



National Aeronautics and
Space Administration



LUNAR CODING CHALLENGE



NASA STEM
INSPIRE - ENGAGE - EDUCATE - EMPLOY
The Next Generation of Explorers

LUNAR CODING CHALLENGE

The goal of this challenge is to utilize block code to design the Artemis I mission. Artemis I is scheduled to launch no earlier than November 14, 2022 and is the first mission as part of the Artemis program to return astronauts to the Moon. To be able to complete this challenge, students will need access to a block code platform. In this guide, Code.org's "App Lab" is used as an example, however, any block code platform can be utilized.



CODE.ORG

Code.org's "App Lab" can be accessed through the following link: <https://code.org/educate/applab>. Students will be given the option to choose from three different avatars of the Orion capsule. All resources to complete the app can be found in the file downloaded from the App Development Challenge Website.

Part 1: Designing the App

1. Under "Design" on top of the phone panel:
 - a. Click the "screen" tool in the Design Toolbox and drag it on top of the phone panel
 - i. Go to the Workspace, click "Properties" and set the RGB to (0,0,0)
2. Go back to the Toolbox, then drag the "Image" icon on top of the phone panel.
 - a. Under "Properties", under id type in "Earth"
 - b. Set the width (px) to 100, and height (px) to 100
 - c. Set the x position (px) to 230, set the y position (px) to 5
 - d. Set the image to the "The_Earth_seen_from_Apollo.." file that you downloaded.
 - e. Leave everything else (fit image, border width, border color) as is.
3. Go back to the Toolbox, then drag the "Image" icon on top of the phone screen again.
 - a. Under "Properties", under id type in "Moon"
 - b. Set the width (px) to 100, and height (px) to 100.
 - c. Set the x-position to 20, y-position to 325
 - d. Set the image to the "moon.0001.jpg" file you downloaded earlier.
 - e. Keep everything else as-is.
4. Repeat step 3, but now add Orion with the following:
 - a. Under "Properties", under id type in "Orion"
 - b. Set the width (px) to 100, height (px) 100
 - c. X-position (170), y-position (-5)
 - d. Image ID will be the provided Orion spacecraft image.
5. Final image – repeat the previous steps, but now add the App Development Challenge (ADC) logo.
 - a. Under "Properties", under ID type in "ADC"
 - b. Set the width (px) to 100, height (px) to 100
 - c. X-position to 110, y-position to 180
 - d. Image set to the ADC logo

6. Let's start adding our mission waypoints. Drag another image icon onto the phone screen as with the previous steps.
 - a. Set the ID type to "waypoint1"
 - b. Set the width and height to 105 px and 25 px, respectively.
 - c. Set the x-position to 105 px, y-position to 25 px.
 - d. Under "image", click "Icons". Under icons, find the circle icon and click it.
 - e. Set the icon color to #ff0000
7. For waypoint2, repeat the same as in step 6, make sure the ID reads "waypoint2" and set the x-position to 140 px and y-position to 105 px.
 - a. Waypoint3 same steps but set the x-position to 190 px and y-position to 275 px.
 - b. Waypoint4 same steps but set the x-position to 4 px and y-position to 385 px.
8. Now add a start button. Go back to the Design Toolbox and drag a button element onto the phone screen anywhere between the Earth and the Moon. Under properties, make sure the ID is named "Button" and set text to "Begin Mission".
9. Let's start adding our mission critical text. These will appear at each waypoint when reached by Orion.
 - a. Go to the Design Toolbox. Drag the "Label" tool onto the phone screen. Set the ID to "waypoint1_text". Under the text box, write: "Orion leaves Earth orbit and begins its journey to the Moon." Set the width to 147 px, height 100 px. X-position to -5 px, Y-position to 5 px. Text color to rgb (255, 255, 255). Background color rgb (0,0,0,0). Font family to Arial. Font size 13 px. Text alignment to the left. Border color rgb (77,87,95).
 - b. Repeat the steps above for the second waypoint, only this time set the ID to "waypoint2_text", the text should read "Orion adjusts its course. Closest lunar approach now is now 60 miles." Width 147 px. Height 100 px. X-position 35 px. Y-position 100 px.
 - c. Repeat the steps above for the third waypoint, only this time set the ID to "waypoint3_text", text should read, "Orion coasts for 5 days and increases its distance from Earth to 240,000 miles." width 147 px, height 100 px, x-position 80 px, y-position 215 px.
 - d. Repeat the steps above for the fourth waypoint. Only this time set the ID to "waypoint4_text", text should read "Orion reaches close lunar approach of 60 miles, then burns its engine to enter into an elliptical orbit around the Moon." Width 147 px. Height 100 px. X position 85 px. Y position 345 px.

Creating the Code

1. Now it's time to start designing the code to make the app work. First thing we'll want to do is make sure all the text disappears when the app starts. The language being used by the app lab is known as Javascript!
 - a. In the toolbox, under "UI controls", find the command "hideElement(id)". Drag this over into the workspace. Set the ID from "id" to "waypoint1_text".
 - i. Repeat this step 5 more times, but set the ID to "waypoint2_text", "waypoint3_text", "waypoint4_text", "ADC_text", and "ADC".

```
hideElement(▼ "waypoint1_text");
hideElement(▼ "waypoint2_text");
hideElement(▼ "waypoint3_text");
hideElement(▼ "waypoint4_text");
hideElement(▼ "ADC_text");
hideElement(▼ "ADC");
```

