App that Simulates Gravity Between Objects in Space (ASGBOS)

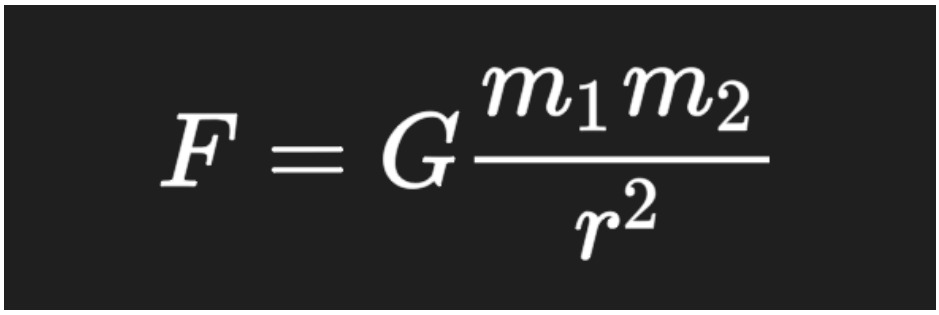
Seth Iris Canonigo, Nathan Huey, Lothar Escobar

Introduction

The purpose of the program is to help expose individuals with the force calculations when first trying to determine the gravitational force between two large bodies in space. The program allows the user to choose two mass values of their choice along with the distance so all can be used within the Gravitational Force Equation. We figured this may be a good topic to review and to make an application for since it's one of the topics that is first introduced within general physics when a student begins their journey in school and chooses to follow through with a stem related topic. We included a help interaction button that demonstrates the steps needed to take for the user to work it properly, making it easy for anyone to maneuver through the application and be able to work on calculations of their choice.

Background

The concept that we based our project on is the Law of Gravitation, which states that the force exerted between two large masses in space which can be found using the equation:



The variables are:

G = 6.67 * 10^(-11) N*m^2 / kg^2 (gravity Constant)

M1 = Mass 1 (In kg preferably)

M2 = Mass 2 (in kg preferably)

R = Distance between the two masses (in meters preferably)

This is a topic that is covered within the first course of physics in college and we also wanted to create an app for anyone to use. We implemented a help interaction button to help those who do not understand the concept but would still like to participate and explore the application so that they can see the outcome of whatever they choose as their inputs.

Flowchart

1) **Input 1: Dropdown**

   Will be the user's choice of an astronomical body in the dropdown with a mass provided to be used as M1 within the equation.

2) **Input 2: Mass 2**

   Will be the user's choice of mass they want to use and it will be saved as M2 for the app to run (app will provide an error if the mass is 0 or below).

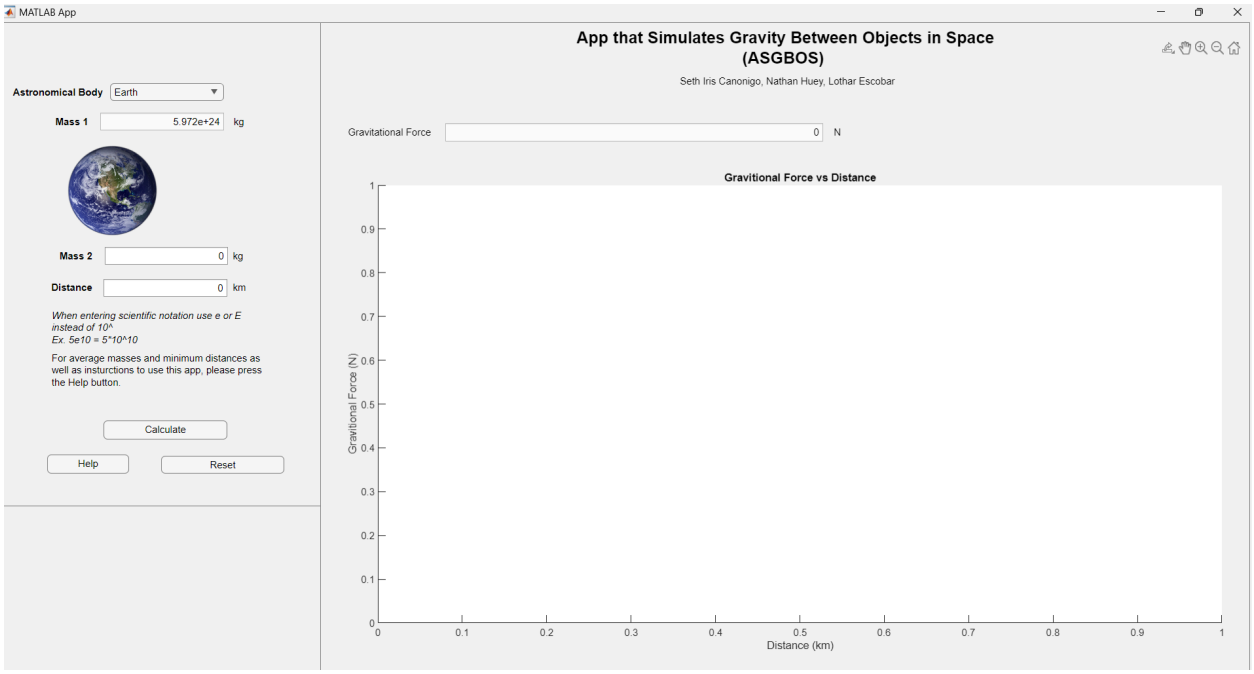
3) **Input 3: Distance**

   The user can choose the distance to be used in the equation as R for the app's calculations(app will provide an error if the distance is smaller than the given planet's minimum distance).

