

Zone-Based

AnimZoneLoader

This class is a subclass of TpFlexAnimated tile and inherits its fields and editor appearance.

It's used to load archived Tilemaps from TpTileFab assets, which are created by the Tools/TilePlus/Prefabs/Bundle Tilemaps command.

IMPORTANT

It's somewhat obsoleted because of the Layout system and may NOT compatible with it.

Public fields:

- Loading Offset: the location where the TileFab will be placed.
- Preview: preview the TileFab at the Loading Offset.
- TileFab: A TileFab archive from a Project folder.
- UseZoneManager: Optionally use a Zone Manager for detection of already loaded TileFabs.
- ShowOffsetPositionGizmo: show a marquee at the

Zone Managers and their uses are discussed [here](#).

More about Preview

Preview loads the tiles into a preview tilemap or maps, if there are multiple TpTileBundle assets referenced by the TpTileFab asset. The preview is active until you click Preview again or the Editor's Selection changes. When preview is active, you can change the loading offset and the preview area will change position.

Loading or previewing tiles depends on there being compatible Tilemaps that are named or tagged in such a way that the system can identity which Tilemap to use. This information is embedded in the TileFab asset when you create it. If you need to change it just edit the asset's TileAssets section in the Project window.

The tile does not automatically load Tilemaps at runtime. Rather, you send a message to it via the TpLib SendMessage methods. As configured, it expects the message to contain a Vector3Int

describing a position. If the position is within the Zone bounds, the tile uses TpLib's PostTileEvent method to post a trigger event.

AnimZoneSpawner

This can be used to spawn prefabs and TPT tiles, using assets with lists of prefabs or tiles. This tile uses the Spawner service.

Prefabs can be unparented or parented to a Scene Object by using the GameObject name or tag.

Tiles can be painted on the same tilemap or on a different tilemap which you specify using a GameObject name or tag, or a reference. This tile can respond to a SendMessage containing a Vector3Int describing a position. If the position is within the Zone bounds, it will spawn prefabs or paint tiles as you've configured it. Alternatively, you can spawn/paint via code using the instance methods SpawnPrefab or PaintTile if this approach doesn't suit you.

You can use either a TpTileList or a TpPrefabList asset (or both) to specify which tiles or prefabs (or both) to use with this tile.

Public fields include those from TpAnimZoneBase and its superclasses, and these additional fields. Public fields:

- PrefabList: a reference to a TpPrefabList asset. Note: preload amounts are set in the asset.
- TileList: a reference to a TpTileList asset provides a list of TPT assets along with the spawn position for each. Spawn position is one of eight positions immediately surrounding the tile position on the Tilemap, a random position in the Zone, or the painting tile's position if the painting is to a different Tilemap.
- SpawnMode: spawn prefabs or paint tiles in the order that they appear in the asset or randomly from those assets.
- Use Trigger: If checked, a the tile posts an event whenever spawning occurs.
- PositioningMode: spawn or paint using the setting in the assets, or force to the Spawner's position, the contact position in the Zone, or somewhere Random in the zone.
- ParentingMode: Tiles painting only, use the same tilemap to paint tiles (can't paint at the Tile's position), or use a provided reference (Painting Tilemap), or use a Tag to locate a Tilemap's GameObject using PaintingTilemapNameOrTag, or use a string to Find a Tilemap's GameObject using PaintingTilemapNameOrTag.
- PaintingTilemap: Tile painting only: Optional reference to alternate Tilemap used to paint tiles.
- PaintingTilemapNameOrTag: Tile painting only, string with name or tag of alternate tilemap.

Tile Painting

When doing Tile painting you have several configuration options using PositioningMode, PaintingTilemap, PaintingTilemapNameOrTag, and Parenting mode. When painting tiles it's important that the tile which is doing the painting is not painted over. Furthermore, since the tile occupies a position on a tilemap, you probably don't want it to appear as an obstacle to a Player or NPC walking through the tile's trigger zone (TpZoneBase or TpAnimZoneBase).

All these controls exist so that you can paint the this tile on a different Tilemap from the one that you use for pathfinding and character movement.

Paint one of these tiles on Tilemap A and use it to paint tiles on Tilemap B. When you do that, the tile needs to know which Tilemap to use. That's done by setting ParentingMode appropriately and filling-in correct values for PaintingTilemap or PaintingTilemapNameOrTag.

Prefab Pooling in TpAnimZoneSpawner

AnimZoneSpawner uses the Spawner service which provides GameObject pooling.

