

Creating Tile Groups Programmatically

Paint tiles on a map by putting tile definitions in a group that you create in code.

Overview

To paint tiles on a map using a tile group, you call `fill(with:)` on the group. The fill behaviors you use depend on the type of group you make:

Group type	Fill behavior
Single tile	Repeats the single center tile.
Multiple tiles without adjacency rules	Chooses randomly among the multiple center tiles, abiding by the placement weights you choose.
Multiple tiles with adjacency rules	Repeats the center tile and borders all of its edges using the peripheral adjacency tiles you define.

Create a Tile Group with a Single Tile

Although a tile group is a container for multiple tiles, you can create one with a single tile, if you wish. Use the `init(tileDefinition:)` initializer, as shown in the following code.

```
let backgroundTexture = SKTexture(imageNamed: "BG.png")
let backgroundDefinition = SKTileDefinition(texture: backgroundTexture)
let backgroundGroup = SKTileGroup(tileDefinition: backgroundDefinition)
```

Create a Heterogenous Tile Group with Random Placement

For a tile group with more than one texture, use the `init(rules:)` initializer. The following code shows how you can create a tile group containing two definitions—one based on an image named RED.png and the other on an image named BLUE.png.

```
let redTexture = SKTexture(imageNamed: "RED.png")
let redDefinition = SKTileDefinition(texture: redTexture)
redDefinition.placementWeight = 3

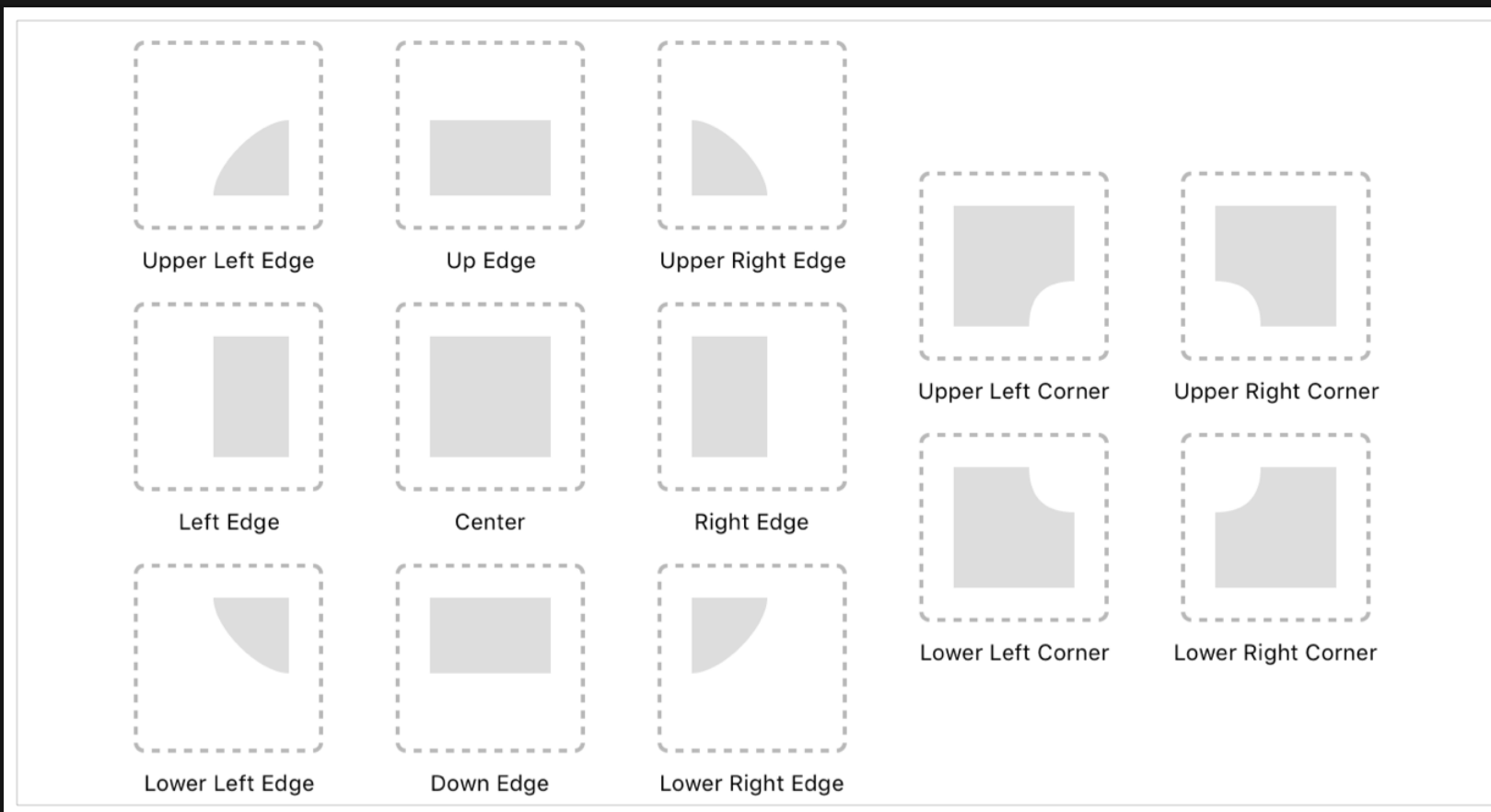
let blueTexture = SKTexture(imageNamed: "BLUE.png")
let blueDefinition = SKTileDefinition(texture: blueTexture)
blueDefinition.placementWeight = 1

let redBlueRule = SKTileGroupRule(adjacency: .adjacencyAll,
tileDefinitions: [redDefinition, blueDefinition])
let group = SKTileGroup(rules: [redBlueRule])
```

When a tile map node is supplied with a tile set using this group and its tiles are filled with the `fill(with:)` method, the placement weights control the random assignment of tiles. In this case, there are three times more red than blue tiles.

Create a Tile Group Using Adjacency Rules

To create a tile group that uses adjacency, you need to create a set of rules for each of the adjacency values. For a grid-based layout, you need 13 rules, as shown in the following illustration:



To create the tile group programmatically, create definitions and rules for each adjacency:

```
let centerTileDefintion = SKTileDefinition(texture: SKTexture(imageNamed:
"CENTER.png"))
let centerRule = SKTileGroupRule(adjacency: .adjacencyAll,
                                tileDefinitions: [ centerTileDefintion ])

let upEdgeDefinition = SKTileDefinition(texture: SKTexture(imageNamed: "N.png"))
let upEdgeRule = SKTileGroupRule(adjacency: .adjacencyUpEdge,
                                tileDefinitions: [ upEdgeDefinition ])

let downEdgeDefinition = SKTileDefinition(texture: SKTexture(imageNamed: "S.png"))
let downEdgeRule = SKTileGroupRule(adjacency: .adjacencyDownEdge,
                                tileDefinitions: [ downEdgeDefinition ])

[...]
```

The 13 rules are used to make a tile group:

```
let group = SKTileGroup(rules: [centerRule, upEdgeRule, downEdgeRule, ...])
```

With automapping enabled, when an individual tile is placed on the map using `setTileGroup(_:forColumn:row:)`, SpriteKit also sets the tile definitions for the surrounding tiles based on the group's adjacency rules.