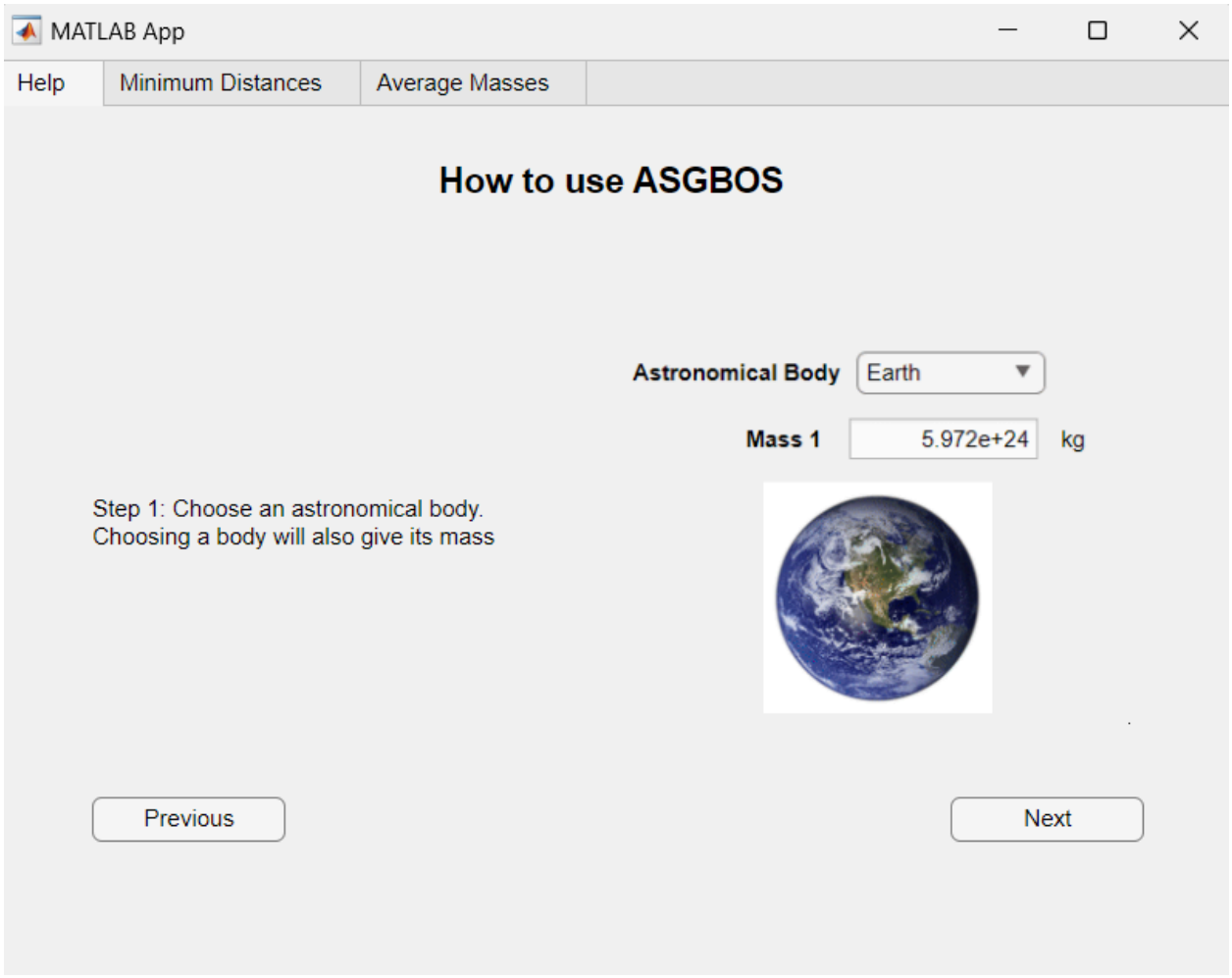
4) **Outputs**

   Press the 'Calculate' button and the result of the force will be provided above and below the calculated force a graph will display the Force vs. distance relationship.

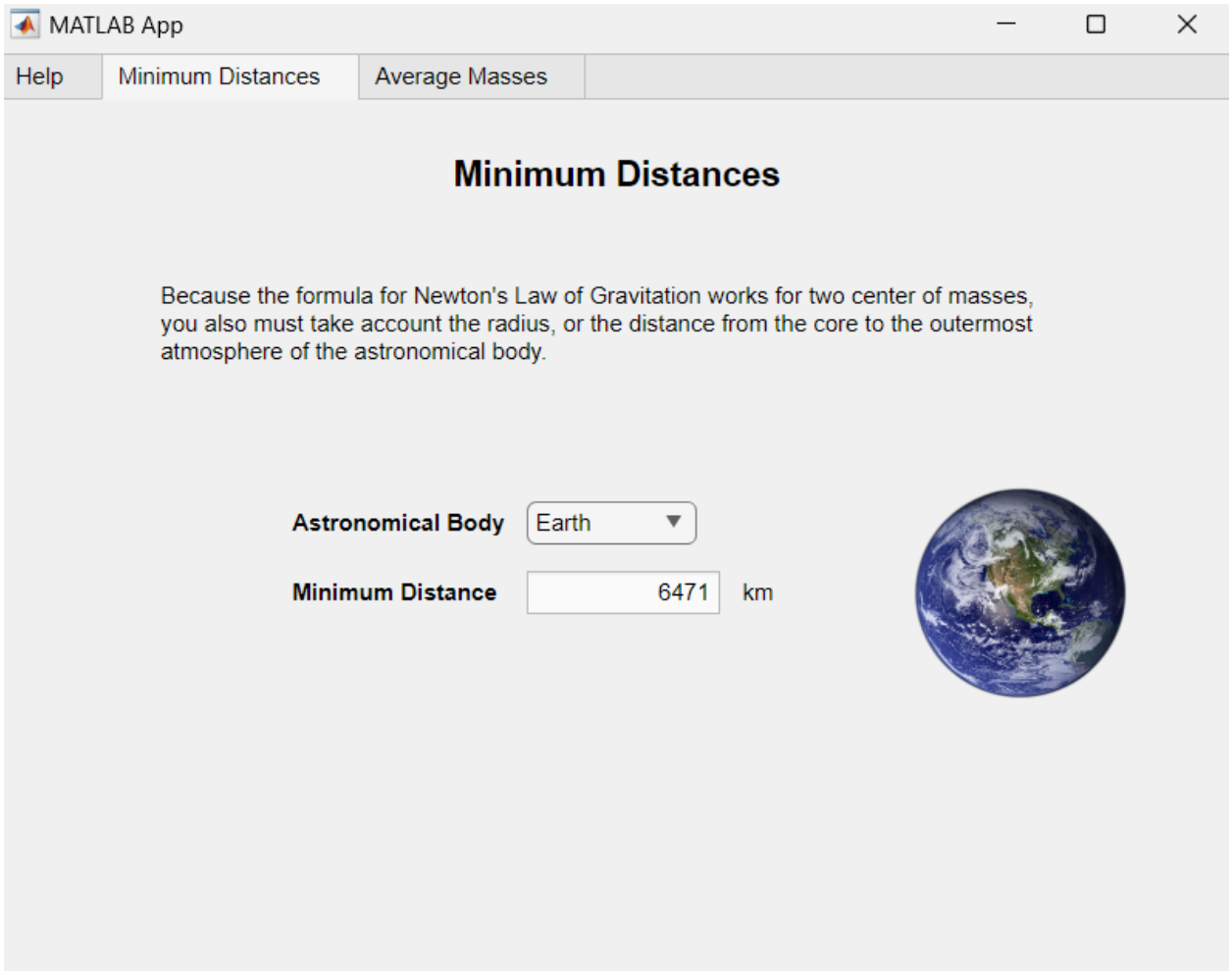
Design and layouts


MATLAB App
App that Simulates Gravity Between Objects in Space
(ASGBOS)
Seth Iris Canonigo, Nathan Huey, Lothar Escobar
Astronomical Body
Earth
Mass 1
5.972e+24
kg
Mass 2
0
kg
Distance
0
km
When entering scientific notation use e or E instead of 10^
Ex. 5e10 = 5*10^10
For average masses and minimum distances as well as insturctions to use this app, please press the Help button.
Calculate
Help
Reset
Gravitational Force
0
N
Gravitional Force vs Distance
Gravitional Force (N)
Distance (km)

