Key Elements

New Tile Class

The key component of TPT is a new Tile base class cleverly dubbed “TilePlusBase” (**TPB**). This tile clones itself when placed on a Tilemap in a scene.

Why would anyone care?

The One With No Instance Data

One of the issues that developers run into with Unity Tilemaps is that there’s no way to add fields (variables) to tiles and have the data serialized and saved in the scene just like the serialized fields of scripts used for components. This is because the Tilemap’s serialization is hard-wired to save data from the basic fields present in a Tile class.

For many (including your TPT developer) this is annoying, to say the least. Perhaps you want a configurable waypoint. Maybe you want to be able to paint a tile and set it up as a spawn zone. And you want to be able to edit fields as you usually do.

Using TilePlusBase and the supporting libraries you can have any sort of code and/or data in a tile. Since the tile is cloned, it is no longer connected to the asset in the project folder: it exists in the Scene, and its data is saved with the scene.

If you’ve used the Unity Tilemap Editor (UTE) you’ve used its Selection Inspector. That inspector is very different from the normal Unity inspector panel: the UTE Selection Inspector is hard-wired to support the fields of the Tile-class tile and that’s it.

The support libraries for TPT have an alternative Selection Inspector that’s available in the UTE as the “Tile+Brush” or by using the TPT’s Tilemap painting and editing tool: Tile+Painter (T+P).

Decorate your TPB-derived tiles with TPT’s custom attributes and this alternative Selection Inspector lets you view and edit those fields or display property values. See Attributes.

Services

Services make it easier to use single-instance Scriptable Object assets as runtime-loadable code blocks.

- Replacing static classes with loadable classes can improve reload time in the editor.
- Unload code when you don't need it anymore to free memory.
- TilePlus includes several bundled services: Spawner, Tweener, and others.
- It's easy to add your own!

See the Services section for more information.

TileBundles and TileFabs

Another important feature of this system is the ability to archive the contents of one or more Tilemaps into archives which can be loaded by code.

With most Unity GameObjects + components you make prefabs.

They're not terribly useful with Tilemaps. If you instantiate such prefabs you'll end up with multiple Tilemaps and Grids. What would be more useful is the ability to quickly add and delete areas of tiles on demand.

That's what you can do with Bundles and TileFabs. These archives are also the *only* way to archive TilePlus tiles. Since these are scene objects and don't correspond to assets in the project, references to these are lost and one ends up with the familiar pink/whatever colored tiles.

The TilePlus system has a custom archiver that can:

- Bundle all tiles from a Tilemap.
- Bundle all the tiles from an area (Grid Selection) of a specified Tilemap.
- Optionally archive references to all Prefabs which are parented to the Tilemap's GameObject.

When presented with several Tilemaps, the archiver also creates a TileFab asset. This asset contains all the Bundle references for all the Tilemaps.

The `TileFabLib` static-class library has methods which allow loading of individual bundles or an entire set from a TileFab. There are both synchronous and asynchronous methods, and the async methods allow distributing individual bundles loads over several frames.

It's important to note that Bundles and TileFabs created from a Grid Selection (that is, just some portion of the Tilemap) are position-independent. This means that you can load a TileFab and its bundles anywhere on the Tilemap(s), not just their original locations.

Built on this framework are `ZoneManager` and `ZoneLayout` which are part of an automatic layout system for top-down views. This system adds and deletes TileFabs and their bundles as the Camera moves.

TileFabs and Bundles are also used extensively with Tile+Painter.

- They can be paintable objects
  - Use them like paintable tile-prefabs.
- Select scene view areas to bundle.
- Select Palette areas to bundle
- And Much More (tm).

An even higher layer of software enables "tile scene" management.

See this for more information.

Revision #14
Created 2025-06-22 15:50:31 UTC by Vonchor
Updated 2025-09-15 18:53:18 UTC by Vonchor