Vertically



Run

Step

Stop

SPACE

RUN

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

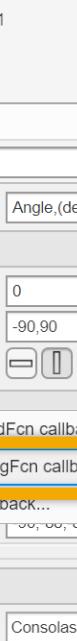
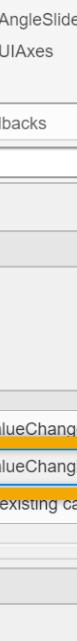
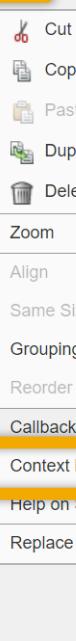
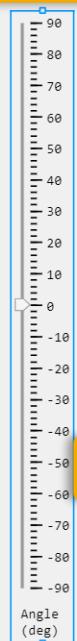


Tree (Check Box)

CONTAINERS



Right-Click



Component Browser

Search

MyAstroVolley

app.UIFigure

app.AngleSlider2

app.VelocitySlider1

app.AngleSlider1

app.UIAxes

Callbacks

r

Callba

k

Zoom

Align

Same Size

Grouping

Reorder

Callbacks

Add ValueChangedFcn callback

Context Menu

Add ValueChangingFcn callback

Help on Selection

Select existing callback...

Replace With

FON

FontName

FontSize

FontWeight

FontAngle

FontColor

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

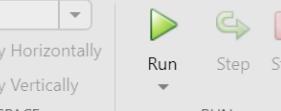
ALIGN



Same Size

Grouping

Reorder



Evenly

SPACE

Run

Step

Stop

RUN

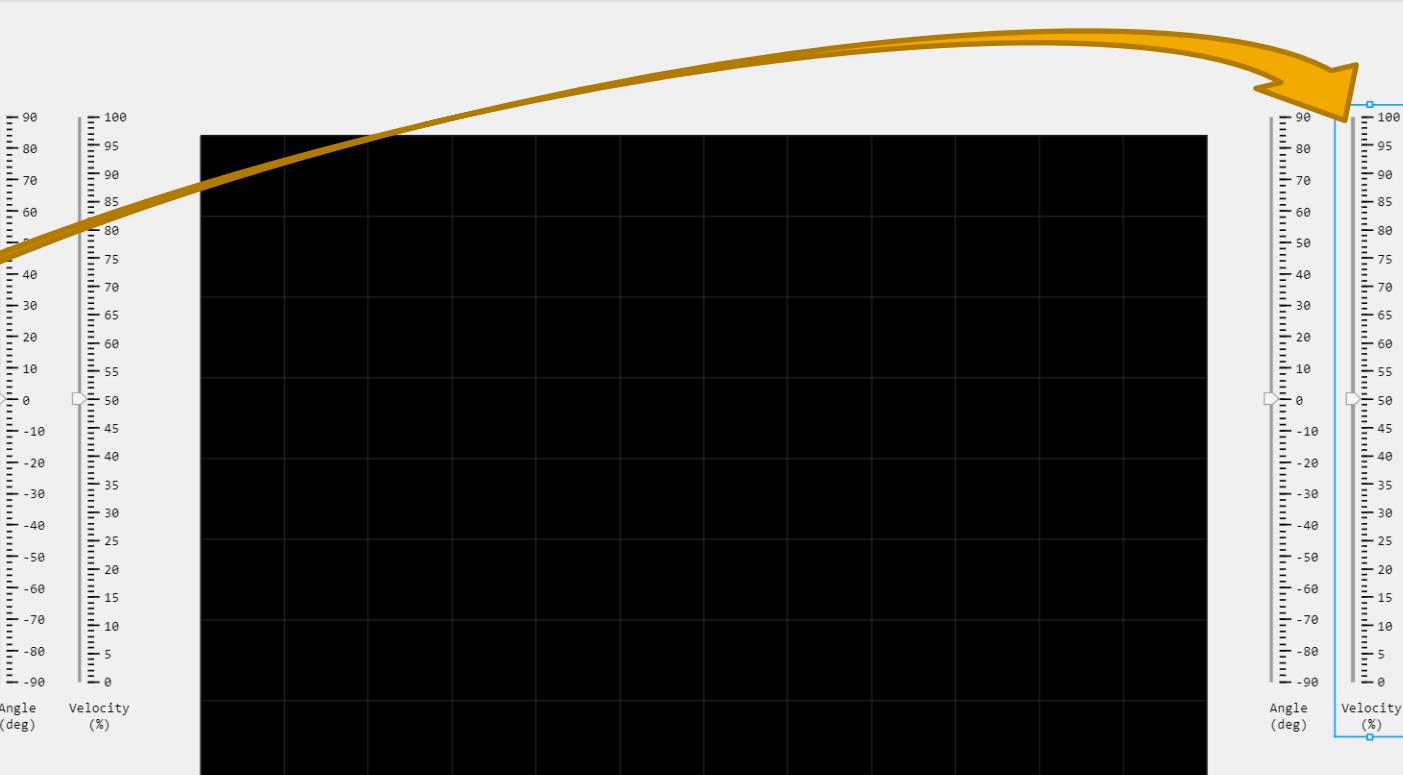
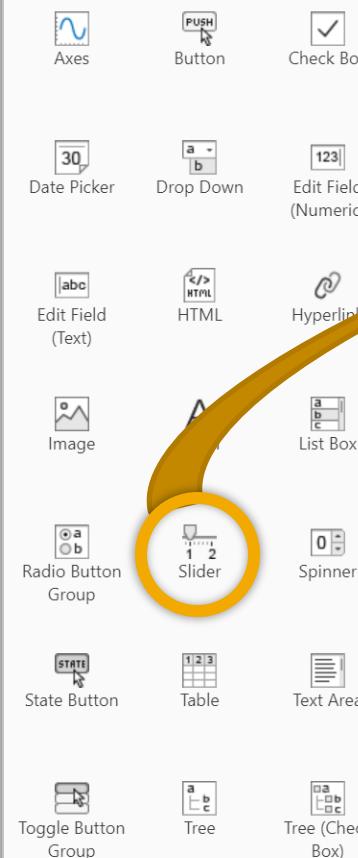
MyAstroVolley.mlapp

Component Library

Search



COMMON



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.

