What's New In Version 6

Changes in Version 6 of TilePlus Toolkit

- Version 6 Changes

Version 6 Changes

If you're upgrading to version 6, have tons of errors, and are looking here: DELETE the Assets/Plugins/TilePlus folder!

[NOTES]

- If upgrading from an earlier version, delete the Plugins/TilePlus folder, upgrade, then after the post-upgrade compilation has completed, please close and re-open the Unity Editor.
  - Since the asset is no longer in the Plugins folder this is very important!!
- Please note that documentation can be found in the TilePlusExtras folder.
- The TpLib version number can be found by using the System Information window, available from the Tools menu.
- Upgrading? Depending on the version, it may be internally very different. It is advised to use a copy of your existing project when upgrading.

IMPORTANT NOTES:

- **No longer resides in Plugins folder.**
- **Requires Unity 6.5 or newer.**
  - Tested on 6.6
- There are API breaks. Please ensure proper backups!

Known Issues

- Domain Reloading: the core library (TPLib) works properly when Domain reloading is disabled in the editor.
  - However, not all Monobehaviors provided with this package will work properly with Domain Reload disabled.
- TilePlusPainter:
  - The horizontal splitter lines can appear in the wrong place; this can depend on your Unity version.
  - When using the UTE to view Palettes: Sometimes when clicking on a Palette in the center column for the first time the right column palette display is blank. If this occurs click a different palette and then return to the one you want to use.
    - This is very intermittent and harmless. Ignore the mathy exceptions.

- Certain Unity 6 versions of the Unity Tile Editor (Palette) have this issue: the marquee (outline) when dragging an area of the palette won't appear the first time that you use the palette.
  - This is a reported bug in the palette window or in Painter's palette window, which are the same thing. Fixed in newest 6.x versions.
- If Painter is docked then changing play/edit mode in the Unity Editor will occasionally produce this error:
  - Screen position out of view frustum (screen pos 1146.000000, 397.000000, 0.000000) (Camera rect ....)
  - AFAIKT, this is some sort of error in the internal Tile Palette UI Element embedded in Painter.
- The ShowSGrid option doesn't work properly except in Quadrant I. (X >= 0, Y >= 0). This will be fixed in the future.
- Using MOVE: pick then change tilemap doesn't work properly. BUG. Workaround is to save clipboard to favorites, change tilemap, paint, and delete the original.
- When using Painter to place TileFabs or Bundles and prefabs are part of a Bundle then Undo isn't possible for the prefabs. This will be fixed in the future.

[6.0.0]

ADDED

- New demo in 'Oddities' folder: *Match*
  - Implements the basic Match-3 mechanic using tiles as the tokens.
  - Token movement done by Tweening the tile sprites then moving the tiles.
  - Not a complete game, just the mechanic.
- The TilePlus Configuration editor's RELOAD button now shows as RELOAD + GC and does a full Garbage Collector pass after the reload.
- System Info shows all Tilemaps with TPT tiles and also a list of all Tilemaps in the project.
- New Tools/TilePlus Menu command to update all tiles on all maps in all scenes.
- TpLib:
  - New Method: RefreshAllTilemaps. Refreshes all Tilemaps in all scenes.
  - New Method: GetAllTilesOutsideRegionForMap(). Uses an input RectInt to return all TPT tiles outside of a region for a particular Tilemap.
  - new event: OnSafeDeleteUsed
    - Fires when SafeDeleteTile is used to delete a tile. Usually this is from within a Tile's code.
  - signature change: SafeDeleteTile.
    - Includes new bool invokeSafeDeleteUsed allows controlling if the OnSafeDeleteUsed callback is invoked or not. Defaults to true. See inline code doc for explanation.
  - TpLibScene: GetAllScenes has new param: ignoreUnloadedScenes. Mostly useful in-editor when you may have unloaded scene in the Hierarchy.

