CesiumJS: Fundamentals

The CesiumJS: Fundamentals learning path will teach you the core features needed to create an application with CesiumJS.

Learning objectives:

- Understand CesiumJS's background and how you can use it
- Understand and implement loading of imagery, terrain, 3D tilesets, and various data sources
- Understand and implement entities
- Understand and implement camera controls

Lesson 1: Getting Started

Introduction:

In this lesson, we will cover the background of CesiumJS, how you can use it, and some background knowledge you may need.

What is CesiumJS?

CesiumJS is a JavaScript library for creating 3D globes and 2D maps in a web browser without a plugin. It uses WebGL for hardware-accelerated graphics, and it is cross-platform and cross-browser. We have also tuned CesiumJS for dynamic data visualization and high-precision analysis on the WGS84 globe.

- We will share some additional resources at the end of this lesson if you are still getting familiar with any geospatial concepts we mention, such as WGS84.

On top of that, we have built CesiumJS to be flexible and interoperable with other web frameworks. As a result, many developers use CesiumJS in use cases with various frameworks, such as react, view, and angular, and build tools such as webpack, esbuild, and vite.

Do I need to know anything before I can start learning CesiumJS?

Before working through this learning path, you should have a basic understanding of the following areas:

- Web development, primarily Javascript
- Geospatial concepts
- 3D graphics concepts

If you're unsure about your experience level, don't worry, we'll do our best to provide you with the resources needed to succeed at each stage. If you get stuck at any point, feel free to explore the additional resources provided and return to the lesson when ready. If you have questions, the Community Forum is an excellent resource for developers of all experience levels.

The two assignments for this lesson will take you through essential concepts needed for CesiumJS, including access tokens, accessing data into Cesium ion, installing with npm, and time series data visualization.

Assignment:

- Complete the CesiumJS Quickstart tutorial.
- Complete the Flight Tracker tutorial.

Additional Resources:

This section contains helpful links related to the lesson's content that might be interesting or helpful.

- If you want a refresher on web development concepts, try exploring The Odin Project or freeCodeCamp.org to see if there are any concepts you wish to learn or relearn.
- If you want a quick overview of geospatial concepts, this primer by mapschool.io is a good starting point.
- A basic understanding of 3D graphics concepts could help you tackle problems in CesiumJS. If you want a crash course on the basic concepts, try CrashCourse's 12-minute video.

Knowledge Check:

This section contains questions to check your understanding of this lesson. If you're having a problem answering a question, click the link and review the content again.

- Which graphics library allows CesiumJS to render a 3D globe in your browser?
- What are some benefits of visualizing flight data on a 3D globe rather than a 2D map?
- When should you use a 3D model versus a 3D tileset?

Lesson 2: Using your data with CesiumJS

Introduction:

This lesson will explore loading and utilizing different data types within CesiumJS.

How do I import 3D models?

One of the benefits of using CesiumJS is that it allows you to use combinations of different types of data, including 3D models. If you completed the Flight Tracker tutorial in the previous lesson, you already have some experience with loading 3D models.

As shown in the Flight Tracker tutorial, the simplest way to show how your 3D models can be used is through Cesium ion, which allows you to position your model's geographic location once uploaded.

- CesiumJS allows for alternatives if you prefer to host your data outside Cesium ion.