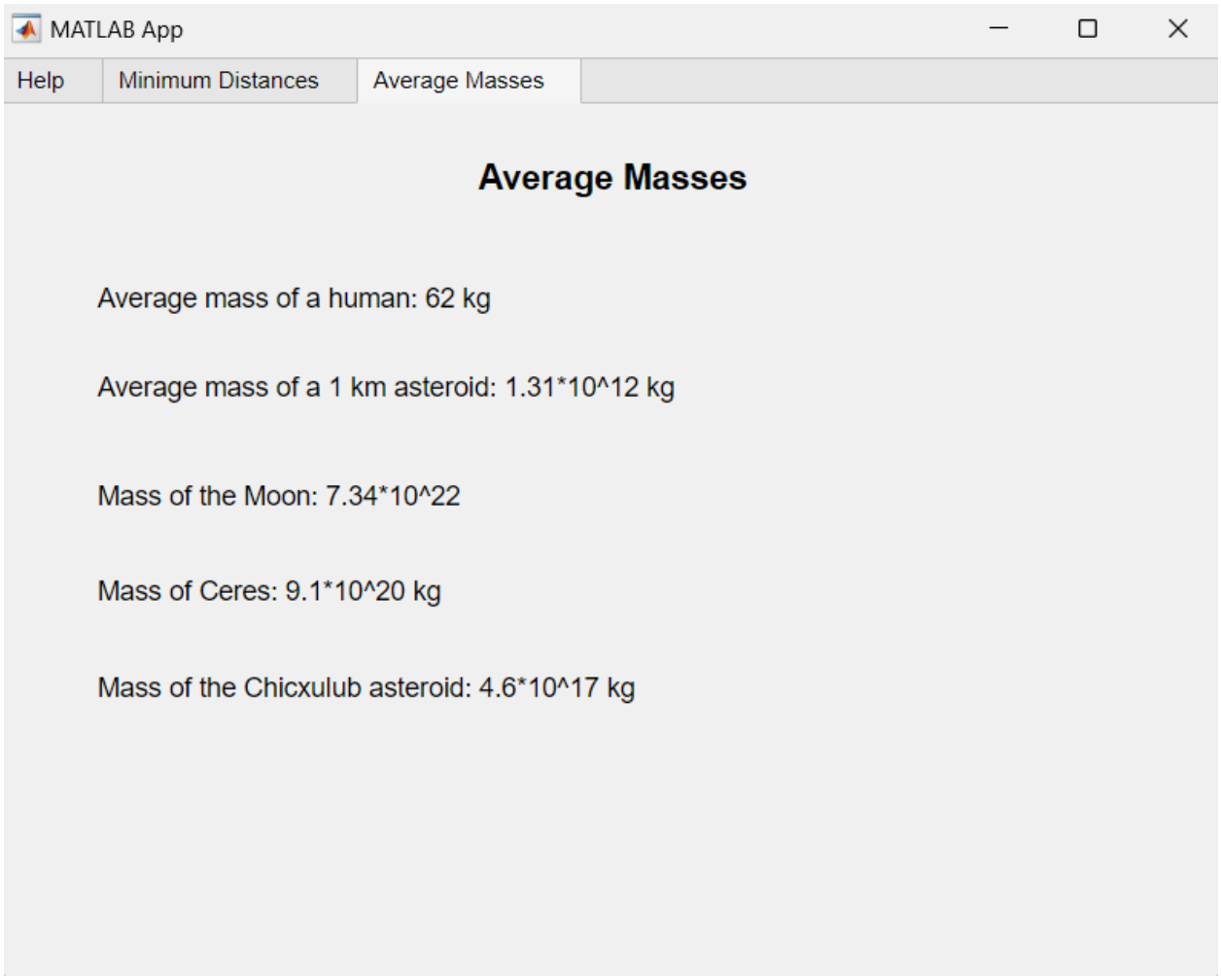
The main app



The help tab in the help menu

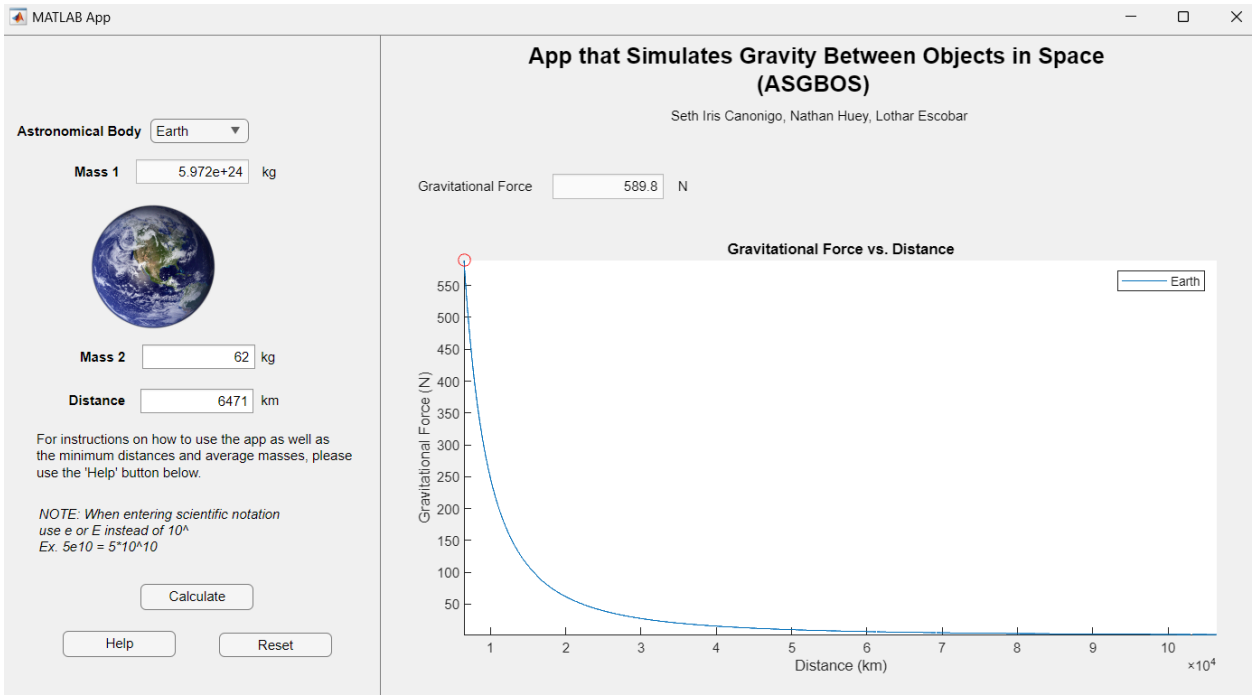


The minimum distance tab



The average example masses tab

Results



Now below are two different calculations to prove the accuracy of the app. The first calculation uses the formula for Newton's Law of Gravitation.

$$F = 6.67430 \times 10^{-11} * \frac{62 * 5.972\times10^{24}}{(6471*1000)^2}$$ (6471 is multiplied by 1000 to convert from km to m)

$$F = 6.67430 \times 10^{-11} * \frac{3.70264\times10^{26}}{4.1873841\times10^{13}}$$

$$F = 6.67430 \times 10^{-11} * 8.84237011 \times 10^{12}$$

$$F = 589.786\ N$$

As you can see the results are the same, the app just rounds to the 1st decimal place.

