# CesiumJS Web App Development
# AI Assistant Prompting Guidelines

*Adapted from the BUILD Framework developed by Rob Thomas; to be used in conjunction with the "Build a 3D Geospatial Web App with CesiumJS" step-by-step app development guide*

## Overview of the BUILD Framework

The BUILD Framework is a structured, beginner-friendly approach to building web apps. It can be especially helpful for developers who are using AI assistance to build. When developing web apps with CesiumJS using AI assistance, the phases of the BUILD Framework can support developers in building with intention, evaluating and iterating on AI outputs, and refining their apps with the end user in mind.

The BUILD Framework follows three phases:

**Phase 1 | Plan & Research:** Before writing any code, define what you're building, who it's for, and what success looks like. You'll identify your audience, prioritize features, and learn the key terms you'll need going forward. This phase keeps projects focused and grounded before a single line of code is written.

**Phase 2 | Build with AI:** This is where the app comes to life. You build one feature at a time, using AI as a coding partner. Rather than copying and pasting code indiscriminately, builders are encouraged to ask what each part does, why it's written that way, and what would break if it changed. The goal isn't just a working app, it's understanding what was built.

**Phase 3 | Test & Polish:** The full user experience is tested, problems are tracked down and fixed, and the project is documented: what it does, who it's for, and what was learned. This phase is what turns a rough build into something presentable and explainable.

## About This Document

Guided by the BUILD Framework, this document provides scenario-based prompting assistance for developers who are building 3D geospatial web apps with CesiumJS. The suggested prompts connect to different phases of the development cycle. The document is meant to complement the core technical step-by-step guidelines, and help developers envision and begin building a project; prompt the AI assistant effectively; get "unstuck" if they are unsure how to proceed; and refine their outputs.

# Prompting Guidelines for Building a 3D Geospatial Web App with CesiumJS Using AI Assistance

# PHASE 1: PLAN & RESEARCH

## Access the CreateAccess CesiumJS Gemini Gem through: bit.ly/createaccess-gem

*All prompts in this phase support THE **GEMINI GEM** INTERACTIONS.*

## 1A. Getting started

## Scenario: You have an idea but no data

You know the subject or theme you want to explore but haven't found a dataset or API yet.

None

```
I want to build a CesiumJS project. I have an idea of what I want to make
but I don't have data yet and I'm not sure where to find it.

My idea: [describe it – the subject, what you want someone to see or do,
who it's for]

Help me figure out what data could power this. I'd prefer to use a live
API over a static file if one exists.
```

## Scenario: You have both an idea and a dataset or API

You know what you want to build and you already have data in hand or know where it's coming from.

None

```
I want to build a CesiumJS project. I have an idea and I already have
data for it.

My idea: [describe it – what it shows, who it's for, what you want
someone to feel or understand]

My data: [describe the dataset or API what it contains, where it comes
from, the format if you know it] or [paste the link and ask]: What does
this data contain and how can I use it in a CesiumJS project?

I'd like to use an API instead of a static file if one is available for
this data. Help me validate whether my data actually supports my idea and
figure out the best approach.
```

## 1B. Data exploration

### Scenario: The Gem suggested an API and you want to understand it before committing

The Gem proposed a specific API and you want to know what it actually returns before deciding to build around it.

```
None
Before I commit to this API/Dataset, help me understand it fully. Walk me through:

- What endpoints I'll actually use

- What a real response looks like — what fields come back and which ones matter for my project

- How that data flows from the API call to something visible on the globe

- Any limitations I should know about — rate limits, CORS, authentication, quirks

I want to understand the data before I decide to build around it.
```

### Scenario: You found an API yourself and want the Gem to evaluate it

You've found a potential data source and want to know if it's a good fit before going further.

```
None
I found an API I think could work: [paste the URL or describe it]

Can you look at this and tell me:

- Whether it's actually a good fit for my project idea

- What the useful endpoints and fields are

- How the data would flow from this API to the globe

- Any problems I should know about before I build around it

- Whether there's a better alternative I should consider instead
```

### Scenario: No good API exists and you need to fall back to a static file

You've explored APIs and they don't exist or aren't usable for your project.

None

```
It sounds like there isn't a good live API for this. Let's talk through the static file option.

What format would work best for my project — GeoJSON, CZML, KML, or something else? And where would I host it so the browser can actually load it? Walk me through what that setup looks like compared to an API approach.
```

### Scenario: Your data needs pre-processing before CesiumJS can use it

You have a file or API but something about it needs to be fixed — wrong format, no coordinates, too large, etc.

None

```
It sounds like my data needs some work before CesiumJS can use it. Help me understand:

- Exactly what needs to change — format conversion, coordinate issues, field cleanup, or something else

- How much effort that is and whether it's worth doing vs. finding a different source

- Whether Cesium ion can handle any of this automatically if I upload it there

I want to understand the cost before I commit to this data source.
```

## 1C. Concept and scope

### Scenario: You want the Gem to propose what your data could show

You have data confirmed but you're not sure what the best visualization concept is.

