

Limitations

Code Changes

Changing code in subclasses may result in missing references in already-placed clone tiles. This is like what happens in the GameObject hierarchy when components' internal structure is altered. Adding serialized fields or altering the names of serialized fields will result in uninitialized fields in the placed clone tile.

Note that the `[FormerlySerializedAs]` attribute can be used as a workaround if you use it while editing your code.

Again, this is not an issue exclusive to TPT. Obviously, one can edit the placed clone tile to update the references if needed, but this can be time consuming.

Move Function and Overwrites

When using Painter or the Unity Palette's "Move" function, it's not possible to prevent overwrites. At least, I haven't figured out how to do that yet for every possible edge case.

Mostly Tested in Top-Down Applications

This extension has mostly been tested with square tiles using a top-down XY view.

Picking/Copying TPT tiles

When you Pick one or more TPT tiles with a brush or with Tile+Painter and subsequently paint those tiles, what you're really doing is painting a copy of a cloned tile. TpLib catches this and re-clones the tile. Two identical tile instances would re-create the problem of asset modification since both tiles share the same instance data.

If you want to copy/paste a TPT tile in a running app, please use `TpLib.CopyAndPasteTile`, and to move a TPT tile use `TpLb.CutAndPasteTile`.

Revision #4

Created 2025-07-08 15:36:19 UTC by Vonchor

Updated 2025-08-13 17:46:13 UTC by Vonchor