    - Tasks: see 'Removed' - tiles can't ask for early update via an interface spec. Instead:
        - TpLib.RegisterForUpdate and TpLib.UnRegisterForUpdate are used to add/remove UnityEngine.Obect hosts and Action<float> targets.
        - These Objects have their Actions invoked every Update.
        - Registrations can be manually removed, but:
            - If a host and target are supplied and either is null OR
            - a host isn't supplied BUT the target becomes null THEN
            - this registration is automatically removed.
        - This feature can be used by tiles OR TpLib.InvokeRepeatingUntil can be used instead (late update).
        - Note that SRS have their own way of requesting LateUpdates. If a normal Update is desired: use this new feature.
        - Note that this uses the PlayerLoop system: see https://docs.unity3d.com/Manual/player-loop-customizing.html
        - Specifically, it's added to this group:
            - Update
            - ScriptRunBehaviourUpdate
            - ScriptRunDelayedDynamicFrameRate
            - ScriptRunDelayedTasks
            - DirectorUpdate
        - It's added to the END of this list and will run last (unless something else changes player loop)
    - New Awaitable helpers: Awaitable WaitWhile(Func test), Awaitable WaitUntil(Func test)
    - new methods:
        - ActiveTpLibInit. Get the active TpLibInit instance. See 'Changed' below.
        - HasTile<T> is a generic version of TpLib.HasTile.
        - RescanTilemap: Rescan a tilemap for TPT tiles after a delay. (Editor only)
    - DelayedCallback:
        - Added a UserData field for any object. This data is passed to the Condition and to the Callback.
            - For the Condition Func, this object is passed to it in addition to deltatime.
            - This feature also applies to InvokeRepeatingUntil. Examine the arg list for invokedFunc. That Func gets deltatime and the provided object.
            - To be clear, using UserData is optional. Comes in very handy at times.
    - InvokeRepeatingUntil has a final callback added. This is invoked when the process ends, and any UserData supplied to InvokeRepeatingUntil() is sent to the callback.
    - A new Debug property 'NoDebug' is TRUE when none of the other Debug properties are true.
        - NoDebug => `!(Errors || Firehose || Informational || Warnings);`
- Tile+Painter:
    - When viewing a TPT tile in Painter in Play mode, the runtime inspector now shows the ZoneBoundsInt. When not in Play mode, a help box shows this info if the LockAll toggle for the ZoneBoundsInt is checked.
    - New Painter shortcut for moving the cursor defaults to F12.

        - Opens a dialog, you enter a position, scene view zooms to that position.
- TpTileBundle:
    - New internal field in the archive (m_SpriteIntegerSize) that gets populated with the size of the sprite from the source tilemap.
    - New property: TypeCache. First use generates the Cache. Returns a Dictionary of Types (keys) with each Type's Instances.
    - Note: as usual, DON'T cache any tpt tiles returned from this property.
        - The Layout system's ChunkZoneSelector implements two methods that are useful to query this cache.
            - List<TemplateSelectorQueryResults<T> GetTilePlusTilesOfType<T>()
            - List<TemplateSelectorQueryResults<TilePlusBase> GetBundleTilesWithInterface()
            - These are useful to extract specific TPT tiles when using the Layout system to load chunks of tiles.
                - For example, if you have a TPT tile Waypoint class with a certain Waypoint as the 'start' waypoint then you can search for it, knowing its position when it would be loaded lets you know to position the camera over that tilemap position and run a Layout pass: that will load the Tile chunks centered around that position, including the waypoint you searched for.
    - Additions for support of the new TpFabUpdater component:
        - A new field: OriginalTilemapBoundsInt. This is similar to TilemapBoundsInt but it also has the original position of the Bounds. This is initialized with an 'Impossible' grid position so older assets can be identified.
        - A new property: AssetIsVersionSixOrNewer. Detects an uninitialized OriginalTilemapBoundsInt. If this fails then the internal value (m_SpriteIntegerSize) of the actual sprite size in the source map is not available either.
        - These changes only affect the use of TpFabUpdater, which requires the original bounds of what's to be archived in order to update a Bundle asset properly.
- TpTileFab:
    - UpdateCompatible can be used to test all child bundles of the TileFab for AssetIsVersionSixOrNewer.
    - MapNames: convenience method to get all of the Tilemap names: a list of Tilemap names for the child bundles.
- TpNoPaint Monobehaviour component has two new new fields:
    - Ignore: If checked, if a Tilemap: ignore this Tilemap. If a Grid, ignore all the Tilemaps which are children of the Grid.
    - ReactivateForPlay: if checked, ignore is forced to false during play mode.
        - This is handy if you want some Tilemaps not to appear in Painter unless in Play mode.
- TpFabUpdater: new EXPERIMENTAL Component.
    - Add to a Tilemap's parent GRID.
    - Add asset references for a TileFab or a ChunkZoneSelector.
        - If both are provided, the TileFab asset referenced in the Selector is used.

