TpZoneLayout

You may have noticed in TSceneList that each TScene can have multiple TSceneSpecs. What's that for?

Each TSceneSpec helps ChunkedSceneManager understand which TileFab is used to add tiles to a particular Tilemap group.

A Tilemap Group comprises a Grid with one or more child Tilemaps. Each Tilemap group uses one Chunk Size. For example, a Chunk Size of 32 means that 32x32 areas of Tiles are swapped in/out for all the Tilemaps which are in the group.

One limitation is that a Tilemap Group must be exclusive to a single ZoneLayout. Using a Tilemap Group with multiple ZoneLayouts will produce undefined results and possible exceptions.

Different Tilemap Groups can have different Chunk Sizes. For example, you can have two Grids with their child TileMaps. Each Grid/Map group is 'handled' by a TpZoneLayout. The SceneSpec tells ChunkedSceneManager which TpZoneLayout to use for a Tilemap Group.

In the demo program, the first TScene has two TSceneSpecs, implying two Groups. The second one paints a TileFab as an outer border and uses a single-TileFab Selector. Very simple, it just returns the same TileFab each time.

Since TpSceneSpecs are in a project asset and a TpZoneLayout is a component in a Unity scene, a direct connection by reference isn't possible. That's the reason for the Layout Name field of the TpZoneLayout component. At runtime, the name embedded in the TSceneSpec is used by TpChunkedSceneManager to locate a particular TpZoneLayout.

It's all handled automatically, behind the scenes. All your code needs to do is to call the ZoneLayout's UpdateTick method as your character moves, and handle several callbacks:

From ChunkedSceneManager

```
public event Action<TpChunkedSceneManager, TSceneList.TScene?>? OnBeforeTSceneChange;
```

Invoked just before current TScene is unloaded.

```
public event Action<TpChunkedSceneManager, TSceneList.TScene>? OnNewTsceneChosen;
```

Invoked just before the new TScene is loaded.

```
public event Action<TpChunkedSceneManager, TSceneList.TScene>? OnAfterTSceneChange;
```

Invoked just after the new TScene is loaded.

```
public event Action<TpChunkedSceneManager, TSceneList.TScene?, TSceneLayout, ZoneReg, TpZoneManager>? OnZoneRegAddedForLayout;
```

Invoked when a Zone Registration was added. You don’t normally need to handle this.

```
public event Action<TpChunkedSceneManager, TSceneList.TScene?, TSceneLayout, ZoneReg, TpZoneManager>? OnZoneRegDeletedForLayout;
```

Invoked when a Zone Registration was deleted. You don’t normally need to handle this.

```
public event Action<TpChunkedSceneManager, Tilemap, List<TilePlusBase>>? OnTptTilesWillBeDeletedForLayout;
```

Invoked when TilePlus tiles will be deleted. This is handy for extracting any save-file information before the tile is deleted. The demo program shows how this works for the particular save file scheme used in the demo.

Revision #2
Created 11 July 2025 11:42:14 by Vonchor
Updated 11 July 2025 11:46:28 by Vonchor