Since the astronomical body is earth, the second calculation uses the formula for the force Weight, derived from the second law of motion.

$$F = m * g$$

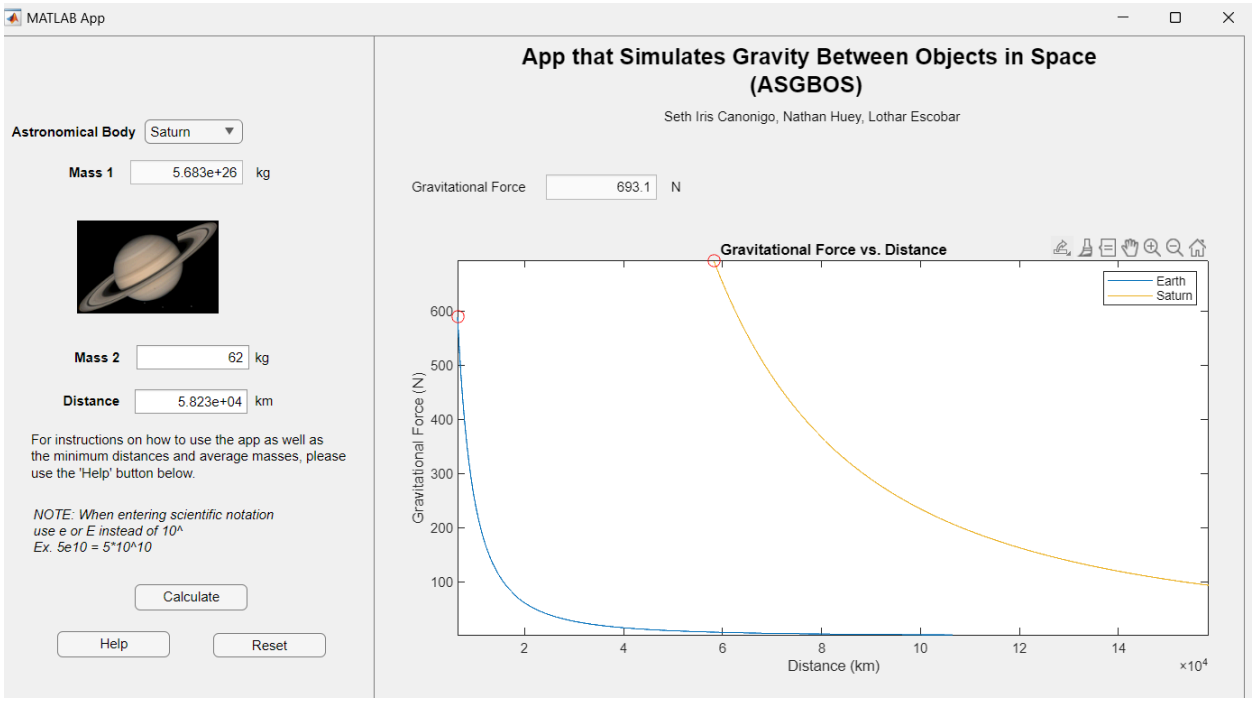
Where m is the mass of the second object and g is the acceleration due to Earth's gravity, which is

$$g = 9.81 \frac{m}{s^2}$$

$$F = 62 * 9.81$$

$$F = 608.22\ N$$

It is within the margin of error.



This is another result with the same second mass but using Saturn for the first mass and the radius of the planet for the distance.

Conclusion

Our group learned a lot during this project. Most of all we learned that you can have a button that opens a different app, which is what we did for the 'Help' button. Another thing we

learned was that you can never technically escape anything's gravitational pull. Some things we could improve in the future could be adding a second graph for more massive objects so it doesn't skew the graph or make it hard to read the results for the smaller planets. This would also allow us to add things like black holes. Additionally, adding sounds for each astronomical body would also give a more immersive feeling to the app. Overall, this project has broadened our technical skills and given us new ideas for future improvements.