        - When using the Layout system use the Selector because that ensures that the proper bundles are updated.
    - Used to update a referenced TileFab's BUNDLES from the child Tilemaps of the GRID.
    - See the component's code for more info. VERY HANDY for iterative level creation using TileFabs.
    - IMPORTANT NOTE: bundles created with from a multiple-selection PICK are incompatable with FabUpdater.
- TpActionToTile and TpInputActionToTile:
    - An 'Initialized' property was added.
    - Swiping support. If enabled, a swipe between two *adjacent* TPT tiles on the *same* tilemap will swap the two tiles
        - This includes clicking on two adjacent tiles : same as a swipe.
    - Tweens the two tile sprites using TpTileUtils.SwapTiles
    - Includes a new callback: OnSwipe. Note that although the two tiles are swapped, they are not messaged. Hence the callback is required if something else has to happen.
        - The callback includes an Awaitable that can be used to determine when the swapping is complete.
    - The ClickOnMouseDown property was removed from the ScriptableObject and the corrsponding field in TpInputActionToTile were removed.
    - SupportSwipe and PerformSwipe properties were added to the ScriptableObject.
    - SwipeEase and SwipeTweenDuration properties:
        - SwipeEase: Ease function for tweening for swipes.
        - SwipeTweenDuration: time for the swipe to complete.
        - These were also added to the TpInputActionToTile component.
    - The OnSwipe callback was added: This is invoked if SupportSwipe is true and two adjacent TPT tiles are swiped.
        - The callback provides the two tiles being swiped and an Awaitable that can be used to determine when the swipe is complete.
            - If PerformSwap is false then the Awaitable will be null.
        - OnSwipe has a default handler in TpInputActionToTile: if there's a sound clip and audio source in the inspector then the sound will play for a swipe.
- TpChunkSelectorBase has a new method: GetBundleTilesOfType<T> which is implemented in TpChunkZoneSelector.
    - Unlike GetTilePlusTilesOfType, this uses a per-bundle TypeCache for greater speed. HOWEVER the cache is created when first accessed, so the first use of this method is slower.
- ITpUiControl:
    - New property: SwipeMode. Returns a value from UiSwipeMode enum.
        - Controls whether this tile accepts Swipes and how they are performed.
        - See UiAnimButtonTile for an example.
    - UiEffectEnum adds a new value: Custom
    - New Property: ToolTip. This is not supported by TPT but can be useful for UI.
    - New Property: CanAcceptHover. Indicates that a tile will EVER accept hover messages.

        - This works in conjunction with AcceptsHover, which indicates that the tile accepts hover messages at a particular point in time.
        - Useful when a SceneInitializer runs for a new TileScene: you can build a list or hash of all UiTiles that might ever accept hover messages even if they're not loading into the scene at that time.
    - The following method needs to be implemented in each tile using this interface.
        - void ITpMessaging<ActionToTilePacket>.MessageTarget(ActionToTilePacket sentPacket){}
- TilePlusBase class
    - A new static property: SetTileUserData.
        - This is a write-once (i.e., when your app starts) that can be used to add an arbitrary data object that's accessible to all TilePlusBase-derived tiles.
        - Example: you have a global state object that isn't static or singleton. Write that to SetTileUserData and all tiles have access to it.
        - There's a corresponding static property TileUserData which provides access to this object.
    - TilePlusBaseEnums has an added PositionDbAddMode
    - Controls how TPT tile additions to the PositionDb service are handled: cached, added immediately, or ignored.
- PositionDb
    - TileBase tiles (AutoTile, RuleTile, AnimatedTile (Unity's) etc) are properly added to the PositionDb.
        - Note: these additions are not internally cached at this time, and these tiles sprite transforms are assumed to be unchanged.
    - For TPT tiles, the tile's PositionDbAddMode property is examined.
        - Controls if a tile is added immediately or cached on a tile-by-tile basis.
        - Caching MUST be used for TPT tiles where the sprite position or size will change; e.g., via tweening, etc.
- TpTileUtils:
    - SwapTiles is now async and some parameters were changed.
    - A new method async Awaitable <bool> MoveTile() can be used to move a Tpt tile with tweening.
        - This tweens the sprite position then resets it and moves the tile.
- Tweener service:
    - Tweens can be paused with PauseTween. Pause state can be queried with TweenIsPaused.
        - The TweenMonitor has been updated to show paused state for Tweens and sequences.
        - TpTweenerFlex and TpTweenerSeq tiles have a new button that can be used to pause a tile or sequence.
        - These tiles also have a new info field in the inspector which shows running or paused states.
    - RUNNING sequences can be paused with PauseSequence. Pause state can be queried with SequenceIsPaused.

