Single channel sequencer algorithm

Abstract: In the current design of the application, all presentations play in the way that all channels in a layout are opened and closed simultaneously. The purpose of this algorithm is to break layout-wise opening and closing. Once one channel window is closed, the best matching channel of the next layout is consecutively played.

1. Known parameters

Each layout(template) has common parameter “duration” which means that all channels in this layout will play for “duration” seconds, i.e. all channels are opened concurrently and closed in “duration” seconds.

Each channel(window) in a layout has common parameter “duration”, usually the same value as the “duration” of the parent layout.

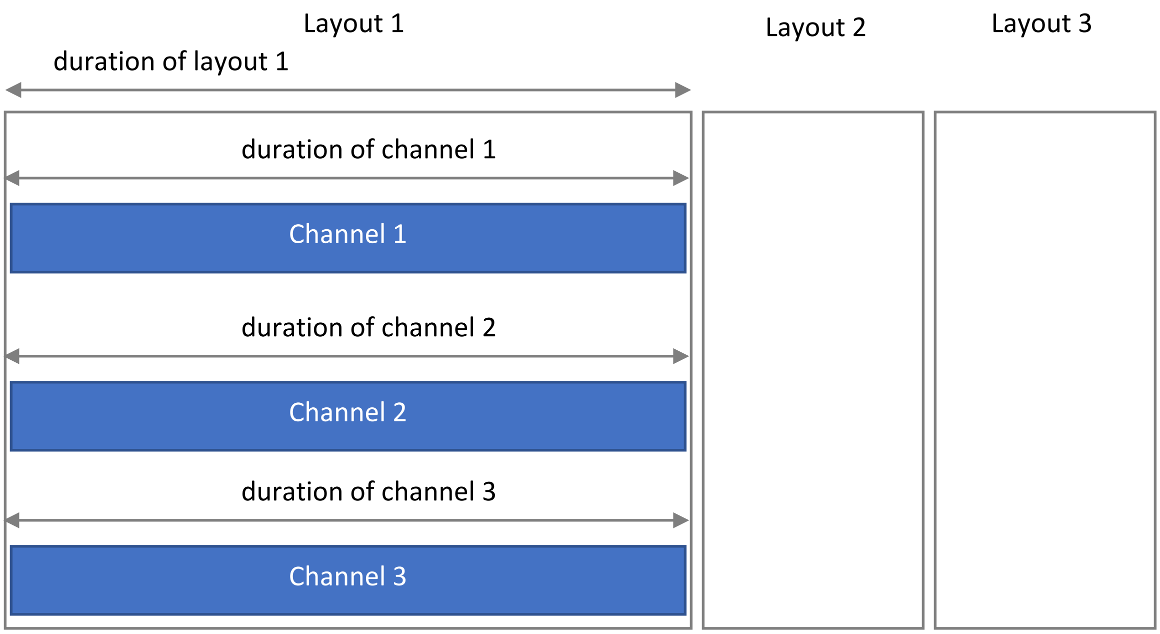


Figure 1: Known parameters

Now we edit presentations like as follows:

Duration of layout 1 = duration of channel 1 = duration of channel 2 = duration of channel 3

2. Condition of layout overlay playing

If “duration” value of each layout of the presentation is equal to the “duration” values of all channels of the layout, then the presentation will play in the way of layout-wise playing.

If “duration” value of at least one channel of some layout in the presentation is less than the “duration” value of the layout, then the presentation will play in the way of layout overlay playing.