Appendix

Main app file

```
classdef ASGBOS < matlab.apps.AppBase
   % Properties that correspond to app components
   properties (Access = public)
       UIFigure                        matlab.ui.Figure
       RightPanel                      matlab.ui.container.Panel
       Names                           matlab.ui.control.Label
       Newtonlabel                     matlab.ui.control.Label
       GravitationalForceEditField     matlab.ui.control.NumericEditField
       GravitationalForceLabel         matlab.ui.control.Label
       AppthatSimulatesGravityBetweenObjectsinSpaceASGBOSLabel
matlab.ui.control.Label
       UIAxes                          matlab.ui.control.UIAxes
       LeftPanel                       matlab.ui.container.Panel
       WhenenteringscientificnotationuselowercaseeEx5e1051010Label
matlab.ui.control.Label
       Mass1EditField                  matlab.ui.control.NumericEditField
       Mass1EditFieldLabel             matlab.ui.control.Label
       CalculateButton                 matlab.ui.control.Button
       kgLabel_2                       matlab.ui.control.Label
       kmLabel                         matlab.ui.control.Label
       kgLabel                         matlab.ui.control.Label
       Label                           matlab.ui.control.Label
       DistanceEditField               matlab.ui.control.NumericEditField
       DistanceEditFieldLabel          matlab.ui.control.Label
       Mass2EditField                  matlab.ui.control.NumericEditField
       Mass2EditFieldLabel             matlab.ui.control.Label
       ResetButton                     matlab.ui.control.Button
       Background                      matlab.ui.control.Image
       HelpButton                      matlab.ui.control.Button
       AstronomicalBodyDropDown        matlab.ui.control.DropDown
       AstronomicalBodyDropDownLabel   matlab.ui.control.Label
```

```
    end

    properties (Access = private)
        % variable to store mass of seleceted dropdown
        Mass1value = 5.972e24; %stores value for minimum distance of each
astronimical body (Default: Earth)
        Mindistance = 6471;%stores value for minimum distance of each
astronimical body (Default: Earth)
        BodyName = 'Earth';% Stores the body name for the graph (Default: Earth)
    end

    methods (Access = private)

    end

    % Callbacks that handle component events
    methods (Access = private)
        % Code that executes after component creation
        function startupFcn(app)
            int32 mass1val;
            int32 mindist;
            char bodyname;

        end
        % Value changed function: AstronomicalBodyDropDown, Mass1EditField
        function AstronomicalDropDownValueChanged(app, event)
            object = app.AstronomicalBodyDropDown.Value;

            % This sets the mass (kg) and minimum distance for each
            % selected planet in the dropdown menu
            switch object
                case 'Earth'
                 app.Background.ImageSource = imread('Earth.png');
                 mass1val= 5.972e24;
                 mindist= 6471;
                 bodyname= 'Earth';
                case 'Mercury'
                 app.Background.ImageSource = imread('Mercury.png');
                 mass1val= 3.285e23;
                 mindist= 2440;
                 bodyname= 'Mercury';
                case 'Venus'
                 app.Background.ImageSource = imread('Venus.png');
                 mass1val= 4.867e24;
                 mindist= 6050;
                 bodyname= 'Venus';
                case 'Mars'
                 app.Background.ImageSource = imread('Mars.png');
                 mass1val= 6.390e23;
```

```
            mindist= 3400;
            bodyname= 'Mars';
        case 'Jupiter'
            app.Background.ImageSource = imread('Jupiter.png');
            mass1val= 1.898e27;
            mindist= 69911;
            bodyname= 'Jupiter';
        case 'Saturn'
            app.Background.ImageSource = imread('Saturn.png');
            mass1val= 5.683e26;
            mindist= 58232;
            bodyname= 'Saturn';
        case 'Uranus'
            app.Background.ImageSource = imread('Uranus.png');
            mass1val= 8.681e25;
            mindist= 25559;
            bodyname= 'Uranus';
        case 'Neptune'
            app.Background.ImageSource = imread('Neptune.png');
            mass1val= 1.024e26;
            mindist= 24622;
            bodyname= 'Neptune';
        case 'Pluto'
            app.Background.ImageSource = imread('Pluto.png');
            mass1val= 1.309e22;
            mindist= 2188;
            bodyname= 'Pluto';
        case 'Sun'
            app.Background.ImageSource = imread('Sun.png');
            mass1val= 1.989e30;
            mindist= 700000;
            bodyname= 'Sun';
            %Sagittarius is too big so it is removed
        %case 'Sagittarius A*'
            %app.Background.ImageSource = imread('Sagittarius A.jpg');
            %mass1val= 8.553e36;
    end

    %stores mass value in property so can use in other callbacks
    app.Mass1value = mass1val;
    %stores min distance values to use in other callback
    app.Mindistance= mindist;
    %displays value in Editfield for Mass 1
    app.Mass1EditField.Value = mass1val;
    %stores name of body picked
    app.BodyName= bodyname;

end
% Button pushed function: CalculateButton
```

```
        function CalculateButtonPushed(app, event)
            M1= app.Mass1value;
            M2= app.Mass2EditField.Value;
            d= app.DistanceEditField.Value * 1000;
            G=6.67e-11;
            % Checking if inputs are legal
            if d < app.Mindistance * 1000
                uialert(app.UIFigure, ['The distance value is below the minimum
allowed distance!' ...
                    ' Please check the table for the minimum distance allowed
for each astronomical body.'], ...
                    'Input Error', 'Icon', 'error');
                return;
            end
            % Checks maximum distance
            if d> (app.Mindistance + 100000)*1000
                uialert(app.UIFigure, ['The distance is too large and will not
fit on the graph. Please only do the minimum' ...
                    ' distance plus 100000 at most. Thank you.'],'Input Error',
'Icon', 'error');
                return;
            end

            % Gives error if User-given mass is or less than 0
            if M2 == 0 || M2 < 0
                uialert(app.UIFigure, ['Mass 2 has been left as 0 or input as an
negative number. Please enter a value above 0' ...
                    ' to get accurate values.'],'Input Error', 'Icon', 'error');
                return;
            end
            %calculates specific distance user wanted
            Fgravspecific=(G*M1*M2)/d^2;
            app.GravitationalForceEditField.Value = Fgravspecific;
            %setting min and max distances for graph
            mindistance= app.Mindistance;
            maxdistance= mindistance + 100000;
            %making array distances
            distances= linspace(mindistance,maxdistance,100000) *1000;%*1000
converts km to m for calculation
            %calculates gravitational force for each distance
            Fgrav= (G*M1*M2)./distances.^2;
            %plots range of distances
            plot(app.UIAxes,distances/1000, Fgrav, 'DisplayName', app.BodyName);
            xlabel(app.UIAxes, 'Distance (km)');
            ylabel(app.UIAxes, 'Gravitational Force (N)');
            title(app.UIAxes, 'Gravitational Force vs. Distance');
            legend(app.UIAxes, 'show');

            %plot specific point asked for
```

```
            hold(app.UIAxes, 'on');
            h= plot(app.UIAxes, d/1000, Fgravspecific, 'ro', 'Markersize', 8);
            h.HandleVisibility= 'off';
            axis(app.UIAxes, 'tight');
        end
        % Button pushed function: HelpButton
        function HelpButtonPushed(app, event)
            HowToUse
        end
        % Button pushed function: ResetButton
        function ResetButtonPushed(app, event)

            % Resets the outputs to 0 and inputs to the default values (Earth)
            app.AstronomicalBodyDropDown.Value = 'Earth';
            app.Background.ImageSource = imread('Earth.png');
            app.Mass1EditField.Value= 5.972e24;
            app.Mass2EditField.Value= 0;
            app.DistanceEditField.Value=0;
            app.GravitationalForceEditField.Value= 0;
            cla(app.UIAxes, 'reset');
            xlabel(app.UIAxes, 'Distance (km)');
            ylabel(app.UIAxes, 'Gravitational Force (N)');
            title(app.UIAxes, 'Gravitational Force vs. Distance');
            legend(app.UIAxes, 'off');
        end
        % Image clicked function: Background
        function BackgroundImageClicked(app, event)
            object = app.AstronomicalBodyDropDown.Value;
            switch object
                case 'Earth'
                case 'Mercury'
                case 'Venus'
                case 'Mars'
                case 'Jupiter'
                case 'Saturn'