When a TpAnimZoneSpawner tile instance's StartUp is invoked and if SpawnMode is set to RandomPrefabs or PrefabsInOrder and if the TpPrefabList has any prefabs where the PoolInitialSize is nonzero, the tile will ask SpawningUtil to preload prefab instances. This only occurs on the first StartUp and will not occur if you change the TpPrefabList asset during runtime.

Note that preloading takes time, as each prefab must be instantiated, so choose the preload sizes carefully. If there's no preload, the pool expands automatically as prefabs are instantiated and spawned.

Normally the pooler does not attach the pooled and/or preloaded prefabs to a parent GameObject. If this bothers you, head over to the Project folder Plugins/TilePlus/Runtime/Assets and drag the Tpp_PoolHost prefab into your scene. This prefab has an attached component which sets DontDestroyOnLoad so that the prefab persists between scene loads. The pooler looks for a GameObject with this specific name.

The pooler will automatically add a component of type TpSpawnLink to spawned prefabs if it doesn't already exist. TpSpawnLink can also despawn the prefab after a timeout.

New In Version 5: if the spawned GameObject has a Collider2D or Collider component AND the TpSpawnlink instance on that GameObject has its m_IgnoreCollider field = false THEN a reference to the spawned GameObject is added to an internal HashSet of "Collidables" and the callback `OnCollidableObjectSpawned` is invoked.

Similarly, if a Collidable is despawner by the Spawner service the callback `OnCollidableObjectDespawnd` is triggered.

A method `IsCollidable()` can be used to see if a GameObject is in the HashSet.

This feature was added as part of the Layout system upgrade, and is used to keep track of prefabs that are spawned by tiles or otherwise – its used to ensure that spawned prefabs in a Chunked Zone (region) can be removed when the Zone’s tiles are removed.

This can get complicated since spawned GameObjects may not be in the same position when they're deleted. Keeping track of them ensures that they are removed only when the Zone that they're actually in is removed.

- You can see this in action during the Layout demo. The NPC characters are collidables; they move around but are despawned based on where they are when the Zone is deleted rather than where they were initially spawned.
- When using the Layout system this is all handled automatically.

The Layout system removes all tiles and GameObjects in a Zone, but it only handles GameObjects parented to one of the Tilemaps controlled by the layout system. The Layout system doesn't know what GameObjects might have been spawned by tiles (such as TpAnimatedSpawner) or exactly what Zone the GameObjects might be in since they might have moved from their original spawning location.

ZoneAnimator

Like other Zone-type tiles, it depends on you sending it a message with a position using TpMessaging, the TilePlus messaging system. Depending on how the Tile is set up, various actions will occur at StartUp() and/or when the tile's internal state changes from 'position is within the Zone' to 'position is outside of the Zone'.

That state change occurs when a message with a new position being sent to the tile.

The Zone is set up in the TilePlusBase section seen when inspecting the tile.

It's merely a BoundsInt which describes an area relative to the tile's position on a Tilemap.

The tile's code is set up to handle six basic types of operation based on a value from the PrefabMode enumeration:

1. PresentWhenNotInZone: Spawn the prefab when the tile's Start method is executed; despawn it when the Zone is entered and respawn it when the Zone is exited.
2. PresentWhenInZone: Spawn the prefab when the Zone is entered and despawn it when the Zone is exited.
3. SpawnInZone: Spawn the prefab when the Zone is entered. Never re-spawn.
4. DespawnInZone: Spawn the prefab when the tile's Start method is executed; despawn it when the Zone is exited. Never re-spawn.
5. AnimInZone: Spawn the prefab when the tile's Start method is executed. Animation on/off when the Zone is entered/exited.

6. `AnimOutZone`: Spawn the prefab when the tile's `Start` method is executed and activate animation. Animation OFF on Zone entry and ON again on Zone exit.

The various state specifications change depending on which mode you select.

As you can see, some of these modes allow you to optionally start animations when the prefab is spawned. `AnimInZone` and `AnimOutZone` always require animations unless all you want to do is spawn a prefab at the tile's position when `StartUp` runs; there are much simpler ways to spawn prefabs.

There's also the option of using scriptable object plugins: `TpAnimatorActions` to handle animations and/or control prefab spawning for special cases; e.g., spawning a timed series of prefabs.

The base code will use the first animator that it finds via `FindComponentsInChildren`.

Optionally, the code won't halt animation when the Zone is exited; rather, you can have a `StateMachineBehaviour` do it; use the base class `AnimStateTilePingerBase` as the base class for your `StateMachineBehaviour`.

Those interested in using this complex tile should check out the source code and [this](#).

Revision #9

Created 22 June 2025 19:51:21 by Vonchor

Updated 22 July 2025 17:32:15 by Vonchor