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

- DESIGNER TAB:** Contains buttons for Save, Convert, FILE; ALIGN; ARRANGE; and RUN.
- CANVAS:** Displays two vertical sliders. The left slider is labeled "Angle (deg)" and the right slider is labeled "Velocity (%)".
- Component Library:** A sidebar listing various UI components like Axes, Button, Check Box, etc.
- Component Browser:** Shows the component tree and properties for the sliders.
 - Slider Properties:**
 - Label: Velocity(%)
 - Value: 50
 - Limits: 0,100
 - Orientation: Vertical
 - Ticks: MajorTicks: 0,5,10,15,20,25,30,35; MinorTicks: 0,1,2,3,4,5,6,7,8,9,10
 - Font: Consolas, Size: 12, Weight: Normal, Angle: 0, Color: 0.00,0.00,1
 - Slider 1 Properties:**
 - Label: Angle(deg)
 - Value: 70
 - Limits: 0,100
 - Orientation: Vertical
 - Ticks: MajorTicks: 0,10,20,30,40,50,60,70,80,90,100; MinorTicks: 0,5,10,15,20,25,30,35
 - Font: Consolas, Size: 12, Weight: Normal, Angle: 0, Color: 0.00,0.00,1

DESIGNER

CANVAS

VIEW



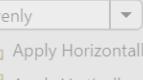
FILE



ALIGN



ARRANGE



SPACE



RUN

MyAstroVolley.mlapp

Component Library

Search



COMMON



Axes



Button



Check Box



Date Picker



Drop Down



Edit Field (Numeric)



Edit Field



HTML



Hyperlink



Image



Label



List Box



Radio Button



Slider



Spinner



State Button



Table



Text Area



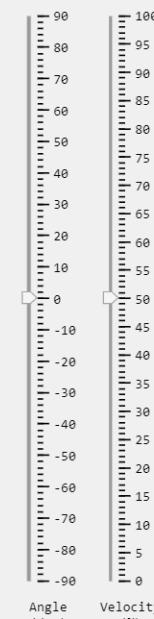
Toggle Button



Tree



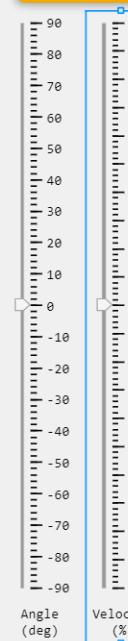
Tree (Check Box)



Angle (deg)

Velocity (%)

Right-Click



Velocity (%)

Angle (deg)

Design View Code View

Component Browser

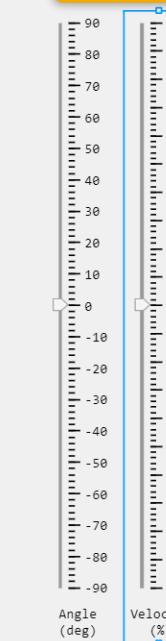
Search

MyAstroVolley

app.UIFigure

app.VelocitySlider2

app.AngleSlider2



Velocity (%)

Angle (deg)

