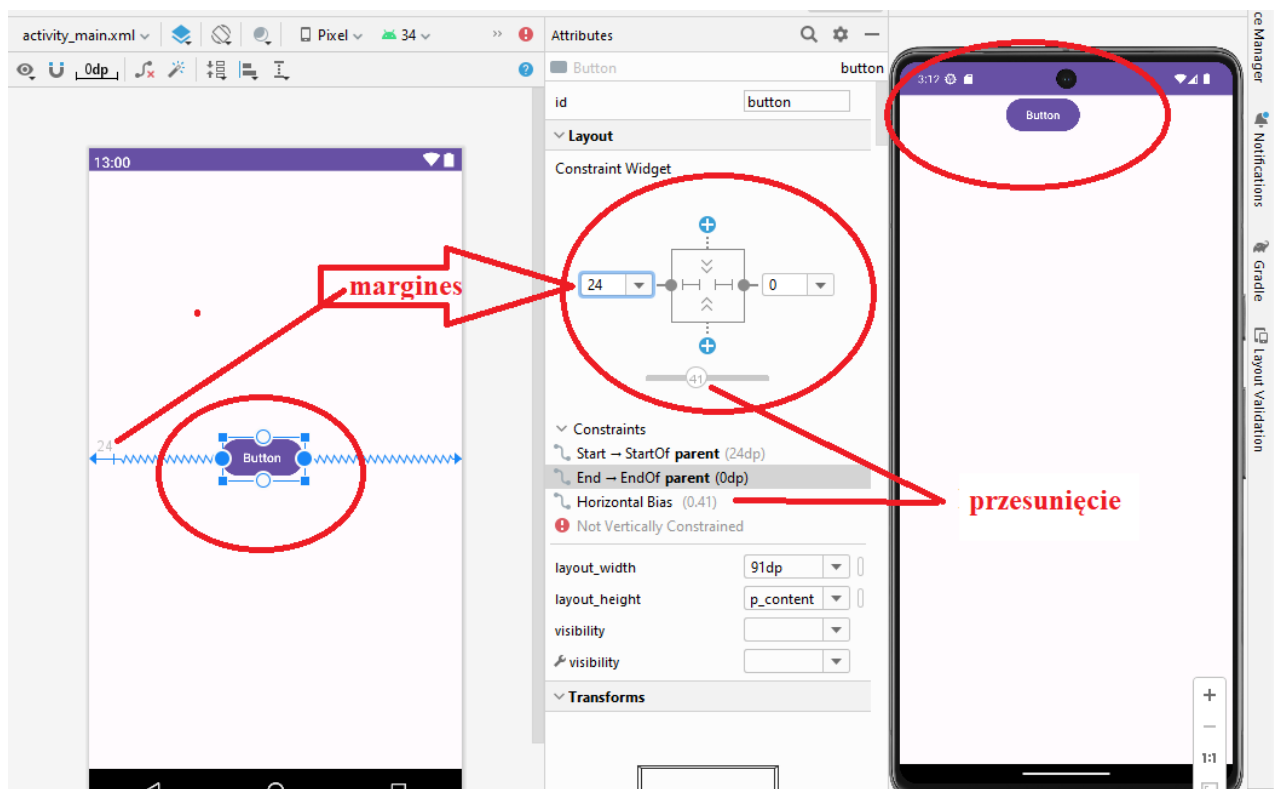
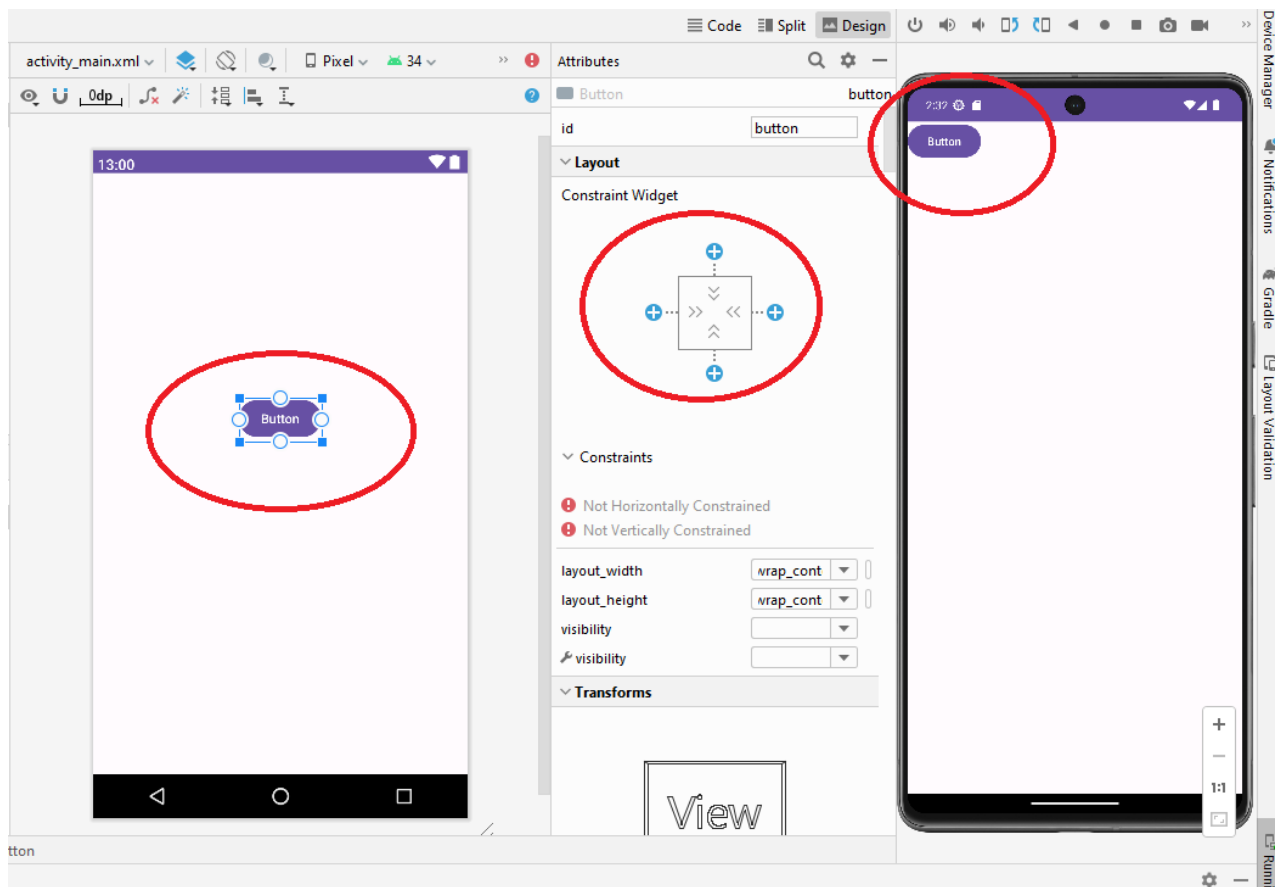
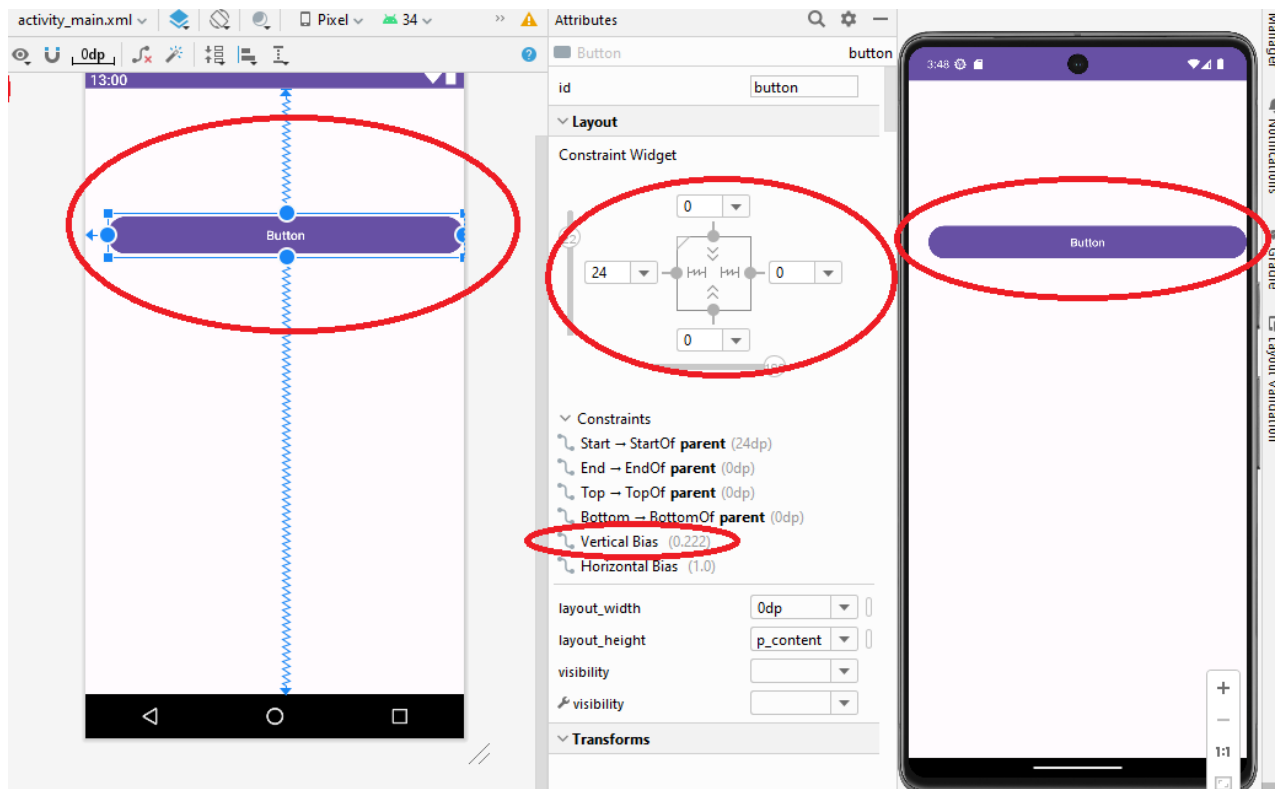