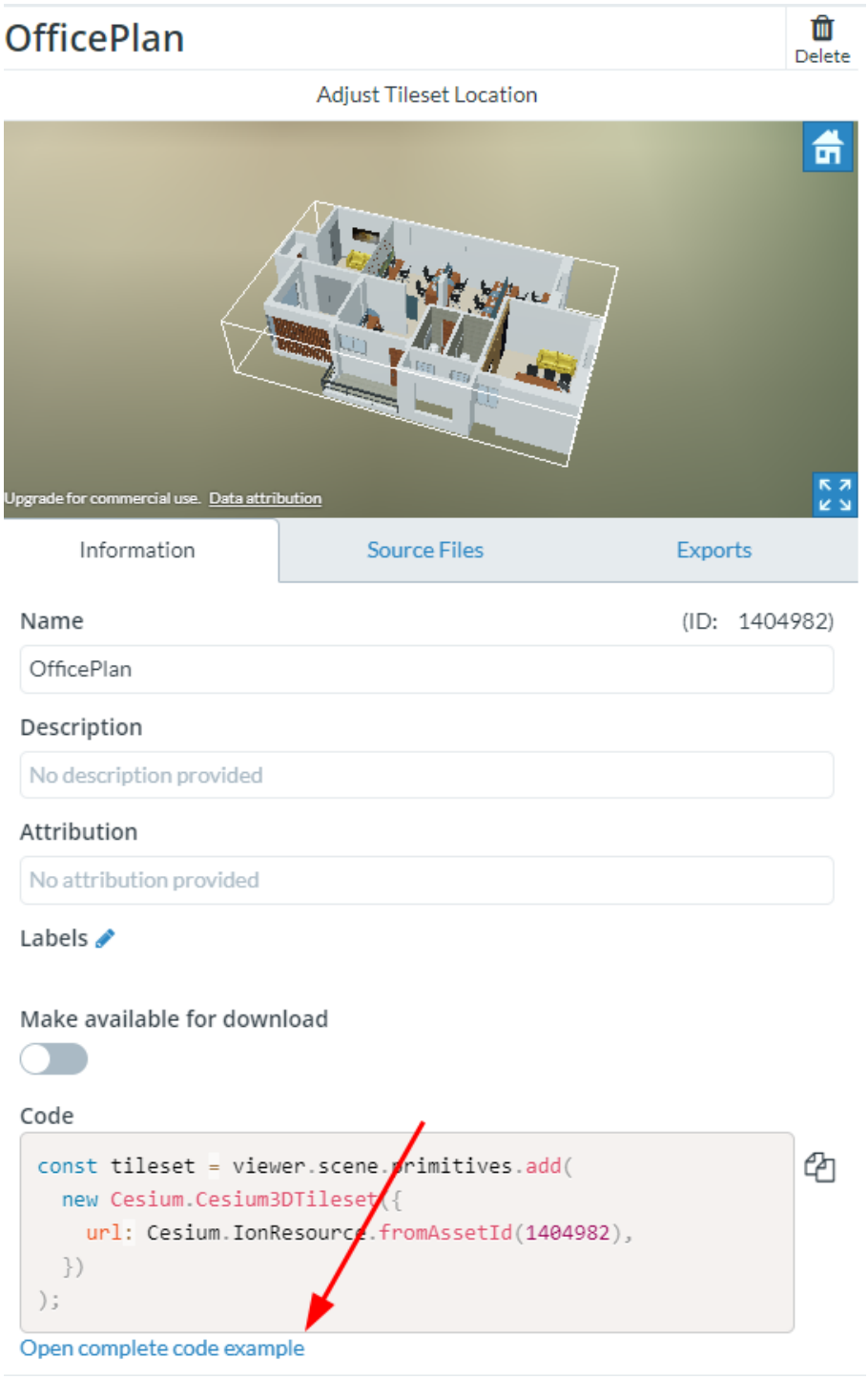
Cesium ion supports a range of 3D model types, as shown below:

Format	File extensions
Wavefront OBJ	.obj
Filmbox	.fbx
Digital Asset Exchange	.dae
glTF	.gltf
Binary glTF	.glb

If you want to practice loading 3D models into Cesium ion so they can be used with your CesiumJS application, follow the steps in the short tutorials below. Once you are comfortable loading your 3D models, we'll move on to other data types.

- Loading 3D models
- Loading 3D buildings

Be sure to click the "Open complete code example" option to see how your 3D model looks once loaded.



How do I import terrain and imagery?