                case 'Uranus'
                 rick
                case 'Neptune'

                case 'Pluto'

                case 'Sun'

            end
        end
    end
    % Component initialization
    methods (Access = private)
        % Create UIFigure and components
```

```
        function createComponents(app)
            % Get the file path for locating images
            pathToMLAPP = fileparts(mfilename('fullpath'));
            % Create UIFigure and hide until all components are created
            app.UIFigure = uifigure('Visible', 'off');
            app.UIFigure.Position = [93 93 1051 591];
            app.UIFigure.Name = 'MATLAB App';
            % Create LeftPanel
            app.LeftPanel = uipanel(app.UIFigure);
            app.LeftPanel.Position = [1 1 340 591];
            % Create AstronomicalBodyDropDownLabel
            app.AstronomicalBodyDropDownLabel = uilabel(app.LeftPanel);
            app.AstronomicalBodyDropDownLabel.HorizontalAlignment = 'right';
            app.AstronomicalBodyDropDownLabel.FontWeight = 'bold';
            app.AstronomicalBodyDropDownLabel.Position = [9 495 115 22];
            app.AstronomicalBodyDropDownLabel.Text = 'Astronomical Body';
            % Create AstronomicalBodyDropDown
            app.AstronomicalBodyDropDown = uidropdown(app.LeftPanel);
            app.AstronomicalBodyDropDown.Items = {'Earth', 'Mercury', 'Venus',
'Mars', 'Jupiter', 'Saturn', 'Uranus', 'Neptune', 'Pluto', 'Sun'};
            app.AstronomicalBodyDropDown.ValueChangedFcn =
createCallbackFcn(app, @AstronomicalDropDownValueChanged, true);
            app.AstronomicalBodyDropDown.Position = [133 495 88 22];
            app.AstronomicalBodyDropDown.Value = 'Earth';
            % Create HelpButton
            app.HelpButton = uibutton(app.LeftPanel, 'push');
            app.HelpButton.ButtonPushedFcn = createCallbackFcn(app,
@HelpButtonPushed, true);
            app.HelpButton.Position = [56 40 100 23];
            app.HelpButton.Text = 'Help';
            % Create Background
            app.Background = uiimage(app.LeftPanel);
            app.Background.ImageClickedFcn = createCallbackFcn(app,
@BackgroundImageClicked, true);
            app.Background.Position = [73 325 126 122];
            app.Background.ImageSource = fullfile(pathToMLAPP, 'Earth.png');
            % Create ResetButton
            app.ResetButton = uibutton(app.LeftPanel, 'push');
            app.ResetButton.ButtonPushedFcn = createCallbackFcn(app,
@ResetButtonPushed, true);
            app.ResetButton.Position = [195 40 100 22];
            app.ResetButton.Text = 'Reset';
            % Create Mass2EditFieldLabel
            app.Mass2EditFieldLabel = uilabel(app.LeftPanel);
            app.Mass2EditFieldLabel.HorizontalAlignment = 'right';
            app.Mass2EditFieldLabel.FontWeight = 'bold';
            app.Mass2EditFieldLabel.Position = [66 295 45 22];
            app.Mass2EditFieldLabel.Text = 'Mass 2';
            % Create Mass2EditField
```

```
            app.Mass2EditField = uieditfield(app.LeftPanel, 'numeric');
            app.Mass2EditField.Position = [126 295 100 22];
            % Create DistanceEditFieldLabel
            app.DistanceEditFieldLabel = uilabel(app.LeftPanel);
            app.DistanceEditFieldLabel.HorizontalAlignment = 'right';
            app.DistanceEditFieldLabel.FontWeight = 'bold';
            app.DistanceEditFieldLabel.Position = [56 256 55 22];
            app.DistanceEditFieldLabel.Text = 'Distance';
            % Create DistanceEditField
            app.DistanceEditField = uieditfield(app.LeftPanel, 'numeric');
            app.DistanceEditField.Position = [125 256 100 22];
            % Create Label
            app.Label = uilabel(app.LeftPanel);
            app.Label.Position = [33 196 287 44];
            app.Label.Text = {'For instructions on how to use the app as well
as'; 'the minimum distances and average masses, please'; 'use the ''Help''
button below. '};
            % Create kgLabel
            app.kgLabel = uilabel(app.LeftPanel);
            app.kgLabel.Position = [232 295 25 22];
            app.kgLabel.Text = 'kg';
            % Create kmLabel
            app.kmLabel = uilabel(app.LeftPanel);
            app.kmLabel.Position = [232 256 25 22];
            app.kmLabel.Text = 'km';
            % Create kgLabel_2
            app.kgLabel_2 = uilabel(app.LeftPanel);
            app.kgLabel_2.Position = [233 459 25 22];
            app.kgLabel_2.Text = 'kg';
            % Create CalculateButton
            app.CalculateButton = uibutton(app.LeftPanel, 'push');
            app.CalculateButton.ButtonPushedFcn = createCallbackFcn(app,
@CalculateButtonPushed, true);
            app.CalculateButton.Position = [125 82 100 23];
            app.CalculateButton.Text = 'Calculate';
            % Create Mass1EditFieldLabel
            app.Mass1EditFieldLabel = uilabel(app.LeftPanel);
            app.Mass1EditFieldLabel.HorizontalAlignment = 'right';
            app.Mass1EditFieldLabel.FontWeight = 'bold';
            app.Mass1EditFieldLabel.Position = [61 459 45 22];
            app.Mass1EditFieldLabel.Text = 'Mass 1';
            % Create Mass1EditField
            app.Mass1EditField = uieditfield(app.LeftPanel, 'numeric');
            app.Mass1EditField.ValueChangedFcn = createCallbackFcn(app,
@AstronomicalDropDownValueChanged, true);
            app.Mass1EditField.Editable = 'off';
            app.Mass1EditField.Position = [121 459 100 22];
            app.Mass1EditField.Value = 5.972e+24;
            % Create WhenenteringscientificnotationuselowercaseeEx5e1051010Label
```

```
            app.WhenenteringscientificnotationuselowercaseeEx5e1051010Label =
uilabel(app.LeftPanel);

app.WhenenteringscientificnotationuselowercaseeEx5e1051010Label.FontAngle =
'italic';

app.WhenenteringscientificnotationuselowercaseeEx5e1051010Label.Position = [35
130 270 44];
            app.WhenenteringscientificnotationuselowercaseeEx5e1051010Label.Text
= {'NOTE: When entering scientific notation'; 'use e or E instead of 10^'; 'Ex.
5e10 = 5*10^10'};
            % Create RightPanel
            app.RightPanel = uipanel(app.UIFigure);
            app.RightPanel.Position = [340 1 773 591];
            % Create UIAxes
            app.UIAxes = uiaxes(app.RightPanel);
            title(app.UIAxes, 'Gravitional Force vs Distance')
            xlabel(app.UIAxes, 'Distance (km)')
            ylabel(app.UIAxes, 'Gravitional Force (N)')
            zlabel(app.UIAxes, 'Z')
            app.UIAxes.Position = [31 24 715 388];
            % Create AppthatSimulatesGravityBetweenObjectsinSpaceASGBOSLabel
            app.AppthatSimulatesGravityBetweenObjectsinSpaceASGBOSLabel =
uilabel(app.RightPanel);

app.AppthatSimulatesGravityBetweenObjectsinSpaceASGBOSLabel.HorizontalAlignment
= 'center';
            app.AppthatSimulatesGravityBetweenObjectsinSpaceASGBOSLabel.FontSize
= 20;

app.AppthatSimulatesGravityBetweenObjectsinSpaceASGBOSLabel.FontWeight =
'bold';
            app.AppthatSimulatesGravityBetweenObjectsinSpaceASGBOSLabel.Position
= [0 529 773 62];
            app.AppthatSimulatesGravityBetweenObjectsinSpaceASGBOSLabel.Text =
{'App that Simulates Gravity Between Objects in Space'; '(ASGBOS) '};
            % Create GravitationalForceLabel
            app.GravitationalForceLabel = uilabel(app.RightPanel);
            app.GravitationalForceLabel.HorizontalAlignment = 'right';
            app.GravitationalForceLabel.Position = [31 446 104 22];
            app.GravitationalForceLabel.Text = 'Gravitational Force';
            % Create GravitationalForceEditField
            app.GravitationalForceEditField = uieditfield(app.RightPanel,
'numeric');
            app.GravitationalForceEditField.Editable = 'off';
            app.GravitationalForceEditField.Position = [153 446 98 22];
            % Create Newtonlabel
            app.Newtonlabel = uilabel(app.RightPanel);
            app.Newtonlabel.Position = [264 446 25 22];
```

```
            app.Newtonlabel.Text = 'N';
            % Create Names
            app.Names = uilabel(app.RightPanel);
            app.Names.HorizontalAlignment = 'center';
            app.Names.Position = [211 508 356 22];
            app.Names.Text = 'Seth Iris Canonigo, Nathan Huey, Lothar Escobar';
            % Show the figure after all components are created
            app.UIFigure.Visible = 'on';
        end
    end
    % App creation and deletion
    methods (Access = public)
        % Construct app
        function app = ASGBOS
            % Create UIFigure and components
            createComponents(app)
            % Register the app with App Designer
            registerApp(app, app.UIFigure)
            % Execute the startup function
            runStartupFcn(app, @startupFcn)
            if nargout == 0
                clear app
            end
        end
        % Code that executes before app deletion
        function delete(app)
            % Delete UIFigure when app is deleted
            delete(app.UIFigure)
        end
    end
end
```