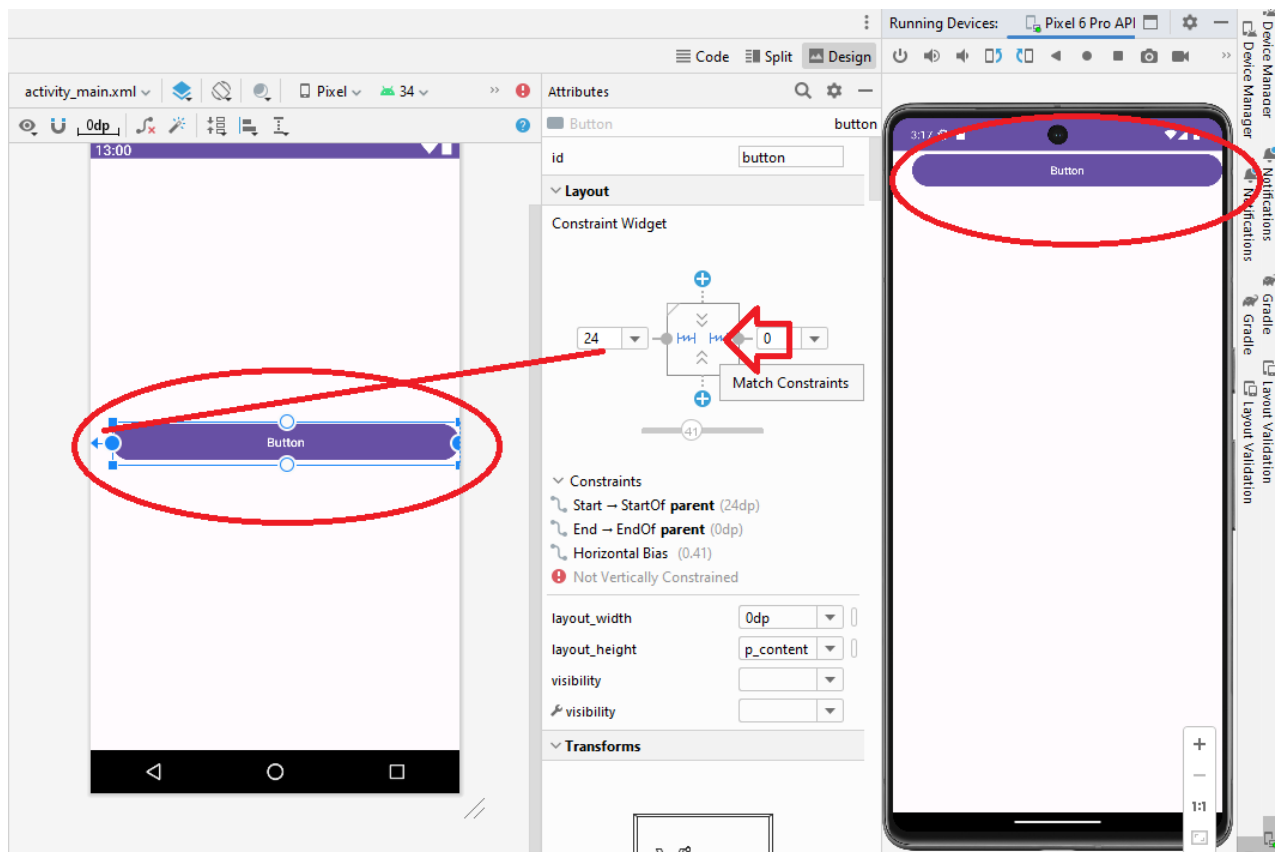


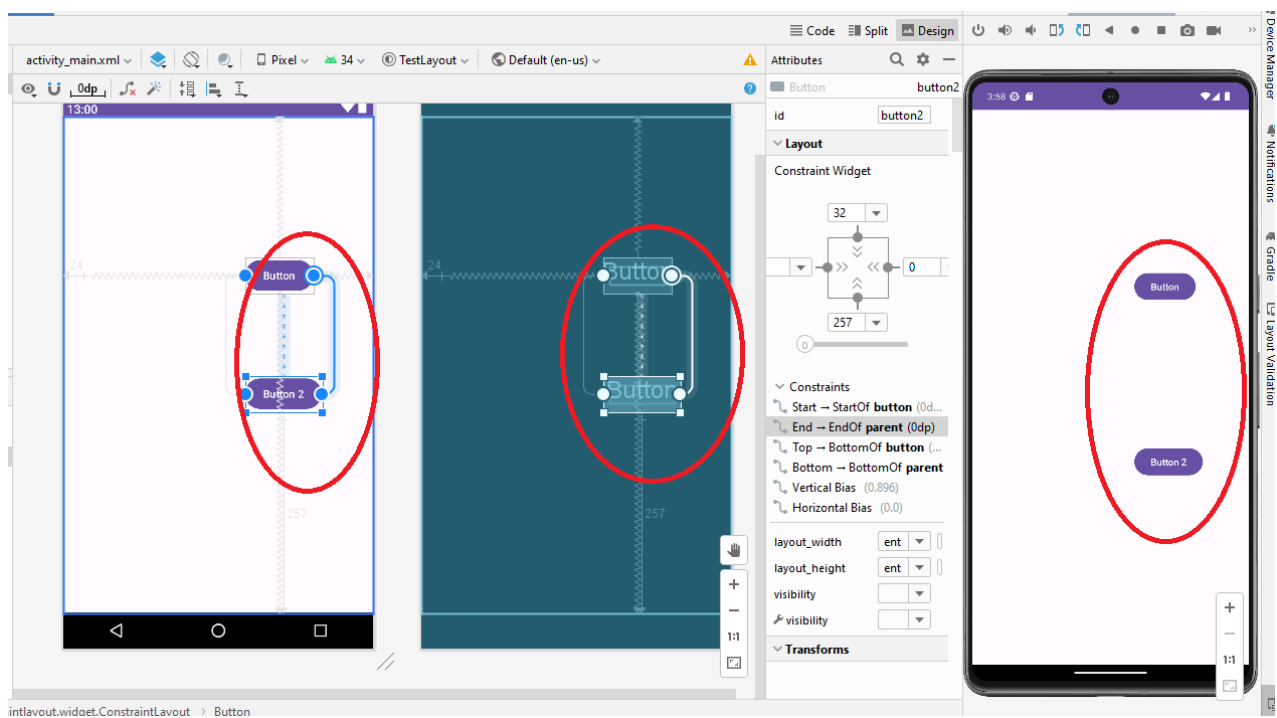
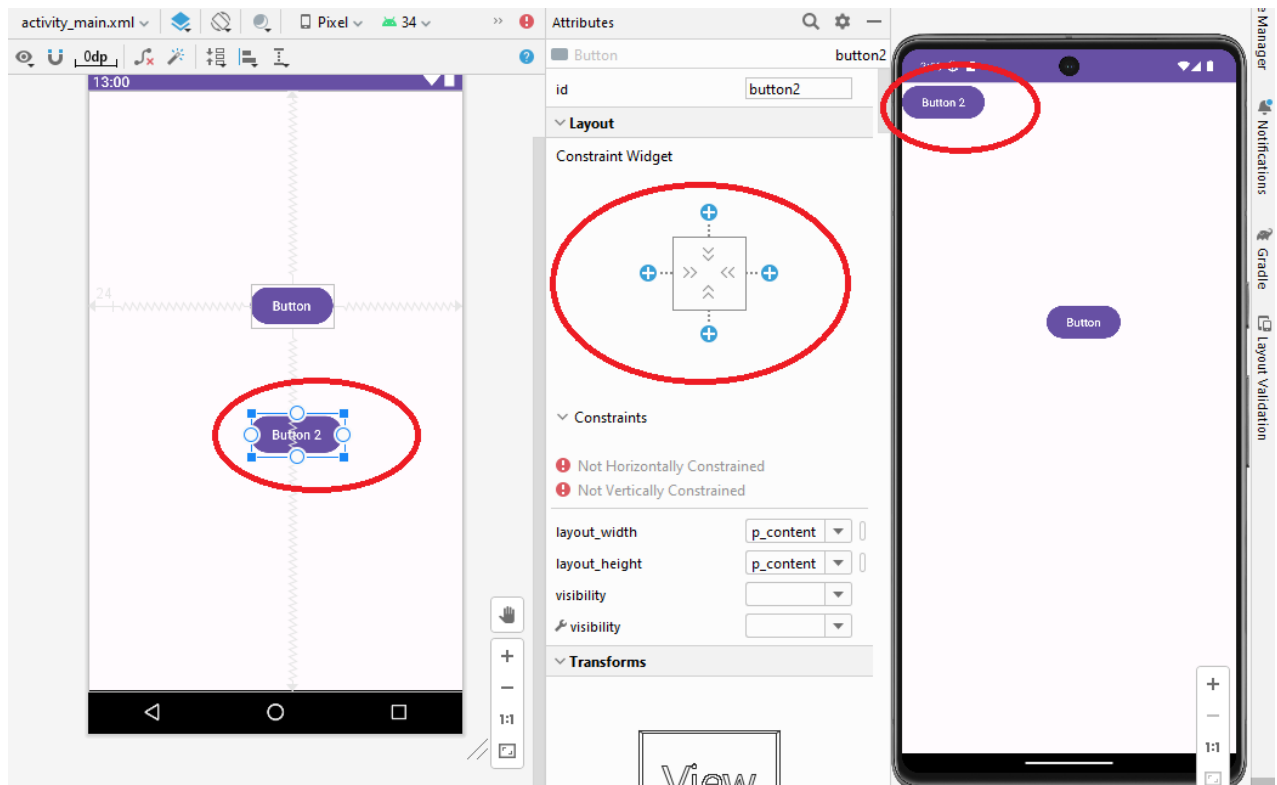