    - PENDING sequence instances (live but not running yet ie PlaySequence not used) can be obtained with GetPendingSequence.
    - Sequences have an added 'UserData' property for adding an arbitrary object.
    - PlaySequence can have a UnityEngine.Object as the parent, not just a TPT tile.
    - The IsRunningSequence property can be used to determine if a sequence is running.
    - A new callback is available for tweens: Finalize. This is an Action internally-called by the Tweener just before OnFinished is invoked.
        - Can be used by a custom tween for cleanup such as restoring the original position of a GO, see ShakeGameObject
    - Added new special GameObject tweens:
        - ShakeGameObject
        - SpriteRendererAlphaTween
        - SpriteRendererShaderTween
        - Awaitable MoveGameObject
        - Generic tweens for any Vector3 or float value: provide a Func for 'Get' and an Action for 'Set'.
- TpChunkedSceneManager: adds functionality for changing Scene List assets.
    - new event: OnSceneListAssetChange
    - new Methods:
        - ChangeSceneList
        - SameSceneList
        - SameTscene (two overloads)
- TpMessages is a static version of the TpMessaging Service (see 'deprecated' section below).
    - Note that all methods which return an array of TptTile instance IDs now return EntityIds.
    - The pools for ObjectPacket, ActionToTilePacket, PositionPacket, and StringPacket have public accessor properties.
    - TpMessages has all of the functionality of the Messaging Service and adds a typed subscriber messaging system:
        - Use this new functionality of TpMessages to allow a Monobehaviour, Scriptable Object, or static class to specify specific types of data packets it want to be sent, similar to how TpMessages works for TPT tiles.
        - New Properties:
            - LogContextStack: if true, logs message context stack pushes and pops.
            - DefaultUserData: Write-once (when your app starts) any object. This is added to all message packets' DefaultUserData property automatically: inject an instance of some class via the messaging system.
        - new methods:
            - RegisterMessageTarget<T>(Object? parent, Action<object> target) where T : MessagePacket<T&gt
            - UnregisterMessageTarget(id), UnregisterMessageTarget(Object).
            - To send: SendToTargets<T>(MessagePacket<T> packet) where T : MessagePacket<T>
- Abstract class MessagePacket:

    - Added default constructor
        - Note that the default Ctor sets the SourceInstance param to null.
    - New 'UserData' Property: It's a simple way to pass custom data as part of an existing packet without having to create a new packet class.
        - Example:

```
public class Xyzzy
{
    public string A;
    public Xyzzy (string s)
    {
       A = s;
    }
}

var posPacket = new PositionPacket
{
    m_Position = new Vector3Int(1,2,3);
    UserData = new Xyzzy("ABCDEFG");
};

Messages.SendMessage<PositionPacket,SomeInterface>(posPacket);
```