2. Time to start the first actual action element of the code. We'll start with the start button. We want it to disappear and make a sound when pressed. Under "UI controls" find the command "onEvent(id)..." and drag it to the workspace. Set the first element id to "button", and "click".
 - a. Find the command playSound, drag it to being below the onEvent command (in the green, representing a function) and set the sound file to "sound://category_tap/level_select_1.mp3", and loop to false. This will generate a sound when the app begins once you click the button.
 - b. Below the "playSound" command, drag two "hideElement" commands, set the first one to be the "button". This will cause the button to disappear after being pressed. sound://category_tap/level_select_1.mp3

```
onEvent(▼ "button", ▼ "click", function(○) {
  playSound(▼ "sound://category_tap/level_select_1.mp3", ▼ false);
  hideElement(▼ "button");
  hideElement(▼ "labell1");
})
```

3. Now let's get Orion moving. The goal is to design an app that reaches waypoints. These waypoints will turn from red to green and display mission-related facts as you reach them. Use the code toolbox to set the code up to be like this. The goal here is to set Orion to match and move over to the x and y coordinates of Waypoint1 while displaying waypoint1_text. The sound file is a "whoosh" sound that'll play whenever Orion moves. The third "setProperty" line changes the icon-color of the waypoint from "red" to "green".

```
onEvent(▼ "Orion", ▼ "click", function(○) {
  playSound(▼ "sound://category_whoosh/animation_whoosh13.mp3", ▼ false);
  setProperty(▼ "Orion", ▼ "x", getProperty(▼ "waypoint1", ▼ "x"));
  setProperty(▼ "Orion", ▼ "y", getProperty(▼ "waypoint1", ▼ "y"));
  setProperty(▼ "waypoint1", ▼ "icon-color", ▼ "green");
  showElement(▼ "waypoint1_text");
})
```

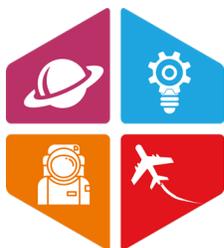
4. Now let's move to Waypoint 2 and 3. Repeat the steps above and ensure your code looks like the below.

```
onEvent(▼ "Orion", ▼ "click", function(○) {
  setProperty(▼ "Orion", ▼ "x", getProperty(▼ "waypoint2", ▼ "x"));
  setProperty(▼ "Orion", ▼ "y", getProperty(▼ "waypoint2", ▼ "y"));
  setProperty(▼ "waypoint2", ▼ "icon-color", ▼ "green");
  showElement(▼ "waypoint2_text");
  onEvent(▼ "Orion", ▼ "click", function(○) {
    setProperty(▼ "Orion", ▼ "x", getProperty(▼ "waypoint3", ▼ "x"));
    setProperty(▼ "Orion", ▼ "y", getProperty(▼ "waypoint3", ▼ "y"));
    setProperty(▼ "waypoint3", ▼ "icon-color", ▼ "green");
    showElement(▼ "waypoint3_text");
    onEvent(▼ "Orion", ▼ "click", function(○) {

```

5. Let's set up the final waypoint. We're going to set up the procedures to move Orion to the 4th waypoint exactly the same, but we want to add some additional steps. Now we want to clean up the app a bit with the extra text. We're going to want to re-hide the text labels again using the "hideElement" tool. We also want to display the App Development Challenge (ADC) logo and direct app users to the website to learn more! The link to the ADC website for you to use is https://www.nasa.gov/stem/nextgenstem/moon/app_challenge.html

```
onEvent (▼ "Orion", ▼ "click", function (●) {  
  setProperty (▼ "Orion", ▼ "x", getProperty (▼ "waypoint4", ▼ "x"));  
  setProperty (▼ "Orion", ▼ "y", getProperty (▼ "waypoint4", ▼ "y"));  
  setProperty (▼ "waypoint4", ▼ "icon-color", ▼ "green");  
  showElement (▼ "waypoint4_text");  
  hideElement (▼ "waypoint1_text");  
  hideElement (▼ "waypoint2_text");  
  hideElement (▼ "waypoint3_text");  
  hideElement (▼ "waypoint4_text");  
  showElement (▼ "ADC");  
  open ("https://www.nasa.gov/stem/nextgenstem/moon/app_challenge.html");  
});
```

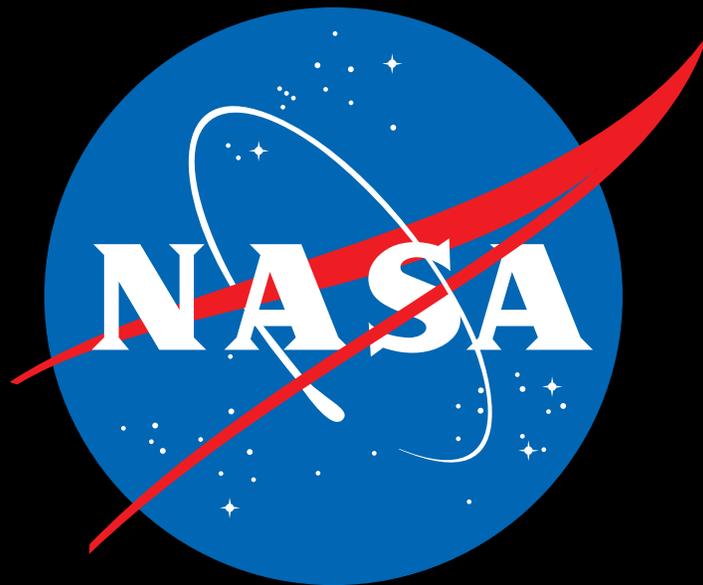


NASA STEM

INSPIRE - ENGAGE - EDUCATE - EMPLOY
The Next Generation of Explorers



There's always more to explore at stem.nasa.gov/
Click the link or scan the QR code with your smartphone to get started.



National Aeronautics and Space
Administration

NASA Headquarters

300 E Street SW
Washington, DC 20546
www.nasa.gov/centers/hq

www.nasa.gov