Unity 2D Toolkit

2D Toolkit

- $65 on the Asset Store
- Current version is 2.00/2.10
- This is based on 1.92 (breaking changes with 2.0)
- http://unikronsoftware.com/2dtoolkit/

Cameras

- Standard Unity Camera in Isometric mode
  - Pros: Familiar, scrolls easily
  - Cons: Scales, rather than uses pixel-perfect display
- tk2dCamera
  - Pros: Uses pixel units
  - Cons: Doesn't behave like a standard Camera
    - Bottom-left is always origin
    - Must adjust items within scene to scroll

Sprite Collection

- Asset
- Builds sprite sheets out of individual images
- Optimizes based on Camera/Player settings
- Advantage over Orthello/Futile/etc.: you don't need an external program for this
- Allows you to edit bounding boxes (polygon mode)

Sprite Animations

- Asset
- Requires Sprite Collection
- A link between "Sprite Collection" and "Animated Sprite"

Sprite

- Requires Sprite Collection
- Is a Unity object that can take other components

Animated Sprite

- Requires Sprite Animation
- Uses Sprite Animation vs. Sprite Collection
- Play("*animationName*")
- IsPlaying("*animationName*")
- Stop()

Static Sprite Batcher

- Used to bake down multiple sprite objects into a single batch for performance

Text Mesh

- Requires a BMFont definition and image
  - BMFont: http://www.angelcode.com/products/bmfont/
  - Glyph Designer (Mac)
  - You can also draw your own manually (but hard to troubleshoot)
- Can update dynamically (call Commit() afterwards)

Patterns I Use

Warning: may contain anti-patterns

Creating Moving Objects

- Sprites still behave as 3D objects, so lock Z movement and X/Y rotation
- Use a Collider generated by the Sprite Collection
- RigidBody.AddForce(*vector*, ForceMode.VelocityChange)

Creating Platformer Movement

- Use CharacterController
  - isGrounded
  - Move method returns detailed collision info (side/above/below) immediately
  - Haven't found anything (free) that replicates this
- CharacterController uses a capsule collider
  - Tweak at runtime to figure out the proper Step, Height, Radius, and Skin Width
  - Remember that capsule collider curves along Z axis
- Must disable the Box Collider (set isTrigger, possibly in your custom controller script)

Creating Platformer Movement

Script from: https://www.youtube.com/watch?v=EW0phq6xoJk&list=PLA5781666685406E1

- Applies its own gravity rather than trying to use physics (feels tighter and more controlled than built-in physics)
- Can adapt this script to use input from an AI logic script

"Is it a...?"

When interacting with other objects (i.e., triggers):

```
if (obj.GetComponent<MyType>() != null) {
   // do something
}
```

- Components define object capabilities, even if I don't interact with their methods or fields
- There are probably better ways to do this.

Layering / Extending Components

- Develop general components which are used by more specific/variant components
- Example:
  - AIMotor knows about the Unity CharacterController
  - WanderAI, HopWanderAI, etc. read and write exposed fields on AIMotor to determine state and move the NPC
- Example:
  - PlatformerController knows how to move the character around; doesn't care what it's attached to
  - PlatformerPlayerAnimation is attached to a tk2dAnimatedSprite and inteprets exposed state fields on PlatformerController to animate it