        - Tiles which implement SomeInterface will get this message.
        - Using UserData allows you to have TPT tile message target handle variant messages using the same interface.
    - The ObjectPacket derived class has three new properties:
        - UObjects for an array of GameObjects or other UnityEngine.Object items
        - Vector3IntValue: an optional Vector3Int
        - HasVector3IntValue: TRUE if a Vector3Int was SET into the above property.
- TpZoneLayout new get-only properties:
    - LastComputedCamViewBounds: Most recent camera view bounds after UpdateLayout is used.
    - LastNumberOfZonesRemoved: Get the number of Zones removed for the last layout pass.
    - LastNumberOfZonesAdded: Get the number of Zones added for the last layout pass.
- Miscellaneous:
    - The TpNote component's text field is now a Text Area.
    - The formatter (ImGuiTileEditor) shows the ZoneBoundsInt in the Selection Inspector (Painter, Tile+Brush).
    - Clear Tilemaps menu function updated to recognize new TpNoPaint component features. Specifically, the validator for the command looks for TpNoPaint and won't clear Tilemaps with that component.

    - Services inspector has a new slider to set the update interval for displaying selected services.
        - Both this new value and the 'Limit' value are saved.
        - The AutoUpdate field purposely always becomes unchecked after a scripting reload.
    - TpTweenSequenceTile has a new Pause/Unpause function button (in inspector) pause/unpause a sequence.

FIXED

- TilePlusBase's Zone editor fixed some errors which can sometime force tile sprites to vanish due to z==0.
- TpImageLib ran into a new Unity bug (reported) relayed to texture importers: change anything and the Tilemap sprites become null.
    - This is fixed in several places by refreshing all of the Tilemaps in all Scenes.
- TpTilePositionDb.AddMapsToMonitor would return false and terminate the loop if one of the maps was already registered for monitoring.
    - Since multiple entities can call this method with dupe Tilemaps this bug induced errors.
    - Method only fails if a passed-in map reference is null.
- Tweener:
    - the default duration for G.O. tweens was .1 sec and should have been 0.001 sec. Caused all tweens with < .1 sec duration to be .1 sec duration. Fixed.
    - TweenAsAwaitable didn't work properly for looping tweens or sequences. Fixed.
    - A Sequence's OnFinished callback didn't properly indicate if the tween was actually complete or not. Fixed.
    - A Tween's OnFinished now properly checks for a non-null GameObject prior to invoking said callback. Fixed.
- TpLib:
    - SceneScan now properly finds all of the tilemaps to add to the internal list of ALL maps (not just those with TPT tiles).
    - Tasks: Fixes rare edge case of tiles requesting refresh but the Tile or parent Tilemap is deleted prior to the refresh being executed from the internal queue.
- TpSlideShow's ResetState method now properly sets the currentClipIndex to -1. This obscure bug prevented animation (SlideShow is actually an animated tile in 'pause') from starting properly.
- The TSceneListEditorWindow wouldn't open properly on a newly-created TSceneList asset: bug in asset script caused null-ref error in the Editor window. Fixed.
- TpPainterSceneView:
    - had a nullref if you use Painter's Grid Sel panel to create a grid selection in the Scene View and then you disable a Tilemap or Grid from that selection. Fixed.
    - wouldn't allow right-click to drag scene in GridSelection mode. Fixed.
    - edge case nullref(s) within the GridSelect panel when a Grid or GridLayout is Destroyed e.g. when a scene is deleted or overwritten in the hierarchy. Fixed.

    - edge case caused intermittent loss of a grid selection set via the GridSelection mode of painter. Fixed.
- TpPainterDataGui: edit mode 'pick' of a non-TilePlus tile doesn't show inspector. Fixed. Also, the multiple selection mode didn't properly work with multiple TPT tiles in certain situations.
- TSceneList S.O. script: the overrides for Fab and Bundle delays were very wrong: ints instead of floats. Fixed, and adjusted slider range to match those in the TpZoneLayout component.
- TpFabLoader component had a require component attribute for Tilemap: changed to Grid.
- ObjectPacket class's Reset method didn't properly reset all of the data.
- TpSpawnlink:
    - checks for the GO to be active before despawning. Fixed edge-case error when Tweener's OnFinished callback being called more than once for the same end-of-tween.
    - m_UseMessaging field added. See comment in code for explanation.