ConstraintLayout

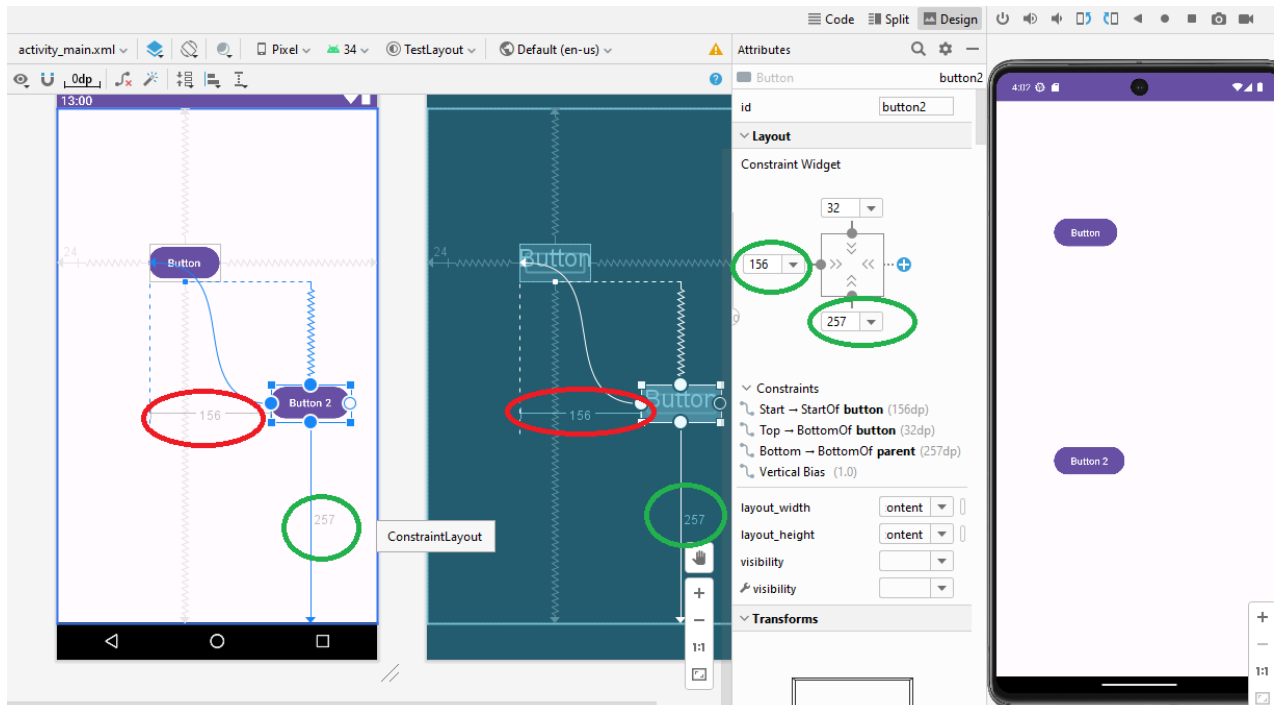
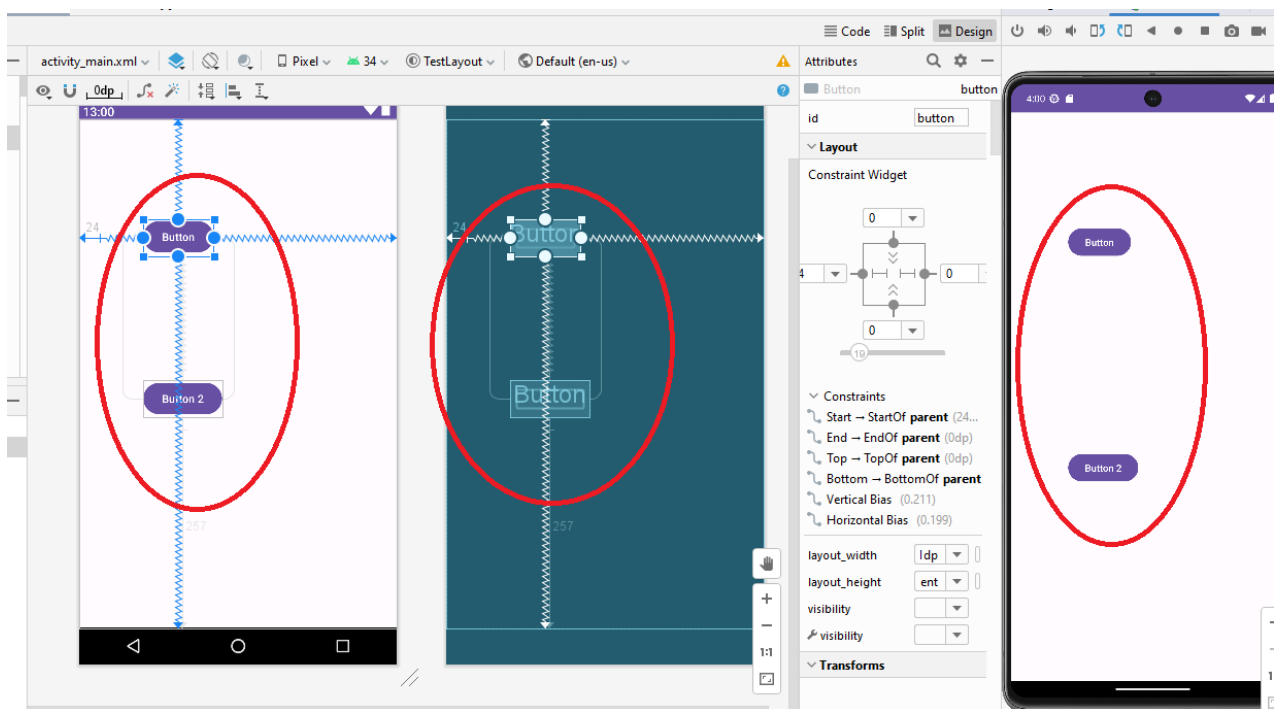