None

```
Now that we understand the data, help me figure out what to actually build with it. What are 2 or 3 genuinely different things this data could show in CesiumJS — and for each one, tell me what specifically CesiumJS or 3D adds that a flat map wouldn't?

I want concepts that fit what the data actually supports, not wishful thinking.
```

### Scenario: You have a concept in mind and want a reality check

You know what you want to build but want to pressure-test it before investing time in a spec.

None

```
Here's the concept I want to build: [describe it — what the user sees, what they do, what the data drives]

Be honest with me — does my data actually support this? Are there gaps or mismatches I haven't thought about? And is there a simpler version that gets most of the value if the full concept is too complex for a first build?
```

## Scenario: You want to understand what CesiumJS can actually do with your data type

You're new enough to CesiumJS that you don't know what's possible for your kind of data.

None

```
I'm not sure what CesiumJS is actually capable of with my data. Without
getting too technical, can you walk me through what this kind of data can
look like and do on the globe?



Describe it in terms of the experience — what would someone see, what
could they click on, what could change or animate — not the API names. I
want to understand the possibility space before I decide what to build.
```

## Scenario: You have too many feature ideas and need to scope down

You've come up with a long list and need help cutting it to something buildable.

None

```
I've been thinking about this a lot and I have a lot of ideas. Here's
everything I'm considering:



[list your feature ideas]



Help me figure out what's essential for a first version versus what
should wait. Be direct — if something sounds simple but isn't, tell me. I
want a v1 that an AI agent can actually build in one pass, not a roadmap.
```

## Scenario: You want a feasibility check on one specific feature

You have a specific feature in mind — something you saw elsewhere or thought up — and want to know if it's realistic.

```
None
I want to check whether one specific feature is realistic before I put it in the spec.

The feature: [describe what the user does and what happens]

Is this possible in CesiumJS? If yes, how hard is it and what does the implementation look like at a high level? If it's complex, is there a simpler version that achieves the same thing? If it's not possible, what's the closest thing CesiumJS can do?
```

## 1D. Design and configuration

## Scenario: You need to figure out how interactions should work

You know what data you want to show but haven't decided what happens when someone clicks, hovers, or moves the camera.

```
None
Help me design the interactive behaviors for my project. I want to think through:

- What happens when someone clicks an entity — built-in infobox, a custom side panel, something else?

- Should anything happen on hover? At what point does hover become a performance problem for my data size?

- When someone clicks something, should the camera fly to it?

- If my data has a time dimension, does the clock run automatically or does the user control it?

Give me recommendations, not just options. Tell me what you'd do for this project.
```

## Scenario: You need to figure out how the data should look visually

You haven't decided how your entities will look on the globe — colors, sizes, labels.

```
None
Help me think through the visual styling for my project.

- Should entity color encode something from my data, or be uniform?

- If color encodes data, what field should drive it and what kind of scale makes sense?

- What base imagery layer fits this project best – satellite, street map, dark/minimal, Google Photorealistic Tiles?

- Do I need labels on the globe, or just in the info panel on click?
```

## Scenario: You need to sort out CesiumJS-specific configuration

You haven't figured out the technical CesiumJS setup — ion token, default camera, which widgets to show.

```
None
Help me nail down the CesiumJS configuration for this project.

- I need a Cesium ion token – where do I get one and how does it live in the project securely?

- Does my project need ion-hosted terrain or imagery, or is the default sufficient?

- Which of the built-in Cesium widgets should I show – timeline, animation controls, geocoder, base layer picker? Which should I hide?
```

## 1E. Closing the planning phase

### Scenario: You think you're ready and want the Gem to confirm before writing the spec

You've had a good conversation and think you've covered everything — you want a sanity check before the spec is generated.

```
None
I think we've covered everything. Before you write the spec, summarize back to me what we've decided:

- What the project is and who it's for

- The data — which API or file, what it returns, how it flows to the globe

- The v1 features — what's in and what's deferred

- The key interactions and visual choices

- Any open questions that still need resolving


If anything is missing or still ambiguous, ask me now. If it all looks complete, let's write the spec.
```

### Scenario: You're ready to generate the spec

Everything is decided and you want the spec written.

```
None
I'm ready. Write the spec.
```

# PHASE 2: BUILD WITH AI

*You now have a spec.md. Take this into Antigravity and paste it into a spec.md file and follow the instructions below.*

## Scenario: Kicking off the Antigravity agent with your spec

You have your spec and you're opening Antigravity for the first time. Create a file called *spec.md* in Antigravity and copy in your spec.

```
None
I have a spec for a CesiumJS project I want you to build. Read it
carefully before writing any code.

Use heavily commented code throughout — the person who owns this project
is learning and should be able to read every file and understand what it
does and why.
```

## Scenario: Debugging the build

If the globe is not loading, check to make sure your .env contains **your Cesium ion Token**.

It is likely that some things won't work as expected on your first iteration—that's okay! You can either describe the issue you see to the AI agent or use **F12** to open up the console and copy any errors into the Gem, and use that response to form a prompt to update the Antigravity agent.