Help App file

```
classdef HowToUse < matlab.apps.AppBase
    % Properties that correspond to app components
    properties (Access = public)
        UIFigure                        matlab.ui.Figure
        TabGroup                        matlab.ui.container.TabGroup
        HelpTab                         matlab.ui.container.Tab
        Gif4                            matlab.ui.control.Image
        Gif3                            matlab.ui.control.Image
        Gif2                            matlab.ui.control.Image
        NextButton                      matlab.ui.control.Button
        PreviousButton                  matlab.ui.control.Button
        Gif1                            matlab.ui.control.Image
        Instructions                    matlab.ui.control.Label
        HowtouseASGBOSLabel             matlab.ui.control.Label
        MinimumDistancesTab             matlab.ui.container.Tab
        AstronomicalBodyDropDownLabel   matlab.ui.control.Label
```

```
        MinimumDistanceEditField       matlab.ui.control.NumericEditField
        MinimumDistanceEditFieldLabel  matlab.ui.control.Label
        kmLabel                        matlab.ui.control.Label
        Background                     matlab.ui.control.Image
        AstronomicalBodyDropDown       matlab.ui.control.DropDown
        EarthLabel                     matlab.ui.control.Label
        MinimumDistancesLabel          matlab.ui.control.Label
        AverageMassesTab               matlab.ui.container.Tab
        MassoftheChicxulubasteroid461017kgLabel  matlab.ui.control.Label
        MassoftheMoon7341022Label      matlab.ui.control.Label
        MassofCeres911020kgLabel       matlab.ui.control.Label
        Averagemassofa1kmasteroid1311012kgLabel  matlab.ui.control.Label
        Averagemassofahuman62kgLabel   matlab.ui.control.Label
        AverageMassesLabel             matlab.ui.control.Label
    end

    properties (Access = private)
        Steps = [1:4]; % An array of all of the help steps

    end

    % Callbacks that handle component events
    methods (Access = private)
        % Code that executes after component creation
        function startupFcn(app)
            app.Steps = 1;
        end
        % Button pushed function: NextButton
        function NextButtonPushed(app, event)

            step = app.Steps;

            % Increments the step number from the Next buttom
            if (step < 4)
                step = step + 1;
                % Checks what the step is and gives the appropriate
                % instructions
                switch step
                    case 1
                     app.Gif1.Visible ="on";
                     app.Gif2.Visible ="off";
                     app.Gif3.Visible ="off";
                     app.Gif4.Visible ="off";
                     app.Instructions.Text = sprintf('Step 1: Choose an
astronomical body. \nChoosing a body will also give its mass');
                    case 2
                     app.Gif1.Visible ="off";
                     app.Gif2.Visible ="on";
                     app.Gif3.Visible ="off";
```

```
                    app.Gif4.Visible ="off";
                    app.Instructions.Text = sprintf('Step 2: Input another
mass(kg) and \nthe distance between the two \nmasses(kg)\n(Use the minimum
distance!)');

                case 3
                    app.Gif1.Visible ="off";
                    app.Gif2.Visible ="off";
                    app.Gif3.Visible ="on";
                    app.Gif4.Visible ="off";
                    app.Instructions.Text = sprintf('Step 3: The gravitional
force will be \noutputted on the right');

                case 4
                    app.Gif1.Visible ="off";
                    app.Gif2.Visible ="off";
                    app.Gif3.Visible ="off";
                    app.Gif4.Visible ="on";
                    app.Instructions.Text = sprintf('Step 4: The graph of the
forces \nwill be outputted on the right');
            end
        app.Steps = step;
        end

    end
    % Button pushed function: PreviousButton
    function PreviousButtonPushed(app, event)
       step = app.Steps;

        % Decrements the step number from the Next buttom
        if (step > 1)
            step = step - 1;
            switch step
                case 1
                    app.Gif1.Visible ="on";
                    app.Gif2.Visible ="off";
                    app.Gif3.Visible ="off";
                    app.Gif4.Visible ="off";
                    app.Instructions.Text = sprintf('Step 1: Choose an
astronomical body. \nChoosing a body will also give its mass');
                case 2
                    app.Gif1.Visible ="off";
                    app.Gif2.Visible ="on";
                    app.Gif3.Visible ="off";
                    app.Gif4.Visible ="off";
                    app.Instructions.Text = sprintf('Step 2: Input another
mass(kg) and \nthe distance between the two \nmasses(kg)\n(Use the minimum
distance!)');
```

```
                    case 3
                     app.Gif1.Visible ="off";
                     app.Gif2.Visible ="off";
                     app.Gif3.Visible ="on";
                     app.Gif4.Visible ="off";
                     app.Instructions.Text = sprintf('Step 3: The gravitional
force will be \noutputted on the right');

                    case 4
                     app.Gif1.Visible ="off";
                     app.Gif2.Visible ="off";
                     app.Gif3.Visible ="off";
                     app.Gif4.Visible ="on";
                end
            app.Steps = step;
            end

        end
        % Value changed function: AstronomicalBodyDropDown
        function AstronomicalBodyDropDownValueChanged(app, event)
            object = app.AstronomicalBodyDropDown.Value;

            % Outputs the minimum distances from the drop down menu
            switch object
                case 'Earth'
                 app.Background.ImageSource = imread('Earth.png');

                 mindist= 6471;

                case 'Mercury'
                 app.Background.ImageSource = imread('Mercury.png');

                 mindist= 2440;

                case 'Venus'
                 app.Background.ImageSource = imread('Venus.png');

                 mindist= 6050;

                case 'Mars'
                 app.Background.ImageSource = imread('Mars.png');

                 mindist= 3400;

                case 'Jupiter'
                 app.Background.ImageSource = imread('Jupiter.png');

                 mindist= 69911;
```

```
            case 'Saturn'
                app.Background.ImageSource = imread('Saturn.png');

                mindist= 58232;

            case 'Uranus'
                app.Background.ImageSource = imread('Uranus.png');

                mindist= 25559;

            case 'Neptune'
                app.Background.ImageSource = imread('Neptune.png');

                mindist= 24622;

            case 'Pluto'
                app.Background.ImageSource = imread('Pluto.png');

                mindist= 2188;

            case 'Sun'
                app.Background.ImageSource = imread('Sun.png');

                mindist= 700000;

        end
        app.MinimumDistanceEditField.Value = mindist;

    end
end
% Component initialization
methods (Access = private)
    % Create UIFigure and components
    function createComponents(app)
        % Get the file path for locating images
        pathToMLAPP = fileparts(mfilename('fullpath'));
        % Create UIFigure and hide until all components are created
        app.UIFigure = uifigure('Visible', 'off');
        app.UIFigure.Position = [100 100 640 480];
        app.UIFigure.Name = 'MATLAB App';
        % Create TabGroup
        app.TabGroup = uitabgroup(app.UIFigure);
        app.TabGroup.Position = [1 1 640 480];
        % Create HelpTab
        app.HelpTab = uitab(app.TabGroup);
        app.HelpTab.Title = 'Help';
        % Create HowtouseASGBOSLabel
        app.HowtouseASGBOSLabel = uilabel(app.HelpTab);
        app.HowtouseASGBOSLabel.FontSize = 18;
```

```
            app.HowtouseASGBOSLabel.FontWeight = 'bold';