CHANGED

- TilePlusBase instance field m_TileGridPosition access changed to 'protected'
- The Bundle menu command has been relocated to the top of the Tools menu and renamed "Archive Tilemaps".
- InstanceIds have been replaced everywhere with EntityIds. This affects many APIs, not all changes are docmented herein.
    - Your IDE will show what you need to change. Most of the affected code is internal to TilePlus.
- The ability to create a prefab from a Grid and its child Tilemaps was removed.
- TpLib initialization has changed. Other code often no longer needs to test if TpLib is initialized in the Start of a Monobehaviour.
    - [RuntimeInitializeOnLoadMethod(RuntimeInitializeLoadType.BeforeSplashScreen)] is used, so TpLib should be all set up prior to any Monobehaviour Awake or Start methods execute.
    - However, if a scene is being loaded the cloning queue may be in use and the internal data structures of TpLib may not be up to date so it's never a bad idea to have this check.
- TpLib.FindTilemap(two overloads) changed to scan for maps each time rather than using a cached list. Slower but more accurate.
- TpLib.InvokeRepeatingUntil can now accept null for the parent parameter.
- Tweener API changes:
    - Individual tweens can be paused. Sequences can be paused: the current running tween from the sequence is what's paused.
    - GameObjectTrs no longer accepts a custom GameObject action.
    - Custom GO tweens use a base class 'TpGoTweenDataBase' for data. These are optionally poolable.
    - Creation of custom GO tweens has changed: see the online documentation.

    - EaseVector3, EaseVector2, and EaseFloat methods are now public static methods.
    - The GameObjTweenEaseTarget enum has a new value: PositionLocal.
        - Allows choice of affecting transform.position OR transform.localposition.
- The TpLibInit Scriptable Object has an added fields to support your having an overriding asset in your project without affecting the instance of this asset that comes with TPT.
    - m_TplName: printed to console when Informational messages are on and when this asset is recognized.
    - m_ChangePlayerFrameRate and m_TargetFrameRate
        - If ChangePlayerFrameRate is true then Application.TargetFrameRate is set to m_TargetFrameRate;
    - m_Priority: the default asset instance has this set to zero. If you add another asset of this type to a Resources/Tp folder in your project AND set this value > 0 then it'll have a higher priority than the default.
    - Note: the new TpLib.TpLibInit property will have the TpLibInit that's in use in PLAY mode (only).
    - Note: this script now has a CreateAssetMenu entry so that you can easily add one or more TpLibInit assets.
    - OverrideDebugMode and RuntimeDebugMode can be used to set exactly which error messages, if any, are emitted in a built app and/or in the editor when in PLAY mode.
        - The TpLibInit asset has a help box that explains how it works.
        - Much more flexible.
    - DON'T FORGET that TpLibInit assets must be in a Resources/TP folder to be recognized.
        - Such a folder can be a subfolder of any other folder ie it doesn't have to be 'Assets/Resources/Tp'.
- TpNoPaint: two new fields and can work on a Grid -or- a Tilemap (Previously: Tilemap only).

REMOVED

- The "Allow Prefab Editing" configuration field was removed.
- Unlocked Tiles Test menu item removed as its no longer needed.
- TpSpawnLink: m_IgnoreCollider field removed.
- TpSpawner: Event subscriptions Action<GameObject, GameObject>? OnCollidableObjectSpawned and Action<GameObject>? OnCollidableObjectDespawned;
- The Archiver no longer has any options to create Tilemap Prefabs. They're pretty useless.
    - The code still exists but is commented out. That code will be deleted in the next release.
- TpLibTiming: The Awaitable-based update loop was deleted. Affected:
    - TpLibInit asset: UsePlayerLoop field removed.
    - SystemInfo panel no longer shows whether or not the PlayerLoop is being used since it's always being used.
    - Diagnostic properties for the Awaitable loop were removed.

- The ability of TPT tiles to get an Update via the use of the WantsEarlyUpdate interface has been removed.
  - Workaround: use TpLib.InvokeRepeatingUntil.
  - Note that the nomenclature about this has been changed to talk about Update since this was never actually 'Early' update as it's added to the player loop 'Update'

DEPRECATED

- TpMessaging has reverted to a static class. For backwards compat, this service is still available but it redirects to a new static class: TpMessages.