

```
//  
// Enums.swift  
// SFGGridScape  
//  
// Created by David on 27/11/2018.  
// Copyright © 2018 David. All rights reserved.  
//
```

```
/// Specifies grid line types  
public enum GridLineTypes {  
    case horizontal  
    case vertical  
}
```

```
/// Specifies cell neighbour position types  
public enum CellNeighbourPositionTypes {  
    case top  
    case topRight  
    case right  
    case bottomRight  
    case bottom  
    case bottomLeft  
    case left  
    case topLeft  
}
```

```
/// Specifies cell side types  
public enum CellSideTypes {  
    case top  
    case right  
    case bottom  
    case left  
}
```

```
/// Specifies cell content position types  
public enum CellContentPositionTypes: Int {  
    case Unspecified = 0  
    case Center = 1  
    case TopCenter = 2  
    case TopRight = 3  
    case RightCenter = 4  
    case BottomRight = 5  
    case BottomCenter = 6  
    case BottomLeft = 7  
    case LeftCenter = 8  
    case TopLeft = 9  
}
```

```
/// Specifies cell group types  
public enum CellGroupTypes {  
    case LineUp  
    case LineUpRight  
    case LineRight  
    case LineDownRight  
    case LineDown  
    case LineDownLeft  
    case LineLeft
```

```
    case LineUpLeft
    case Alternate
    case NonAlternate
    case DisplayEdge
    case Columns
    case Rows
}

/// Specifies grid cell property keys
public enum GridCellPropertyKeys {
    case rotationDegrees
}

/// Specifies grid tile property keys
public enum GridTilePropertyKeys {
    case rotationDegrees
}

/// Property-changed types
public enum PropertyChangedTypes {
    case Property
    case Attribute
    case SideAttribute
}

/// Specifies grid dragging types
public enum GridDraggingTypes {
    case Cell
    case Tile
    case Token
}

/// Specifies grid position reference to types
public enum GridPositionReferenceToTypes {
    case topleft
    case center
}

public enum MoveAlongPathResponseTypes {
    case Continue
    case Suspend
    case Cancel
}

public enum PathStatusTypes {
    case NotStarted
    case Started
    case Suspended
    case Finished
}
```