<https://developer.android.com/develop/ui/views/layout/constraint-layout>

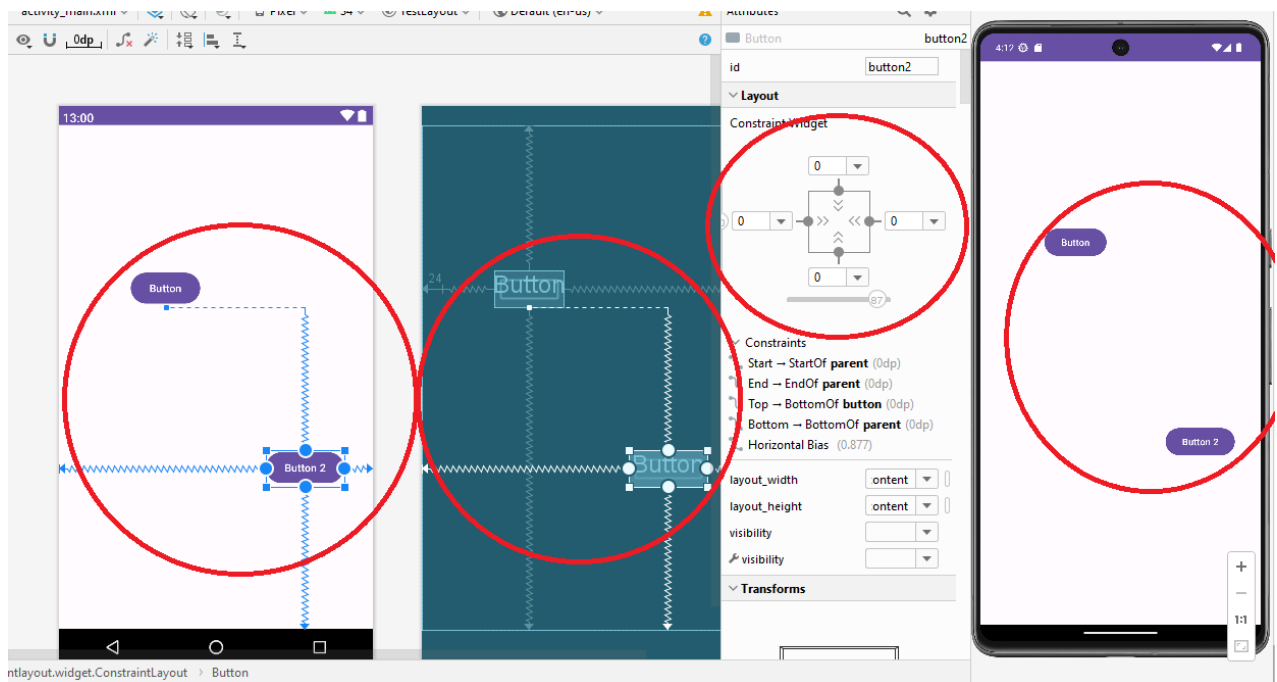
