Polyline, Feature line, Breakline and Boundary

SiteDesign · https://civilwhiz.com/docs/polyline-feature-line-breakline-and-boundary/

Motivation

This document is written to clarify certain concepts in Civil 3D, which are quite powerful in terms of creating, editing, and shaping an arbitrary shape of the Surface. A correct understanding of `polyline`, `feature line`, `breakline` and `boundary` is essential to create a site design.

Polyline

A polyline is just a series of connecting lines with 3D coordinates including elevations; it's just a CAD object and not a Civil object. You can imagine it as a series of lines with `{x,y,z}` coordinates, floating in midair.

Feature Line

A Feature Line is an upgraded version of a polyline. It is a Civil object and no longer just a CAD object. It has all the properties of a polyline, but it also has properties that a polyline doesn't have. Chief among them is the `Style` information (which is only available to a Civil object).

Because it is a Civil object, it can interact with other Civil objects like `Surface`. For example, it supports grading (you can design slopes starting from the Feature Line), it can link to a surface dynamically (when the surface elevations change, so do the Feature Line elevations, or vice versa), and it can interact with the `Site` object.

Breakline

A Breakline is also a Civil object and is used exclusively in the `Surface` object. Like the Feature Line, it has all the properties of a polyline, along with additional properties like the `Style` object.

You use Breaklines to define the constraints around your development area—also known as a **platform**—including the `elevations`, and more importantly, how the triangles are formed around the development area. You can import a Feature Line into the Breakline Collection and create a Breakline from Feature Line. But they are distinct objects because Breakline has to live within a Surface, whereas Feature Line is an entity that can exists on its own. Plus, Feature Line can use to do grading and all but not Breakline.

A Breakline essentially only functions as a constraint that influences how surface triangulation behaves. It ensures that the triangles within the breakline will stay inside of it, and the triangles outside the breakline will stay outside of it.

By assigning `elevations`, your development area will acquire a 3D form, and Civil 3D can interpolate the points inside the Breakline (which is often a closed polyline).

The Wall Breakline

If there are drops between two platforms, then you need to explicitly define a `Wall` Breakline between them. Civil 3D can't automatically infer a drop in elevation when transitioning from one platform to another; a Wall Breakline helps it recognize and represent the vertical difference.

When a Wall Breakline is defined, Civil 3D carves out a small sliver of space—not zero, but certainly not visible unless examined under a microscope—for the vertical drop to happen.

Boundary

But enclosing your platforms with Breaklines is not sufficient. The philosophy of Civil 3D surfaces is to interpolate between all the available points within the Surface, subject to the constraints of Breaklines. It doesn't know that there are gaps or holes between your disconnected platforms. So if you have unwanted areas between your multiple disconnected platforms, you need to delete those triangles.

You could do this manually, but it's very time-consuming. Instead, you use `Boundary` objects.

A boundary doesn't have elevations to begin with-- unlike Feature Line or Breakline, it is only a 2D object that constraints your sites and turns patches of them off. Like Breakline, it must live within a `Surface`.

How to generate disconnected Platforms

Approach 1:

- Define **Wall Breakline** as necessary
- Use an **Outer Boundary** to define the total area of your platforms. The exact size doesn't matter as long as it covers all the platforms.
- Use **Inner Boundaries** to define the "holes" that should not be part of the surface. These must precisely exclude everything that is not on your platforms but is within the **Outer Boundary**.

It should be noted that the **Inner Boundaries** can be automatically generated once you subtract the all the defined Breaklines from the **Outer Boundary**. Alas this has to be done manually in Civil 3D.

Approach 2:

Alternatively, this is what you can do also

- Define **Wall Breakline** as necessary
- Get the first platform boundary and define it as **Outer Boundary**. This will turn the platform on, but will turn off the rest of the Platforms
- Then, add a **Show Boundary** for the rest of the Platforms. This will turn on the rest of the Platforms, leaving the "holes" in between the platforms on.



Ngu Soon Hui

My passion is in sharing knowledge — Civil 3D or otherwise– with the world.

I'm also the benevolent dictator for life at MES Innovation Sdn Bhd.