            app.HowtouseASGBOSLabel.Position = [228 406 183 23];
            app.HowtouseASGBOSLabel.Text = 'How to use ASGBOS';
            % Create Instructions
            app.Instructions = uilabel(app.HelpTab);
            app.Instructions.Position = [49 137 210 207];
            app.Instructions.Text = {'Step 1: Choose an astronomical body. ';
'Choosing a body will also give its mass'};
            % Create Gif1
            app.Gif1 = uiimage(app.HelpTab);
            app.Gif1.Position = [321 137 272 207];
            app.Gif1.ImageSource = fullfile(pathToMLAPP, 'Step1.gif');
            % Create PreviousButton
            app.PreviousButton = uibutton(app.HelpTab, 'push');
            app.PreviousButton.ButtonPushedFcn = createCallbackFcn(app,
@PreviousButtonPushed, true);
            app.PreviousButton.Position = [49 76 100 23];
            app.PreviousButton.Text = 'Previous';
            % Create NextButton
            app.NextButton = uibutton(app.HelpTab, 'push');
            app.NextButton.ButtonPushedFcn = createCallbackFcn(app,
@NextButtonPushed, true);
            app.NextButton.Position = [493 76 100 23];
            app.NextButton.Text = 'Next';
            % Create Gif2
            app.Gif2 = uiimage(app.HelpTab);
            app.Gif2.Visible = 'off';
            app.Gif2.Position = [321 137 272 207];
            app.Gif2.ImageSource = fullfile(pathToMLAPP, 'Step2.gif');
            % Create Gif3
            app.Gif3 = uiimage(app.HelpTab);
            app.Gif3.Visible = 'off';
            app.Gif3.Position = [321 137 272 207];
            app.Gif3.ImageSource = fullfile(pathToMLAPP, 'Step3.gif');
            % Create Gif4
            app.Gif4 = uiimage(app.HelpTab);
            app.Gif4.Visible = 'off';
            app.Gif4.Position = [321 137 272 207];
            app.Gif4.ImageSource = fullfile(pathToMLAPP, 'Step4.gif');
            % Create MinimumDistancesTab
            app.MinimumDistancesTab = uitab(app.TabGroup);
            app.MinimumDistancesTab.Title = 'Minimum Distances';
            % Create MinimumDistancesLabel
            app.MinimumDistancesLabel = uilabel(app.MinimumDistancesTab);
            app.MinimumDistancesLabel.HorizontalAlignment = 'center';
            app.MinimumDistancesLabel.FontSize = 18;
            app.MinimumDistancesLabel.FontWeight = 'bold';
            app.MinimumDistancesLabel.Position = [2 406 638 23];
            app.MinimumDistancesLabel.Text = 'Minimum Distances';
```

```
            % Create EarthLabel
            app.EarthLabel = uilabel(app.MinimumDistancesTab);
            app.EarthLabel.Position = [85 318 469 44];
            app.EarthLabel.Text = {'Because the formula for Newton''s Law of
Gravitation works for two center of masses, '; 'you also must take account the
radius, or the distance from the core to the outermost'; 'atmosphere of the
astronomical body.'};
            % Create AstronomicalBodyDropDown
            app.AstronomicalBodyDropDown = uidropdown(app.MinimumDistancesTab);
            app.AstronomicalBodyDropDown.Items = {'Earth', 'Mercury', 'Venus',
'Mars', 'Jupiter', 'Saturn', 'Uranus', 'Neptune', 'Pluto', 'Sun'};
            app.AstronomicalBodyDropDown.ValueChangedFcn =
createCallbackFcn(app, @AstronomicalBodyDropDownValueChanged, true);
            app.AstronomicalBodyDropDown.Position = [274 226 88 22];
            app.AstronomicalBodyDropDown.Value = 'Earth';
            % Create Background
            app.Background = uiimage(app.MinimumDistancesTab);
            app.Background.Position = [467 140 126 122];
            app.Background.ImageSource = fullfile(pathToMLAPP, 'Earth.png');
            % Create kmLabel
            app.kmLabel = uilabel(app.MinimumDistancesTab);
            app.kmLabel.Position = [386 190 25 22];
            app.kmLabel.Text = 'km';
            % Create MinimumDistanceEditFieldLabel
            app.MinimumDistanceEditFieldLabel =
uilabel(app.MinimumDistancesTab);
            app.MinimumDistanceEditFieldLabel.HorizontalAlignment = 'right';
            app.MinimumDistanceEditFieldLabel.FontWeight = 'bold';
            app.MinimumDistanceEditFieldLabel.Position = [148 190 111 22];
            app.MinimumDistanceEditFieldLabel.Text = 'Minimum Distance';
            % Create MinimumDistanceEditField
            app.MinimumDistanceEditField = uieditfield(app.MinimumDistancesTab,
'numeric');
            app.MinimumDistanceEditField.Editable = 'off';
            app.MinimumDistanceEditField.Position = [274 190 100 22];
            app.MinimumDistanceEditField.Value = 6471;
            % Create AstronomicalBodyDropDownLabel
            app.AstronomicalBodyDropDownLabel =
uilabel(app.MinimumDistancesTab);
            app.AstronomicalBodyDropDownLabel.HorizontalAlignment = 'right';
            app.AstronomicalBodyDropDownLabel.FontWeight = 'bold';
            app.AstronomicalBodyDropDownLabel.Position = [148 226 115 22];
            app.AstronomicalBodyDropDownLabel.Text = 'Astronomical Body';
            % Create AverageMassesTab
            app.AverageMassesTab = uitab(app.TabGroup);
            app.AverageMassesTab.Title = 'Average Masses';
            % Create AverageMassesLabel
            app.AverageMassesLabel = uilabel(app.AverageMassesTab);
            app.AverageMassesLabel.HorizontalAlignment = 'center';
```

```
            app.AverageMassesLabel.FontSize = 18;
            app.AverageMassesLabel.FontWeight = 'bold';
            app.AverageMassesLabel.Position = [2 406 638 23];
            app.AverageMassesLabel.Text = 'Average Masses';
            % Create Averagemassofahuman62kgLabel
            app.Averagemassofahuman62kgLabel = uilabel(app.AverageMassesTab);
            app.Averagemassofahuman62kgLabel.FontSize = 14;
            app.Averagemassofahuman62kgLabel.Position = [52 343 211 22];
            app.Averagemassofahuman62kgLabel.Text = 'Average mass of a human: 62
kg';
            % Create Averagemassofa1kmasteroid1311012kgLabel
            app.Averagemassofa1kmasteroid1311012kgLabel =
uilabel(app.AverageMassesTab);
            app.Averagemassofa1kmasteroid1311012kgLabel.FontSize = 14;
            app.Averagemassofa1kmasteroid1311012kgLabel.Position = [52 297 310
22];
            app.Averagemassofa1kmasteroid1311012kgLabel.Text = 'Average mass of
a 1 km asteroid: 1.31*10^12 kg';
            % Create MassofCeres911020kgLabel
            app.MassofCeres911020kgLabel = uilabel(app.AverageMassesTab);
            app.MassofCeres911020kgLabel.FontSize = 14;
            app.MassofCeres911020kgLabel.Position = [52 190 184 22];
            app.MassofCeres911020kgLabel.Text = 'Mass of Ceres: 9.1*10^20 kg';
            % Create MassoftheMoon7341022Label
            app.MassoftheMoon7341022Label = uilabel(app.AverageMassesTab);
            app.MassoftheMoon7341022Label.FontSize = 14;
            app.MassoftheMoon7341022Label.Position = [52 240 195 22];
            app.MassoftheMoon7341022Label.Text = 'Mass of the Moon: 7.34*10^22';
            % Create MassoftheChicxulubasteroid461017kgLabel
            app.MassoftheChicxulubasteroid461017kgLabel =
uilabel(app.AverageMassesTab);
            app.MassoftheChicxulubasteroid461017kgLabel.FontSize = 14;
            app.MassoftheChicxulubasteroid461017kgLabel.Position = [52 140 289
22];
            app.MassoftheChicxulubasteroid461017kgLabel.Text = 'Mass of the
Chicxulub asteroid: 4.6*10^17 kg';
            % Show the figure after all components are created
            app.UIFigure.Visible = 'on';
        end
    end
    % App creation and deletion
    methods (Access = public)
        % Construct app
        function app = HowToUse
            % Create UIFigure and components
            createComponents(app)
            % Register the app with App Designer
            registerApp(app, app.UIFigure)
            % Execute the startup function
```

```
            runStartupFcn(app, @startupFcn)
            if nargout == 0
                clear app
            end
        end
        % Code that executes before app deletion
        function delete(app)
            % Delete UIFigure when app is deleted
            delete(app.UIFigure)
        end
    end
end
```