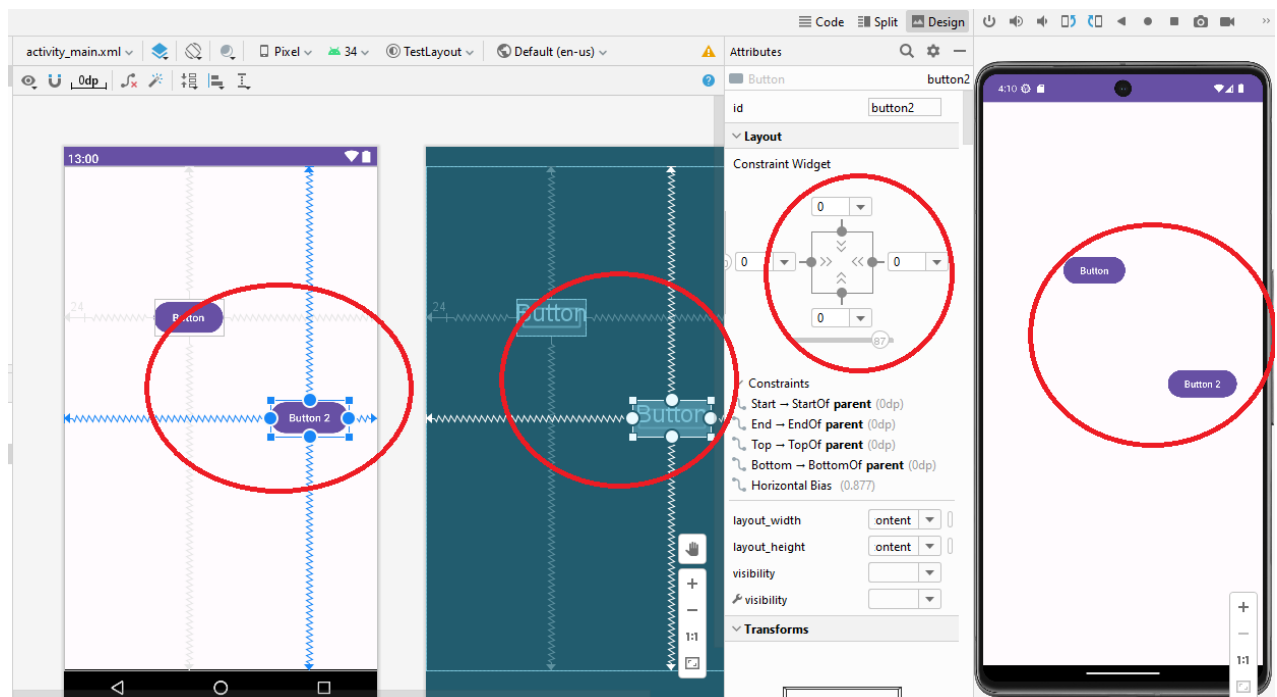
<https://www.youtube.com/watch?v=VsgXFdynDuQ>

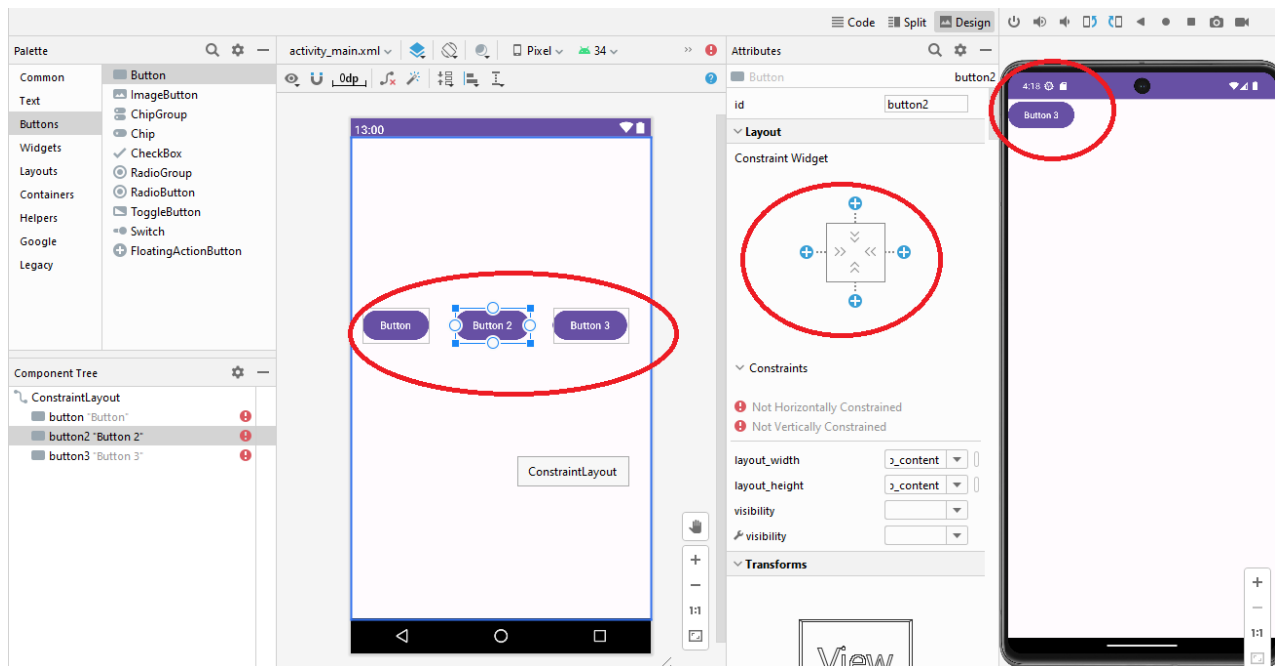