Now that you're familiar with loading 3D models, it will be straightforward also to load terrain and imagery for use in CesiumJS. The main difference between imagery and terrain data compared to 3D models is that you must georeference your imagery and terrain before loading it into Cesium ion. Cesium ion accepts various formats for both types of data, as shown below:

Accepted terrain formats:

Format	File extensions
GeoTIFF	.tiff, .tif
Floating Point Raster	.flt
Arc/Info ASCII Grid	.asc
Source Map	.src
Erdas Imagine	.img
USGS ASCII DEM and CDED	.dem
Cesium Terrain Database	.terraindb
Other common raster formats	

Accepted imagery formats:

Format	File extensions
GeoTIFF	.tiff, .tif
Floating Point Raster	.flt
Arc/Info ASCII Grid	.asc
Source Map	.src
Erdas Imagine	.img
USGS ASCII DEM and CDED	.dem
JPEG	.jpg, .jpeg
PNG	.png
Other common raster formats	

Follow the short tutorials below to see how both data types can be loaded into Cesium ion and pulled into a CesiumJS application.

- Loading terrain data
- Loading imagery data

Assignment:

- Complete the Visualize a Proposed Building in a 3D City tutorial.

- Complete the Visualizing 3D Terrain tutorial.
- Complete the Visualizing Imagery tutorial.

Additional Resources:

This section contains helpful links related to the lesson's content that might be interesting or helpful.

- If you need help getting your data to be correctly georeferenced, follow this short guide.
- If you want more examples of 3D models used in CesiumJS, look at this Sandcastle example.
- If you are looking for more Sandcastle examples to show how you can manipulate imagery layers, look at these links:
    - Imagery Adjustments
    - Imagery Layer Manipulation

Knowledge Check:

This section contains questions to check your understanding of this lesson. If you're having a problem answering a question, click the link and review the content again.

- Why is it essential that your terrain and imagery data be georeferenced?
- Do you need to unzip 3D models before you upload them into Cesium ion?
- How can you view your 3D models, terrain, and imagery data before loading them into your CesiumJS application?

