Layout System Block Diagram

ChunkSysBlkDiagram.png

This block diagram illustrates the main parts of the chunking system.

The lowest-level APIs are within TileFabLib and ZoneManagerLib. These libraries handle all the tile/prefab loading and unloading as the camera moves around.

The next highest-level component is a ZoneLayout. This is attached to a GameObject in your Unity scene. As your camera moves, your custom code calls a special Update method in the ZoneLayout to unload/load tiles and prefabs as they move out of and into the camera view.

Each ZoneLayout component will instantiate a ZoneManager scriptable object at runtime. The ZoneManager tracks which chunks are currently loaded.

It’s entirely possible to have more than one TPZoneLayout, as I’ll discuss a bit further on.

The Layout Name field in the TpZoneLayout component is used to link the TSceneSpec to the ZoneLayout (the dashed line in the block diagram). The TpChunkedSceneManager component uses the Layout Name to determine which TpZoneLayout to use for a particular TSceneSpec.

The next highest-level component is the TpChunkedSceneManager. It has a public field which can be used to provide a List of TpZoneLayouts, but it will automatically use those which are attached to the same GameObject.

The ChunkedSceneManager uses data in the TSceneList project asset file to load TScenes.

The chunking system can be used with Tile Scenes or `TScene`s. One or more TSceneList project assets (in the middle of the picture) are referenced by the ChunkedSceneManager (upper-right).

You create an instance of this SceneList asset in a project folder as you normally would. The inspector for the asset prompts you to click a button which opens a customized Editor window for the asset.

The SceneList asset is added to the ChunkedSceneManager as a reference (dragging it in or by code). Then your code can make calls to ChunkedSceneManager to load one of the individual Tile Scenes (TScenes) within the SceneList. One can use the index of the TScene in the SceneList, the SceneName string value (part of the TScene data but can be changed by you) or the GUID of the TScene (added automatically when they’re created and not alterable).

It would probably be helpful to open TSceneList.cs. Scroll down to about line 200 or so and you’ll see m_TileScenes. That’s the list of TScene instances. Each of these can have one or more TSceneSpec instances. TSceneSpec instances connects the Selector (selects what TileFab to load)

to the appropriate Layout.

In your scene and on some GameObject, add the TpChunkedSceneManager component and as many TpZoneLayout components as there are groups of Tilemaps. A Tilemap group is a single Grid with one or more child Tilemaps.

Revision #3
Created 10 July 2025 18:51:59 by Vonchor
Updated 11 July 2025 11:37:35 by Vonchor