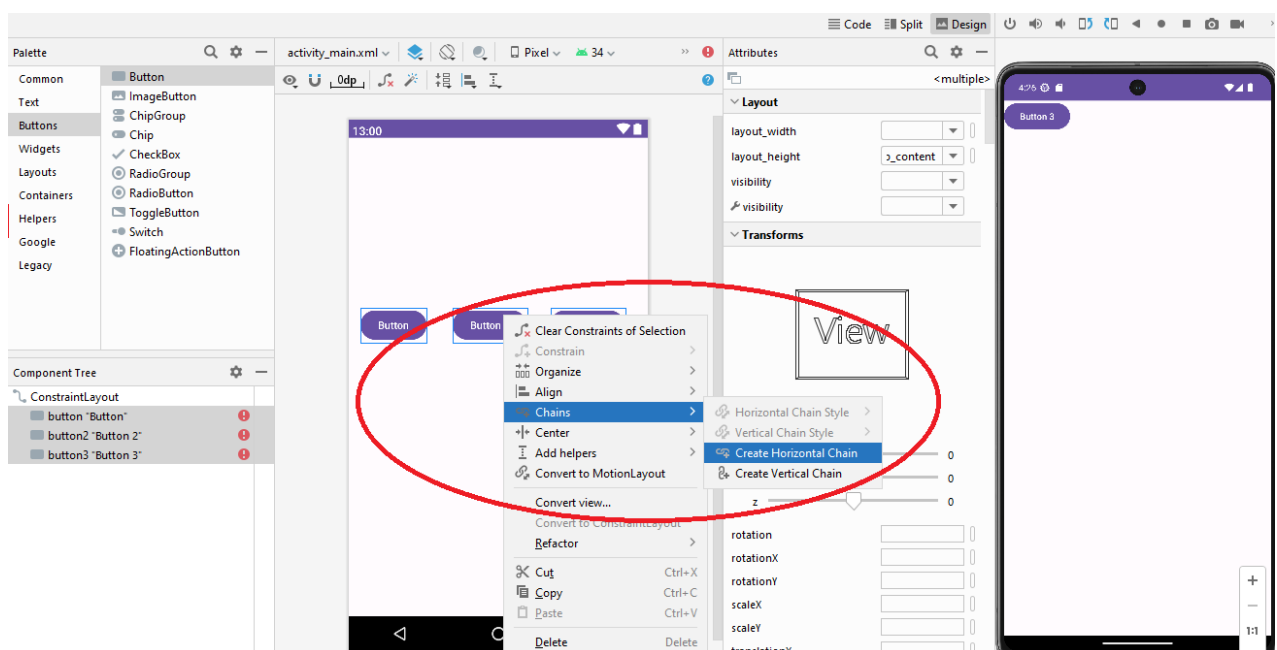


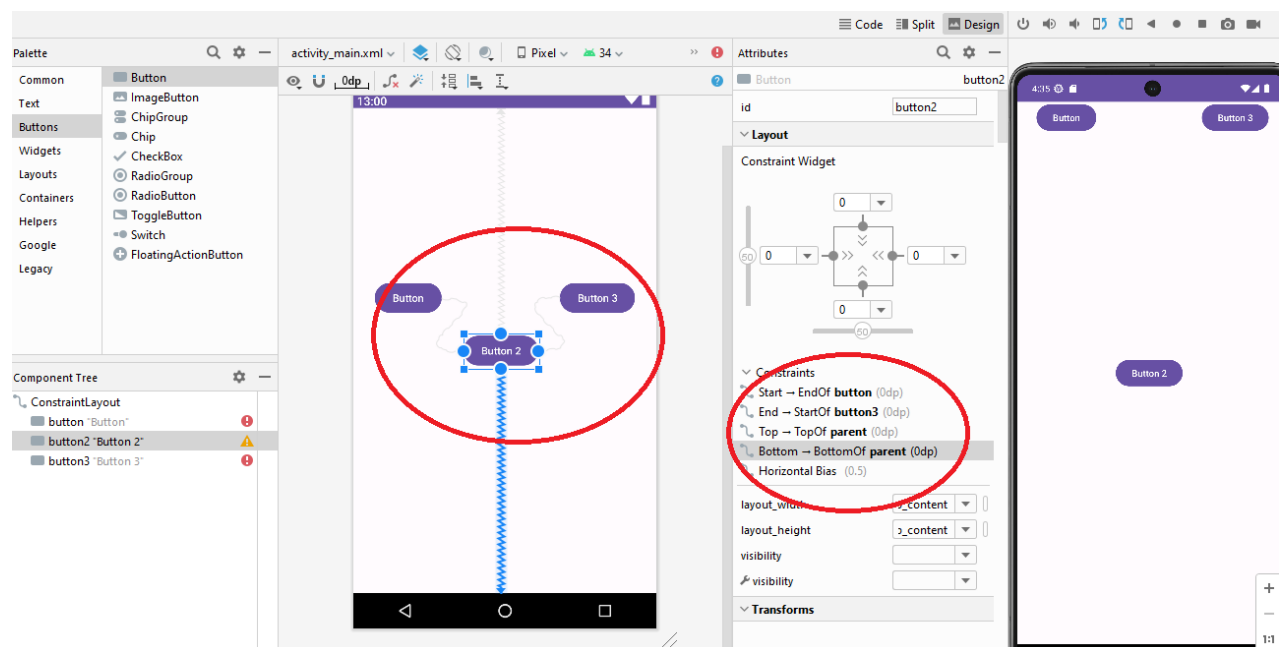
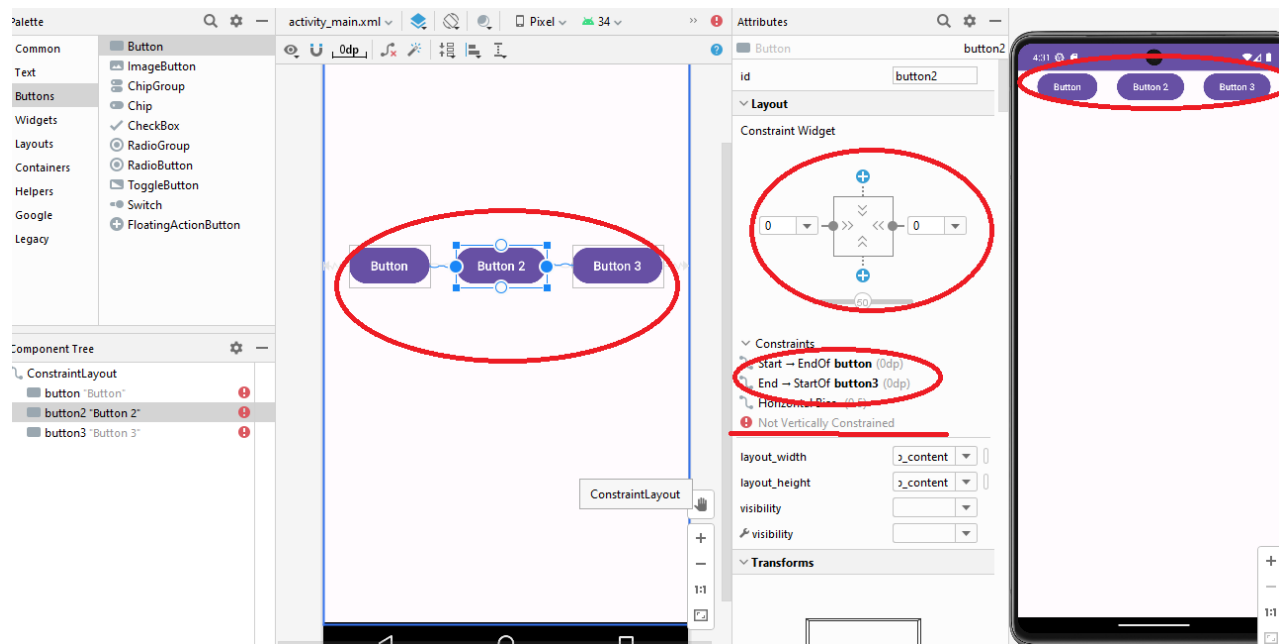