FONT

FontName

FontSize

FontWeight

FontAngle

FontColor

INTERACTIVITY

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

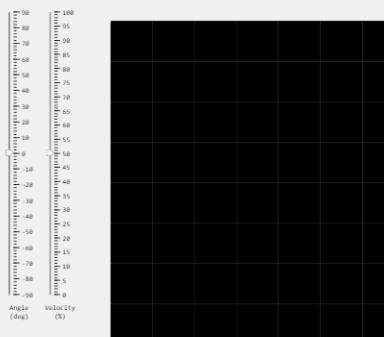
AngleSlider1ValueChanging

VelocitySlider1ValueChanging

AngleSlider2ValueChanging

VelocitySlider2ValueChanging

App Layout



```
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
```

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



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

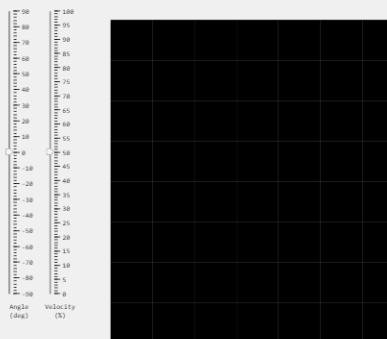
AngleSlider1ValueChanging

VelocitySlider1ValueChanging

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

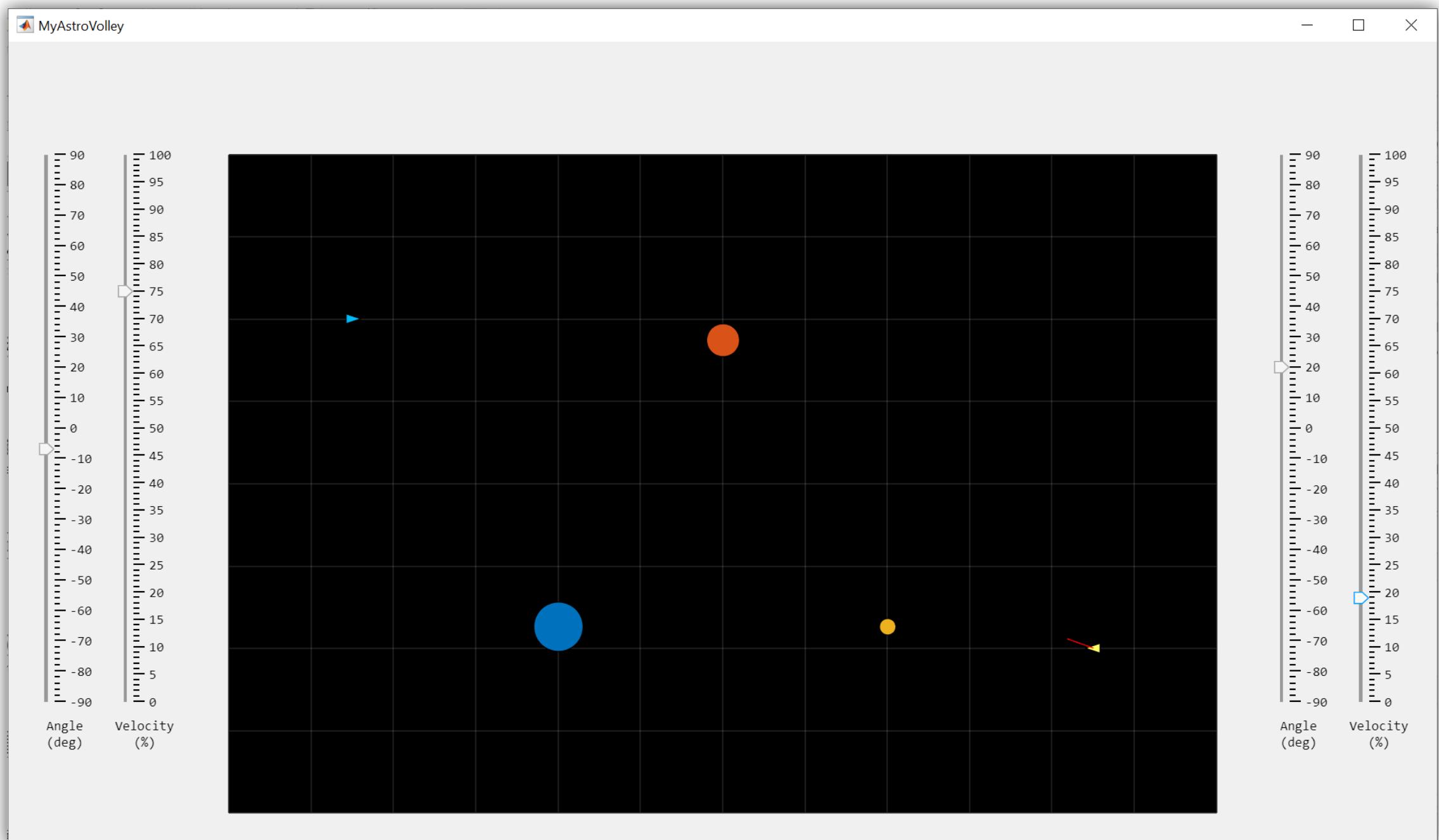
FontWeight



```

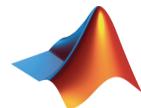
41         vx = vel*cosd(ang);
42         vy = vel*sind(ang);
43         delete(app.hq)
44         app.hq = quiver(app.UIAxes,app.xs(1),app.ys(1),vx,vy,"Color","red");
45     end
46
47     % Value changing function: VelocitySlider1
48     function VelocitySlider1ValueChanging(app, event)
49         ang = 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

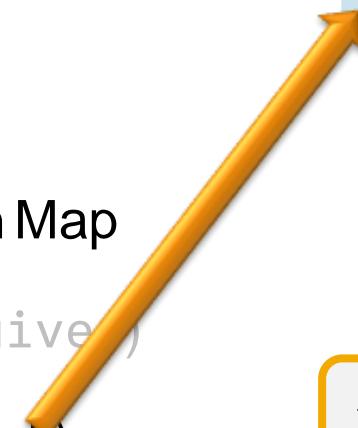
- Installing and Playing AstroVolley
- Plotting Ships (Triangles)
- Plotting Grav. Bodies (Circles)
- Gravity and Orbits
- Launching Volleys
 - Creating a Random Mission Map
 - Plotting Velocity Vectors (quiver)