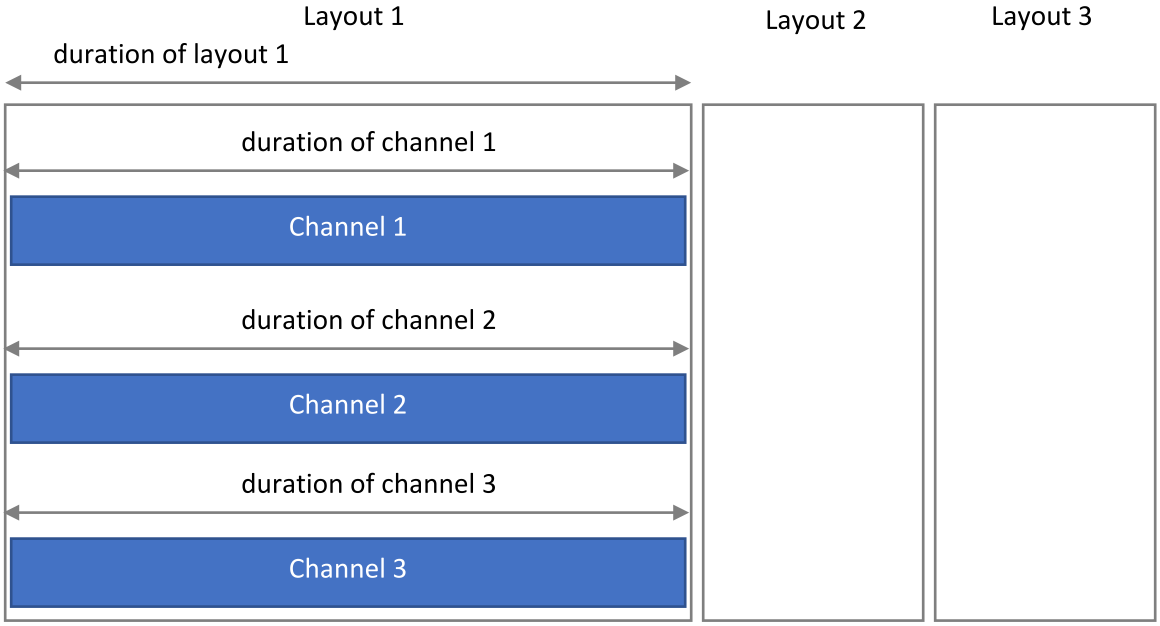


Figure 2: Layout-wise playing

In figure 2,

Duration of layout 1 = duration of channel 1 = duration of channel 2 = duration of channel 3

Duration of layout 2 = duration of channel 1 = duration of channel 2

Duration of layout 3 = duration of channel 1

If we edit the presentation as shown in figure 2, then the presentation will play in the way of layout-wise playing.

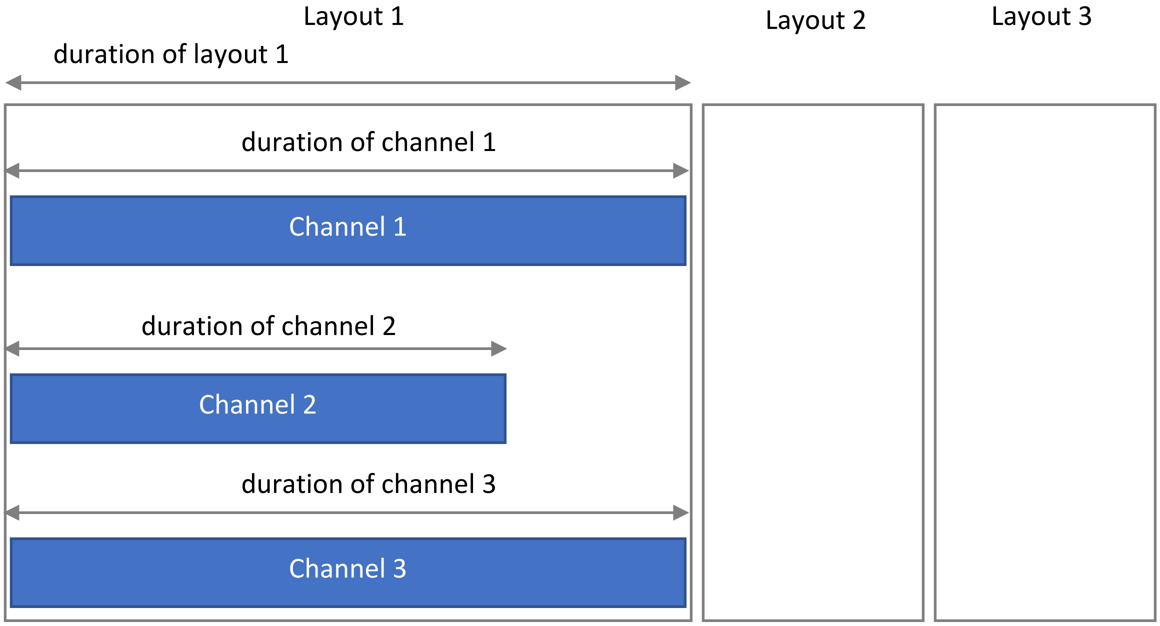


Figure 3: layout overlay playing

In figure 3,

Duration of the channel 2 < duration of the layout 1

Then the presentation will play in the way of layout overlay playing.

3. Best matching algorithm

Let’s suppose that the presentation is playing in layout overlay mode.

Then we do opening and closing not in layout-wise, but in channel-wise.