# PHASE 3: TEST & POLISH

*These prompts are for when you want to add features or improve what exists (**interacting with the Gem**).*

## Giving the Gem context of a project

To give the Gem context about any updates made in Antigravity, use the following prompt in Antigravity to generate a *project-description.md*. Save this Markdown file and upload it to the Gem:

```
None
Please generate a file called gem-context.md in the root of this project.

This file will be uploaded to an AI assistant (Gemini) as a replacement for direct
GitHub repo access, so it needs to give the AI enough context to understand the full
project and answer questions about it, suggest changes, and write accurate coding
agent prompts.

The file should include:

1. Project overview — what this app does, what technology stack it uses (CesiumJS,
   React, TypeScript, Vite, etc.), and what problem it's solving

2. Project structure — a file tree of the project with a one-line description of
   what each file or folder is responsible for

3. Key files — for each of the most important files (components, config, styles,
   data files), include:
   - The full file path
   - A plain-English explanation of what it does
   - The full file contents, clearly labeled

4. Data shape — describe and show the structure of any data files (e.g. JSON),
   including field names, types, and what each field represents

5. Feature inventory — a plain-English list of every feature currently in the app,
   describing what it does and roughly where in the code it lives

6. Known patterns and conventions — any consistent patterns used in the codebase
   (e.g. how state is managed, how CesiumJS entities are created, how styling is
   organized) that an AI would need to know to suggest changes that fit the existing
style

7. Current limitations or TODOs — anything in the code that is incomplete, hardcoded,
   or flagged for future improvement

Please write this in clear, readable markdown. Prioritize completeness over brevity —
this file is meant to fully replace reading the source code directly.
```

## Adding features (Gem)

### Scenario: You want to add a new feature and need to plan it before touching the code

You know what you want to add but want to think it through before opening the editor.

```
None
I want to add a new feature. Help me plan it before I touch the code.

Current project: bring in your project-description.md file you created

Feature I want to add: [describe it — what the user does, what they see, what data drives it]

Tell me: does this feature fit cleanly into the existing architecture, or does it require rethinking something? What's the simplest version of it? And write a addingfeatures.md file I can give to Antigravity to build it on top of the existing project.
```

### Scenario: You want to add a new data layer

You want to bring in an additional dataset or API alongside what's already on the globe.

```
None
I want to add a new data layer to my existing project.

New data: [describe the API or file — what it contains, where it comes from]

How it relates to the existing data: [does it overlap geographically? is it a different time period? does clicking it do something different?]

Help me figure out: whether a good API exists for this, how it fits with the existing data layer, and whether there are any conflicts or complexity I should know about before adding it. Then describe what needs to change in the project.
```

## Scenario: You want to improve the interactivity

The map works but interactions feel flat or incomplete.

```
None
The interactions in my project feel incomplete. Help me improve them.

Current behavior: [describe what clicking, hovering, and camera movement do now]

What feels missing: [describe the gap — info panels that feel sparse, no hover feedback, camera doesn't respond well, etc.]

Suggest specific improvements that are realistic to implement and explain what each one would add to the experience. Then tell me what to change in the code.
```

## Scenario: You want to improve the visual design

The project works but it doesn't look the way you want.

```
None
I want to improve the visual design of my project. It works but it doesn't look the way I imagined.

What I have now: [describe the current look — colors, base imagery, terrain, entity appearance] (or provide a screenshot)

What I want it to feel like: [describe the target — darker and more dramatic, cleaner and minimal, more data-dense, etc.]

Give me specific changes: base imagery layer, entity color scheme, terrain settings, and any Cesium scene settings that would get me closer to the look I want.
```

## Scenario: You want to improve performance

The app works in development but is slow for real users.

```
None
My project is live but real users are finding it slow. Help me optimize it.

What's slow: [describe the specific complaints or what you've observed]

My data: [format, feature count, how it loads]

Current setup: [entities vs. primitives, clustering, any optimization already in place]

Walk me through the highest-impact changes I can make — data loading strategy, rendering approach, caching, or anything else. Prioritize the ones that will make the biggest difference for the most users.
```

## Sharing and testing (once we publish to the web!)

## Scenario: You want to share the project with testers and need a guide

Your project is live and you want to send it to people for feedback.

```
None
I want to share my project with testers. Write me a short guide I can paste into a message or email.

Project URL: [your URL] (when we get to this stage)

Who the testers are: [non-technical / developers / domain experts / general public]

What I most want feedback on: [specific features, the data, the overall experience]

Keep it short and friendly — no technical jargon. Include what it is, what to try, and how to send me feedback.
```

## Scenario: You got feedback and want to use it

Testers responded and you have a mix of suggestions, bugs, and reactions to sort through.

```
None
I got feedback from testers. Help me use it.

Here's what they said: [paste or summarize the feedback]

Help me sort this into: bugs that need fixing, UX improvements worth making, feature requests to consider for a future version, and things to ignore or deprioritize. Tell me what to tackle first.
```