 - Animating Trajectories (comet)



App Designer

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

>> open MyAstroVolley2.mlapp



DESIGNER

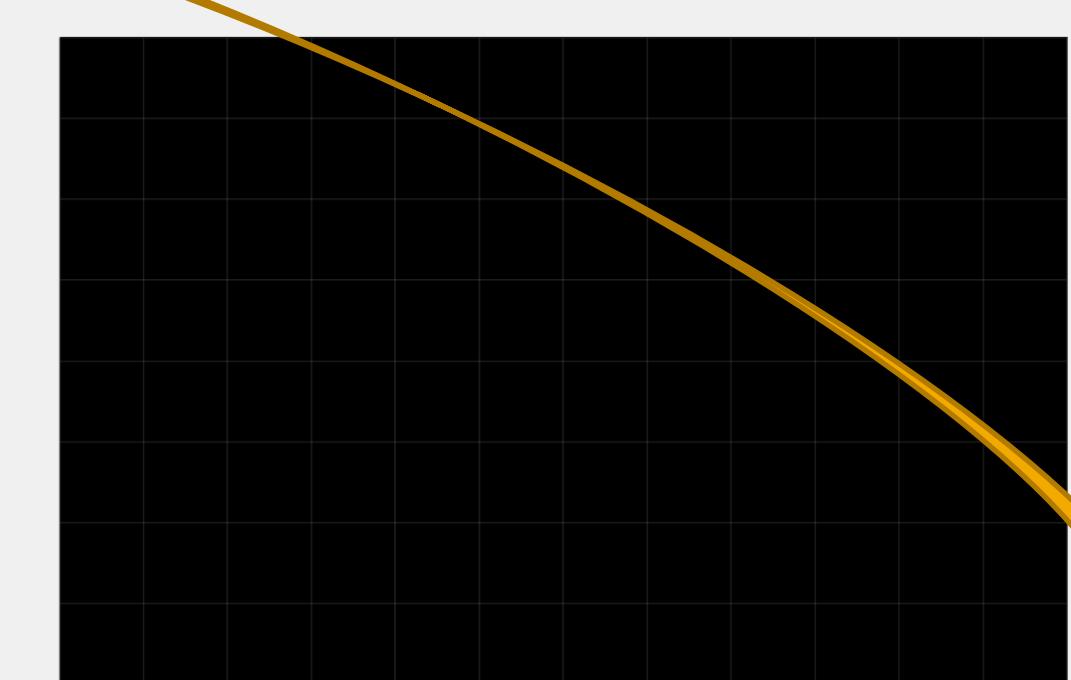
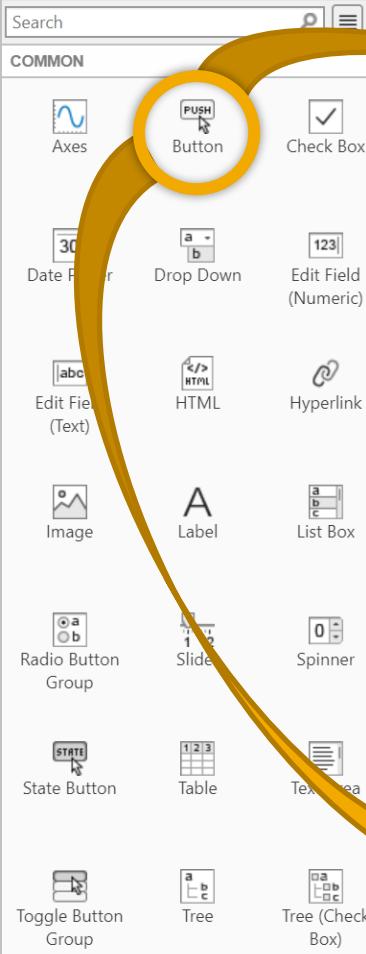
CANVAS

VIEW



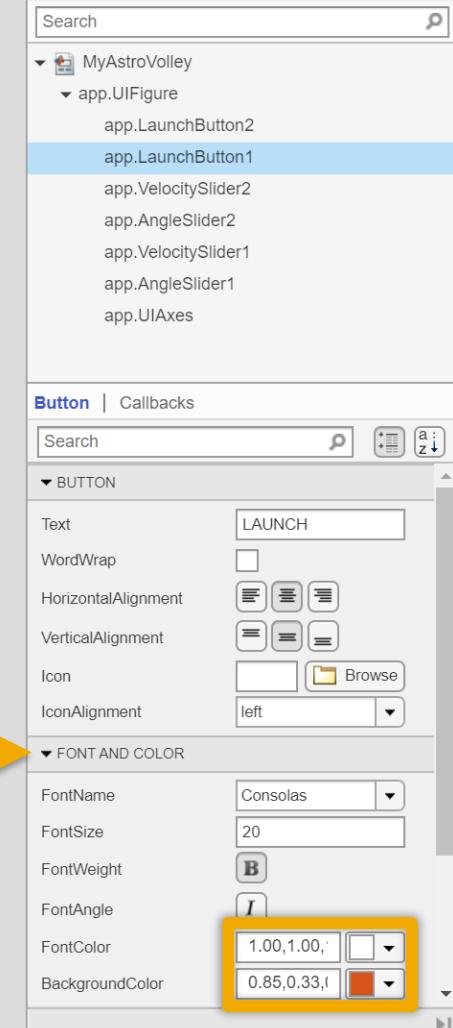
MyAstroVolley.mlapp

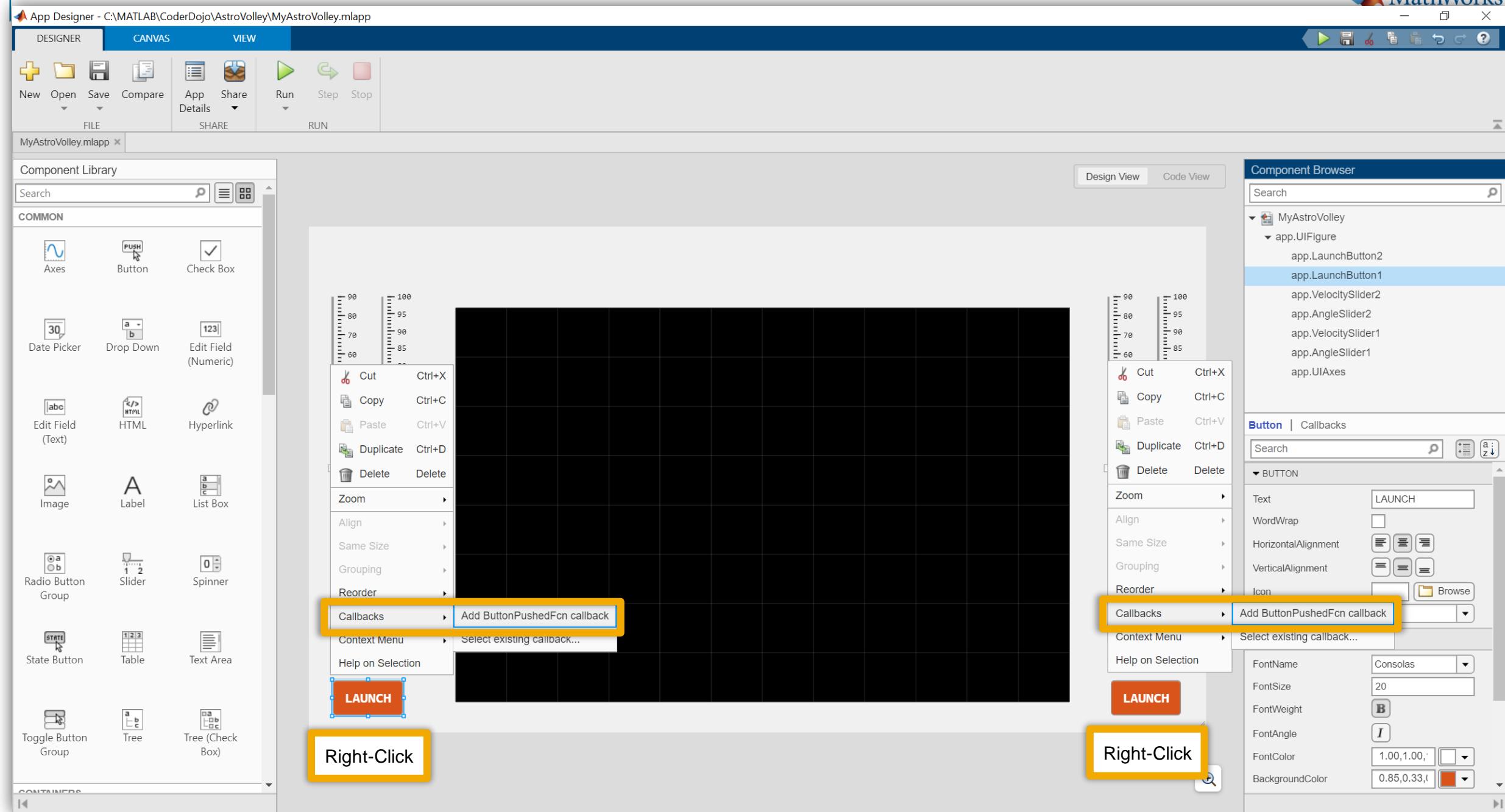
Component Library



Design View Code View

Component Browser





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

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

- DESIGNER** tab selected.
- EDITOR** tab open, displaying the code for `MyAstroVolley.mlapp`.
- VIEW** tab open.
- Toolbar** with various icons for file operations, navigation, and code editing.
- Code Browser** pane on the left showing callbacks and properties for components like `AngleSlider1`, `VelocitySlider1`, `LaunchButton1`, and `LaunchButton2`.
- App Layout** pane showing a grid-based layout with a black background and a red button labeled `LAUNCH`.
- Component Browser** pane on the right listing components under `MyAstroVolley`, including `UIAxes`, `LaunchButton1`, `LaunchButton2`, `VelocitySlider1`, `AngleSlider1`, `VelocitySlider2`, and `AngleSlider2`. The `VelocitySlider2` and `AngleSlider2` entries are highlighted with a yellow box.
- Editor Content** (Code View):