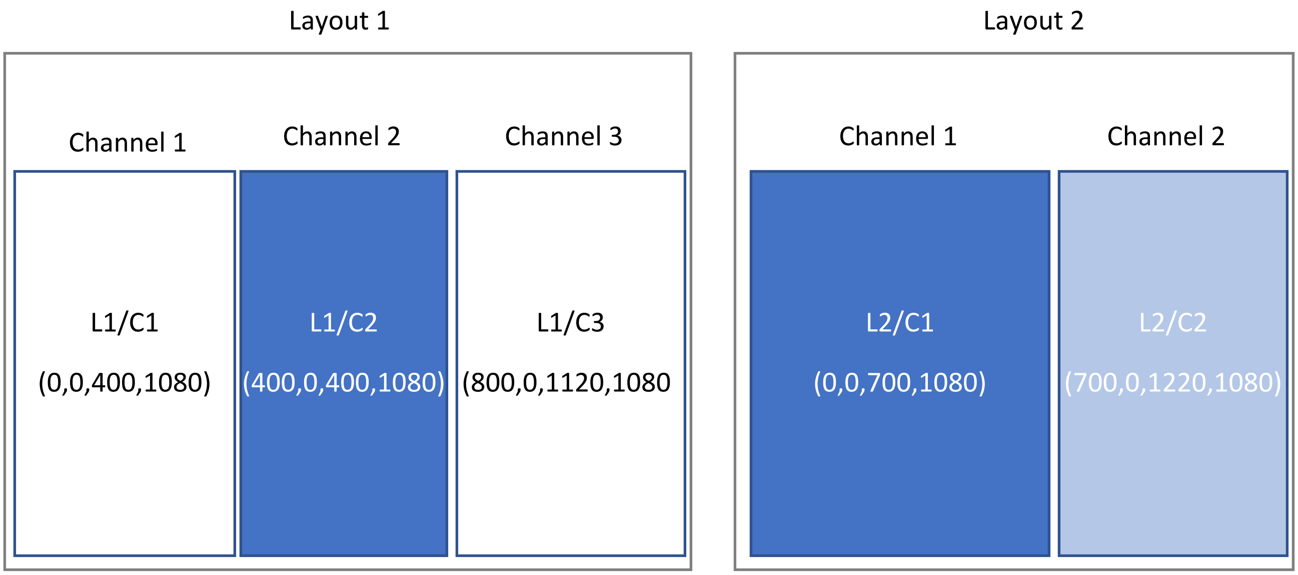
So when one channel is closed, then the best matching channel of the next layout will be played.

For example, in figure 3

“Channel 2” of “layout 1” is closed in “duration” seconds, then we find the best matching channel in the layout 2.

Algorithm is simple as follows.

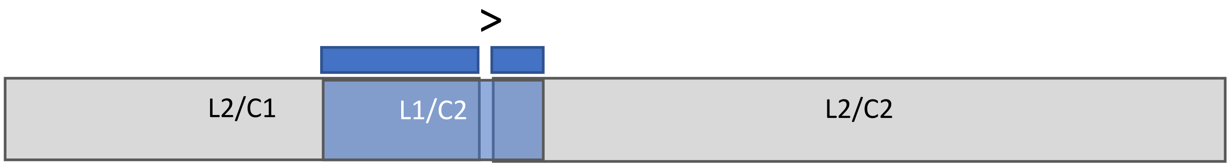
Let the rectangle of the “channel 2” of “layout 1” be L1/C2. Let “layout 2” has 2 channels and their rectangles be L2/C1, L2/C2.



The intersection of L1/C2 and L2/C1 is (400, 0, 300, 1080) and its area is 300x1080=324000.

The intersection of L1/C2 and L2/C2 is (700, 0, 100, 1080) and its area is 100x1080=108000.

The best matching channel is determined by the area of the intersection, so L2/C1 is the best matching channel.



4. Consecutive playing of layouts

When one channel of the layout is ended, then it finds out the best matching channel in the next layout.

When one channel of the last layout of the presentation is ended, then it finds out the best matching channel of the first layout of the presentation.

So, in this overlay mode, we can't say like "layout 2 is playing" because all channels are playing individually, independent of the layout.

But we have to indicate the number of currently playing layout. How?

That's done by the "duration" of the layouts.

For example,

Duration	Layout
20	Layout 1
30	Layout 2
10	Layout 3

At the 15 seconds of the presentation playing, the currently playing layout number is Layout 1.

At the 45 seconds of the presentation playing, the currently playing layout number is Layout 2.

At the 55 seconds of the presentation playing, the currently playing layout number is Layout 3.

In pseudo-code,

0 <= current timestamp < 20, then Layout 1.

20 <= current timestamp < 20 + 30, then Layout 2.

20 + 30 <= current timestamp < 20 + 30 + 10, then Layout 3.

Again,

20 + 30 + 10 <= current timestamp < 20 + 30 + 10 + 20, then Layout 1.

20 + 30 + 10 + 20 <= current timestamp < 20 + 30 + 10 + 20 + 30, then Layout 2.

20 + 30 + 10 + 20 + 30 <= current timestamp < 20 + 30 + 10 + 20 + 30 + 10, then Layout 3.

…

But the currently playing layout number is not pointing to actual playing channels of the layout.

That's all for my theory.

Thank you, Eric.