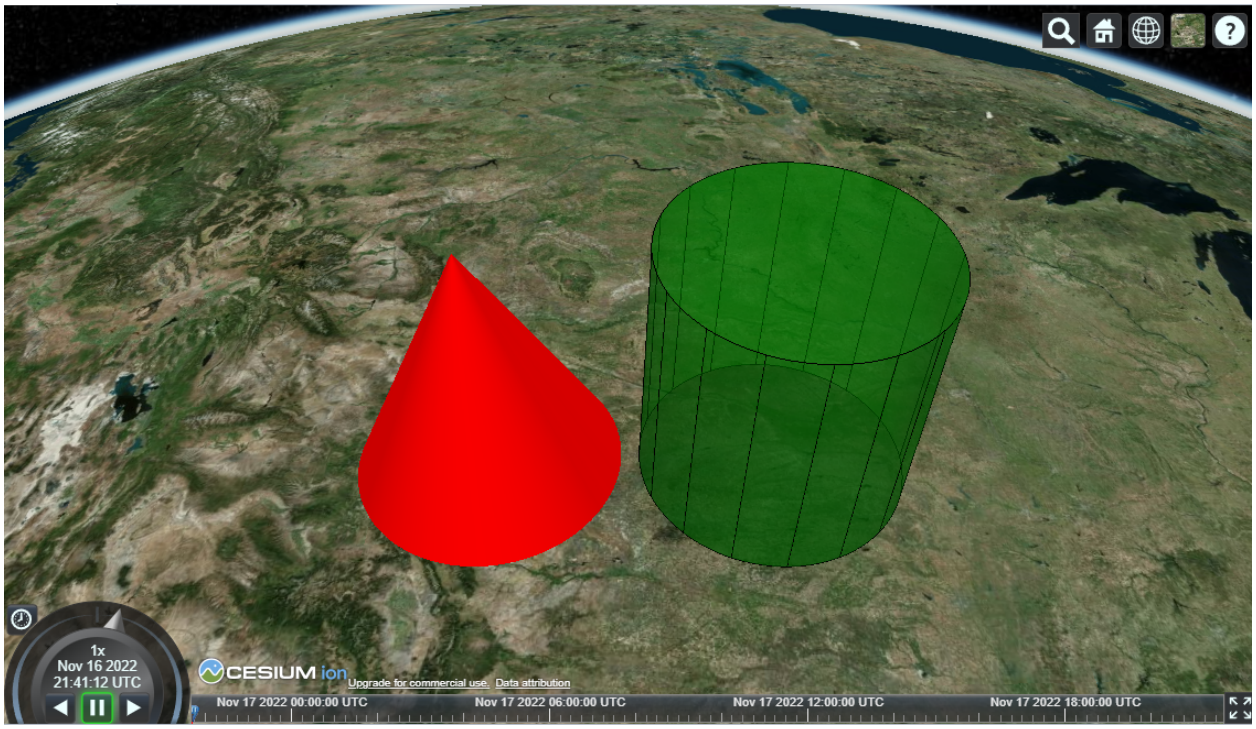
Lesson 3: Entities in CesiumJS

Introduction:

This lesson will cover the basics of creating and manipulating entities in CesiumJS.

What are Entities?

Entities are high-level visualizations you can make in CesiumJS, such as points, markers, shapes, and labels, which you can incorporate into your application. Unlike Primitives, which we will look at in a later lesson, Entities operate consistently through a unified data structure, making them simpler to use in most applications.



If you have completed the previous CesiumJS lessons, you have already created and interacted with Entities through the assigned tutorials.

How can I use Entities in my application?

Entities enable you to visualize concepts and data on the 3D globe in the simplest way possible. For example, if you want to highlight a portion of the map or call attention to a model moving across the screen, you will need to use Entities to make that happen.

In the assignment for this lesson, you will follow along with the Creating Entities tutorial, which will demonstrate all of the properties available through the Entity API. Once you have completed the tutorial, look through the additional resources to see how you can apply these concepts in interactive applications.

Assignment:

- Complete the Creating Entities tutorial.

Additional Resources:

This section contains helpful links related to the lesson's content that might be interesting or helpful.

- You can read more if you want to dig deeper into Entities' documentation.

- If you would like to explore code examples of Entities, follow the links below:
    - Points
    - Labels
    - Billboards
    - Show or Hide Entities

Knowledge Check:

This section contains questions to check your understanding of this lesson. If you're having a problem answering a question, click the link and review the content again.

- What is the default height of an Entity if it is `undefined`?
- What is the difference between `zoomTo` and `flyTo` methods?
- What is the difference between `scene.pick` and `scene.drillPick`?
- What is the difference between a `point` and a `billboard`?

Lesson 4: Camera Controls in CesiumJS

Introduction:

This lesson will cover camera controls in CesiumJS.

Why are camera controls necessary?

Even though we built CesiumJS to allow users to interact with the globe directly, additional features in your application that limit or take control of the user's camera are often necessary. Without these controls existing in CesiumJS, all camera controls would need to be built from scratch, reducing the time available for other critical features.

How can I control the camera?

We already touched on a few basic controls for the camera in Lesson 3, such as `flyTo` and `zoomTo`, but those are only two of the methods available.

The camera methods also provide developers with tools to change and rebind the default camera controls to UI elements or keystrokes. Watch the video below to see how you can use these controls to create the right experience for your application.

- [Insert a video that demonstrates the types of camera controls that are available]

Assignment:

- Complete the Control the camera tutorial.
- Parse through the Sandcastle Camera example, taking notes on constructing each camera function.

Additional Resources:

This section contains helpful links related to the lesson's content that might be interesting or helpful.

- If you want to read through the documentation for the camera controls, start here.
- Explore the Sandcastle example that looks at manipulating camera controls with custom keybindings.

Knowledge Check:

This section contains questions to check your understanding of this lesson. If you're having a problem answering a question, click the link and review the content again.

- How do the `camera.setView` and `camera.flyTo` methods differ?
- What is an easing function?

Practical Application

Now that you understand the fundamental concepts in CesiumJS, you need to put them into practice. Therefore, before progressing onto the next Learning Path, create a CesiumJS application that utilizes the following concepts:

1. Import a 3D model
2. Import terrain or imagery
3. Create at least three different types of Entities
4. Implement camera controls that move between all of your Entities
    a. Try to use a different method for each one

Your application can be a standalone application, or you can use the Hello World Sandcastle example to start your application.