```

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
91 % Component initialization
92 % methods (Access = private)
93

```

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**

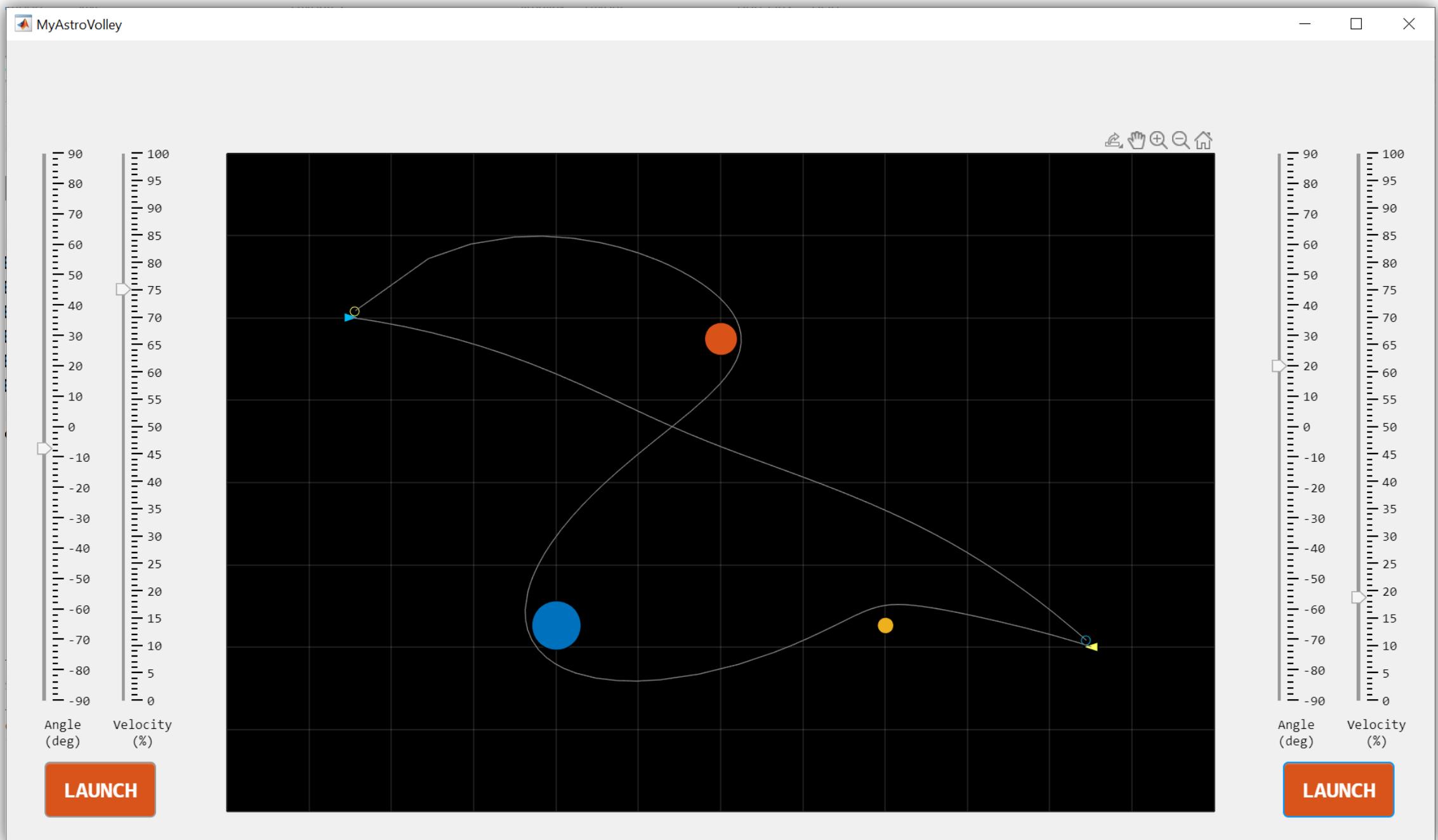
FontAngle: *I*

FontColor: 1.00,1.00,1.00