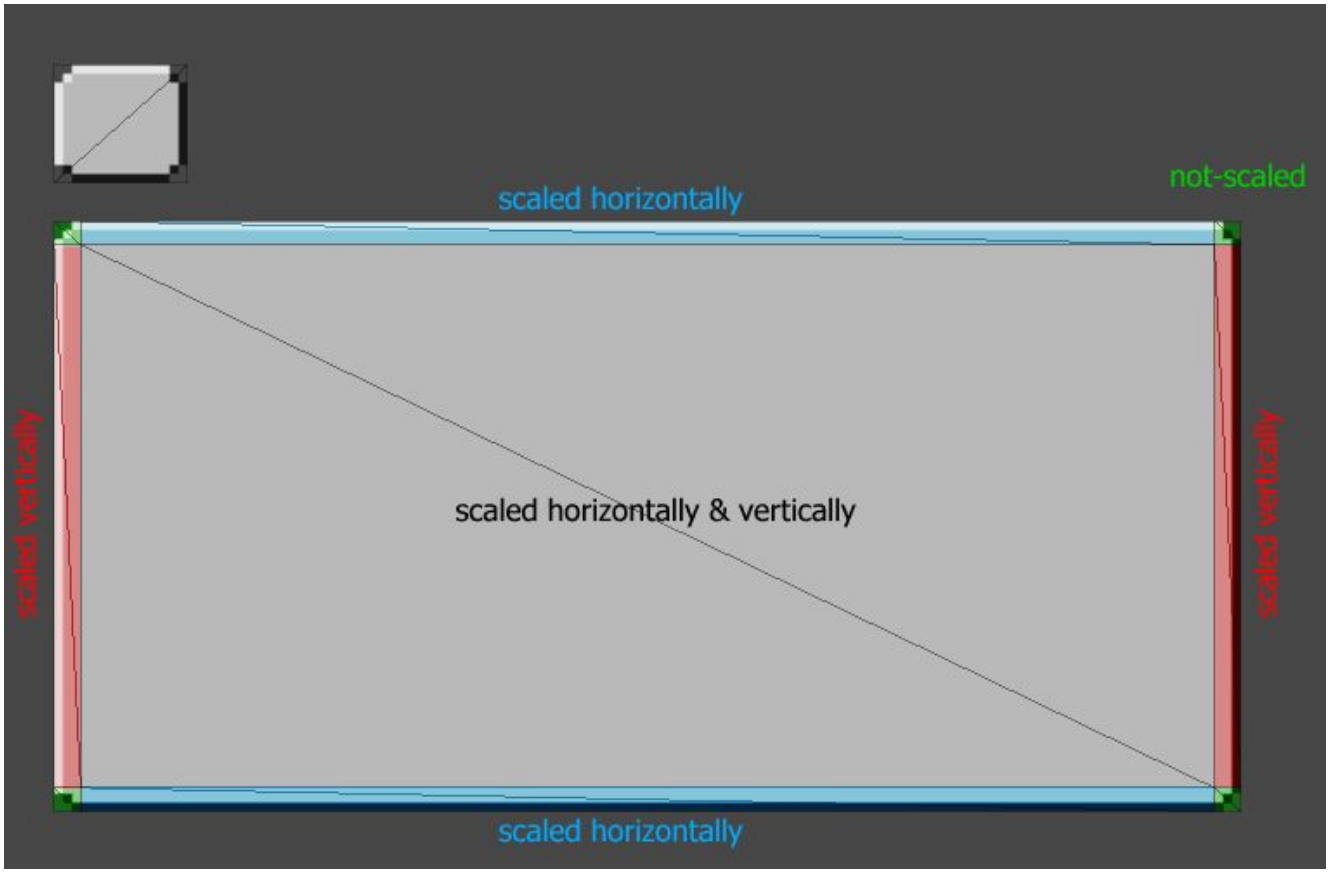
Destroyed Objects Become Null

- You can keep references to other objects in your components, then test if they are destroyed by checking whether they're null
- Example:
  - AISpawner has a maximum number of spawns, and keeps a list of enemies it has spawned
  - When an enemy is killed, it is destroyed with DestroyObject
  - When AISpawner fires, it goes through the list and removes all NULL objects, then tests whether it is under its spawn limit
- This isn't exactly how .NET normally works

Stuff I'm not familiar with...

Sliced Sprite

- Used to create "windows," etc.

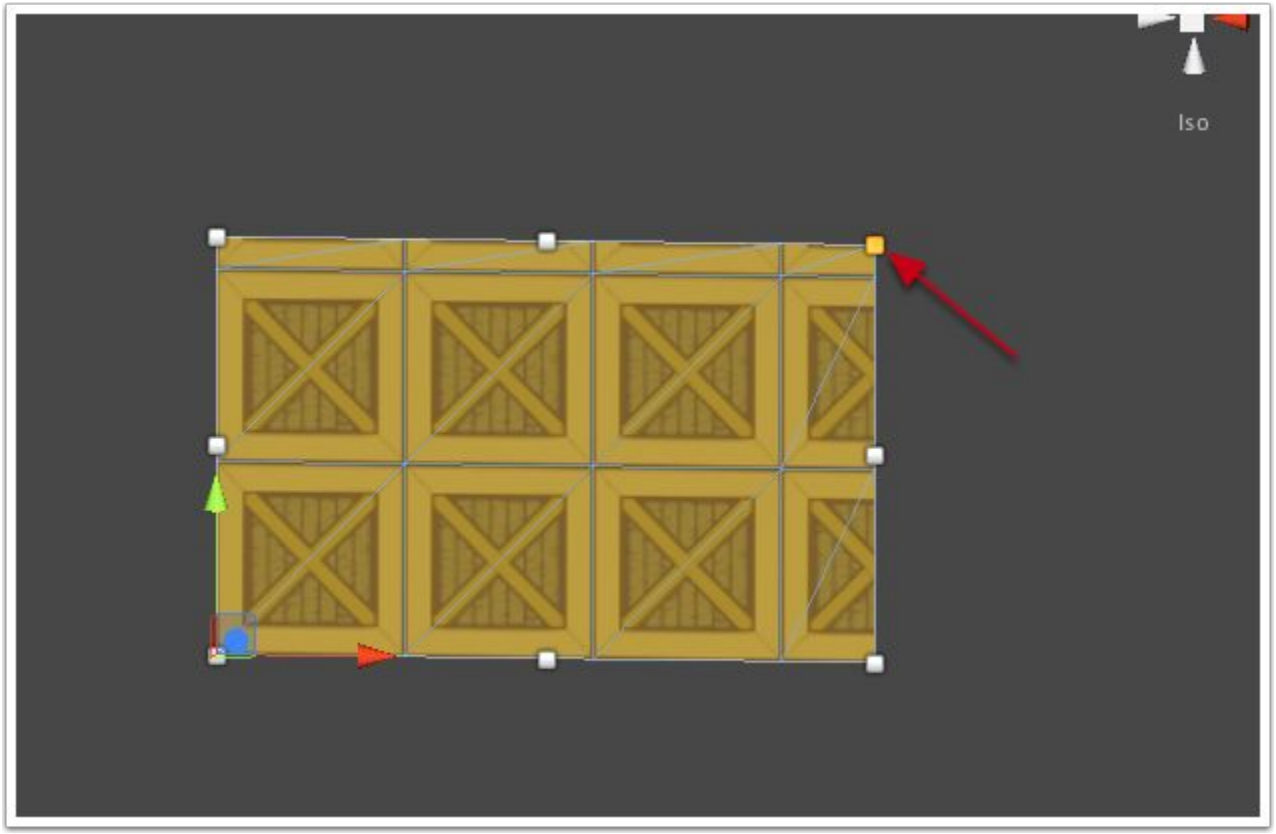


Clipped Sprite

- Displays a clipped rectangle from a sprite

Tiled Sprite

- Creates an area filled in with a repeated pattern



Tile Maps

- Beta
- Allows you to "paint" a map based on a tileset
- Can import .tmx files