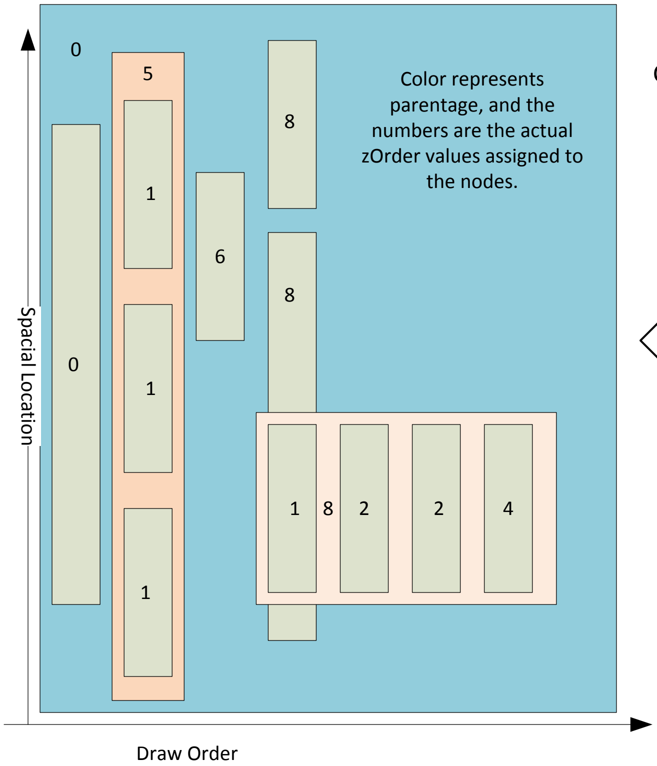
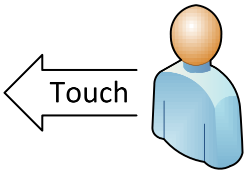
# GLOBAL Z AND TOUCH PROPAGATION



Color represents parentage, and the numbers are the actual zOrder values assigned to the nodes.

## Current Implementation

Touch is processed according to the list of touch delegates. No priority or draw ordering is used to bias the processing of touches. It is possible for the bottom most visible node to process and handle the touch, leaving the top node starving for touches.



{ Delegate List }

0, 8, 8, 6, 5, 5.1, 5.1, 8, 8.2, 5.1, 8.1, 8.4, 8.2

This delegate handles the touch and stops the touch processing.

Intended target

## Proper Implementation

Process the touch according to the zOrder bias. This would handle the touch delegates in their draw order, using a LIFO queue for the nodes.

{ Delegate List }

8.4,8.2,8.2,8.1,8,8,8,6,5.1,5.1,5.1,5,0,0

This delegate handles the touch and stops the touch processing.

## The Problematic Implementation

CCScrollView and CCMenu manage their touch delegates by specifying a non-zero touch priority.

```
if (m_eTouchMode == CCTouchMode.AllAtOnce)
{
    pDispatcher.AddStandardDelegate(this, 0);
}
else
{
    pDispatcher.AddTargetedDelegate(this, m_nTouchPriority, true);
}
```

Note the priority=0 which makes all touches act with the same priority.

## The Proposed Change

```
public interface ICCTouchDelegate
{
    int TouchPriority { get; }
}
```

This method already exists in CCNode, but it is not derived using the draw order.