```

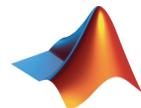
59 % Value changing function: AngleSlider2
60 function AngleSlider2ValueChanging(app, event)
61     ang = 180 - event.Value;
62     vel = app.VelocitySlider2.Value;
63     vx = vel*cosd(ang);
64     vy = vel*sind(ang);
65     delete(app.hq)
66     app.hq = quiver(app.UIAxes,app.xs(2),app.ys(2),vx,vy,"color","red");
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 Random Mission Map
 - 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 Map
- 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

```

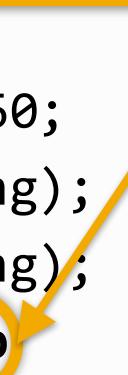
function app = mapMission
ax = gca; % get axes
cla(ax) % clear axes
app.xs = [-225 225]; % ship x-pos
app.ys = 300*rand(1,2) - 150; % y-pos
app.Rs = [ 5 5]; % ship sizes
cs = [0 .73 .96; 1 1 .4]; % ship colors
xt = [-5 4 4 -5]/5; % triangle x-points
yt = [ 0 3 -3 0]/5; % triangle y-points
fill(ax,app.xs(1)-app.Rs(1)*xt,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

MyAstroVolley.mlapp x

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



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



Copy



Paste



Duplicate



Delete



Zoom



Align



Same Size



Grouping



Reorder



Callbacks



Add ButtonPushedFcn callback



Context Menu



Select existing callback...



Help on Selection



Angle (deg)



Velocity (%)

LAUNCH

Design View Code View

Component Browser

Search

MyAstroVolley

app.UIFigure

app.NewMissionButton

app.LaunchButton2

app.LaunchButton1

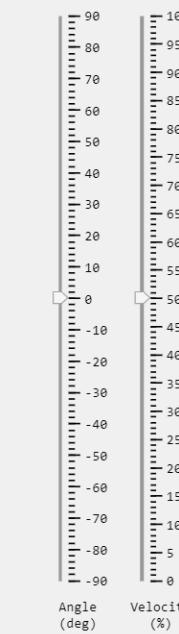
app.VelocitySlider2

app.AngleSlider2

app.VelocitySlider1

app.AngleSlider1

app.UIAxes



LAUNCH

Button | Callbacks

Search

▼ BUTTON

Text



New.Mission

WordWrap

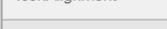


HorizontalAlignment



VerticalAlignment

Icon



IconAlignment



left



▼ FONT AND COLOR

FontName



Consolas

FontSize



18

FontWeight



B

FontAngle



I

FontColor

0.00,0.00,1

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

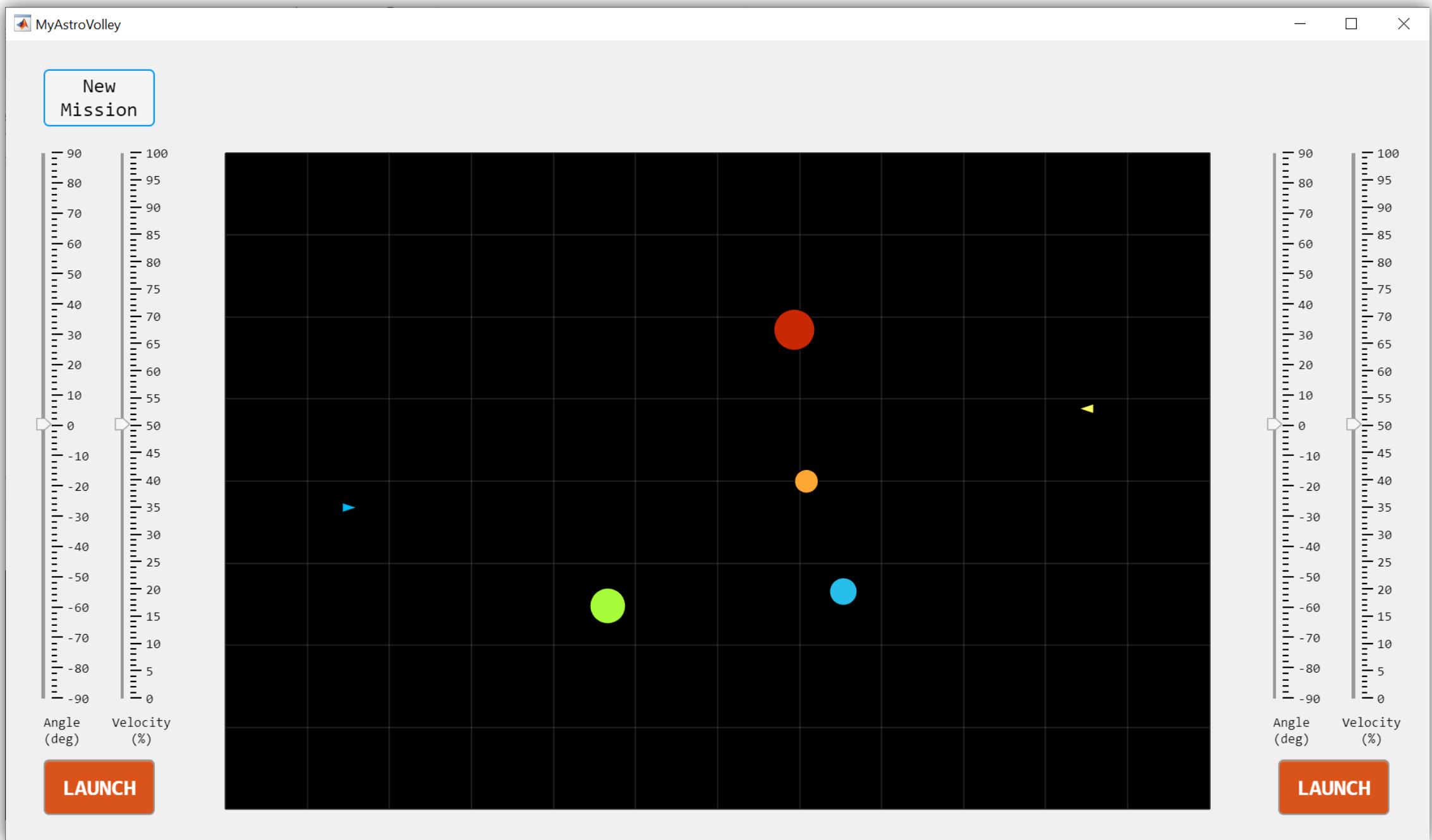
```

The left sidebar shows the 'Code Browser' with 'Callbacks', 'Functions', and 'Properties' sections, and the 'App Layout' section which includes a preview of the application's user interface.

The right sidebar shows the 'Component Browser' with a tree view of components and the 'Button' section with properties for the 'NewMission' button.

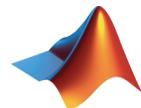
A callout bubble points to the `NewMissionButtonPushed` function with the following text:

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 Random Mission Map
 - 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 Map
- 5) Keeping Score

>> open MyAstroVolley4.mlapp

DESIGNER

CANVAS

VIEW



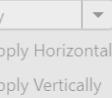
FILE



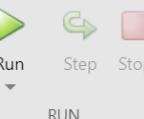
ALIGN



ARRANGE



SPACE



RUN

MyAstroVolley.mlapp

Component Library

Search

COMMON



Axes



Push Button



Check Box



Drop Down



Edit Field (Numeric)



HTML



Hyperlink



Image



Label



List Box



Radio Button Group



Slider



Spinner



State Button



Table



Text Area



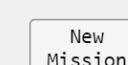
Toggle Button Group



Tree



Tree (Check Box)



New Mission



Design View | Code View

Component Browser

Search

MyAstroVolley

app.UIFigure

app.CallSignField2

app.CallSignField1

app.NewMissionButton

app.LaunchButton2

app.LaunchButton1

app.VelocitySlider2

app.AngleSlider2

app.VelocitySlider1

app.AngleSlider1

app.UIAxes

Edit Field (Text) | Callbacks

Search

TEXT

Value

Placeholder

HorizontalAlignment

FONT AND COLOR

FontName

FontSize

FontWeight

FontAngle

FontColor

BackgroundColor

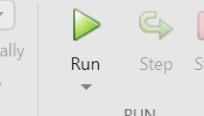
INTERACTIVITY

POSITION

DESIGNER

CANVAS

VIEW



MyAstroVolley.mlapp

Component Library

Search



COMMON



Axes



Button



Check Box



30



Drop Down



Edit Field (Numeric)



abc



Image



Hyperlink



Label



List Box



a b



1 2

Slider



0

Spinner



STATE



1 2 3

Table



Text Area



a b

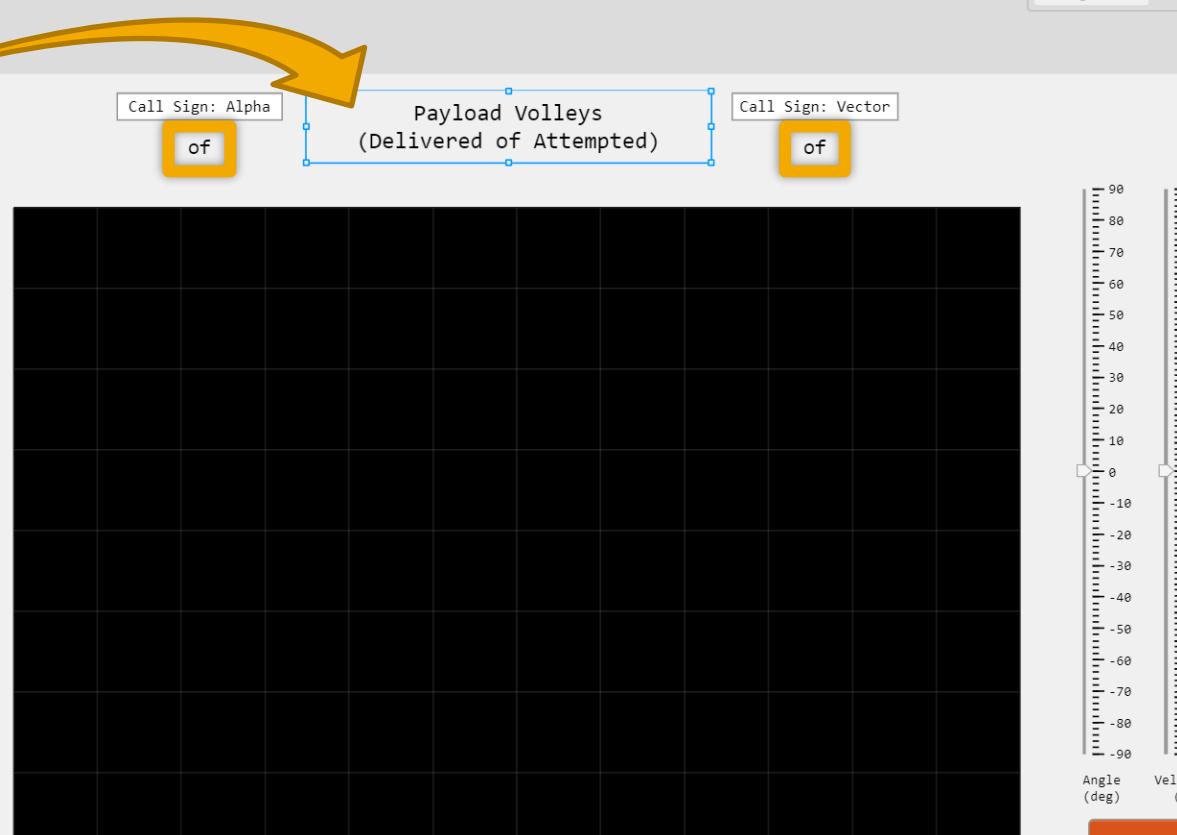
Toggle Button Group



Tree

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

FONT AND COLOR

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

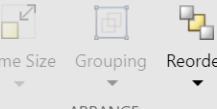


Save

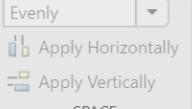
Convert



ALIGN



ARRANGE



SPACE



RUN

MyAstroVolley.mlapp x

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



Design View | Code View

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

ClearScores

WordWrap

HorizontalAlignment

VerticalAlignment

Icon

Browse

IconAlignment

left

▼ FONT AND COLOR

FontName

Consolas

FontSize

18

FontWeight

B

FontAngle

I

FontColor

0.00,0.00,1

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.
- FILE** menu open, showing options like Save, Print, Go To, and NAVIGATE.
- EDITOR** tab.
- VIEW** tab.
- INSERT** toolbar with buttons for Callback, Function, Property, App Input Arguments, App Help Text, Comment, Indent, Run, Step, and Stop.
- CODE** toolbar with buttons for RUN and RUN.
- Code Browser** pane on the left showing callbacks, functions, and properties. It lists methods like startupFcn, NewMissionButtonPushed, AngleSlider1ValueChanging, VelocitySlider1ValueChanging, AngleSlider2ValueChanging, VelocitySlider2ValueChanging, LaunchButton1Pushed, LaunchButton2Pushed, ClearScoresButtonPushed, and ClearScoresButtonPushed.
- Component Browser** pane on the right showing the component tree under MyAstroVolley. It includes app.UIFigure, app.VolleyesLabel, app.ofLabel2, app.ofLabel1, app.CallSignField2, app.CallSignField1, app.ClearScoresButton, app.DeliveredField2, app.AttemptedField2, app.DeliveredField1, and app.AttemptedField1.
- App Layout** pane at the bottom showing the UI design with a grid, call sign input fields, and launch buttons.
- Code Editor** pane showing the MATLAB code for the application. The ClearScoresButtonPushed function is highlighted with an orange box.

```

% 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 Component Browser panel on the right lists the application's components, including UIAxes, labels, sliders, and buttons. The DeliveredField2 button is selected and highlighted with an orange box. The code browser on the left shows the callbacks and properties for each component.

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: 0.00,0.00,0.00

Code Browser

Callbacks | Functions | Properties

Search

```

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

```

App Layout

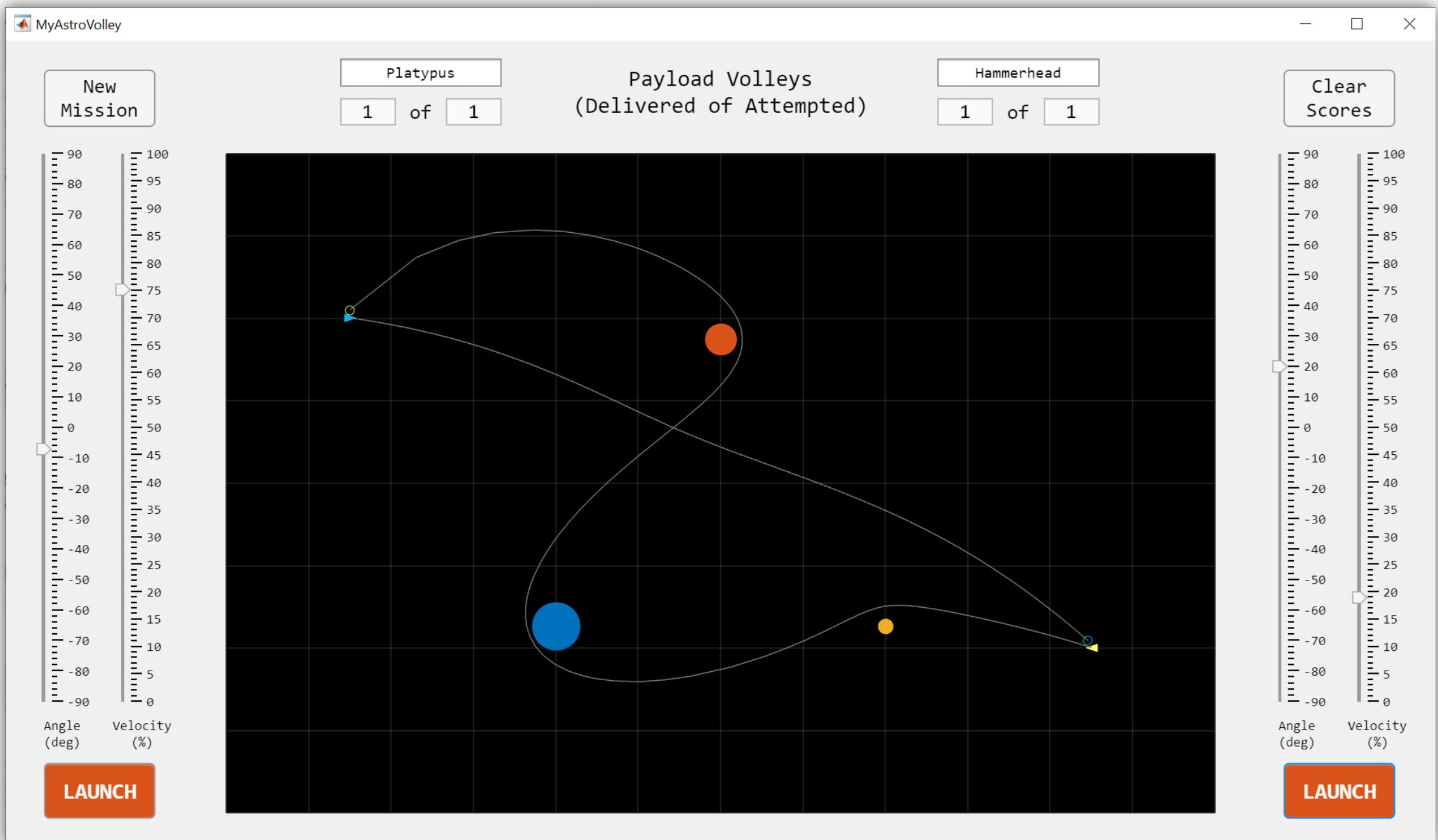
New Mission

Call Sign: Alpha

Payload Volleys (Delivered of Attempted)

Angle (deg) Velocity (m/s)

LAUNCH



Excellent work everyone!

Thank you for being a part of
MATLAB AstroVolley!