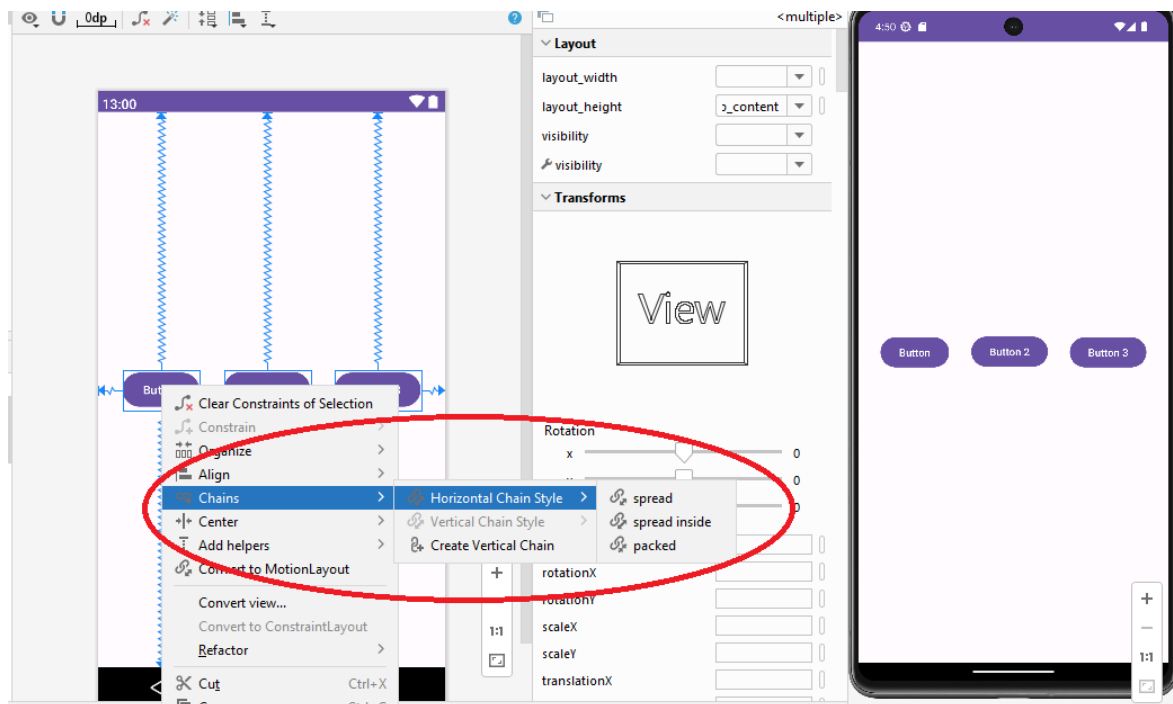


A chain is a group of views that are linked to each other with bi-directional position constraints. The views within a chain can be distributed either vertically or horizontally.

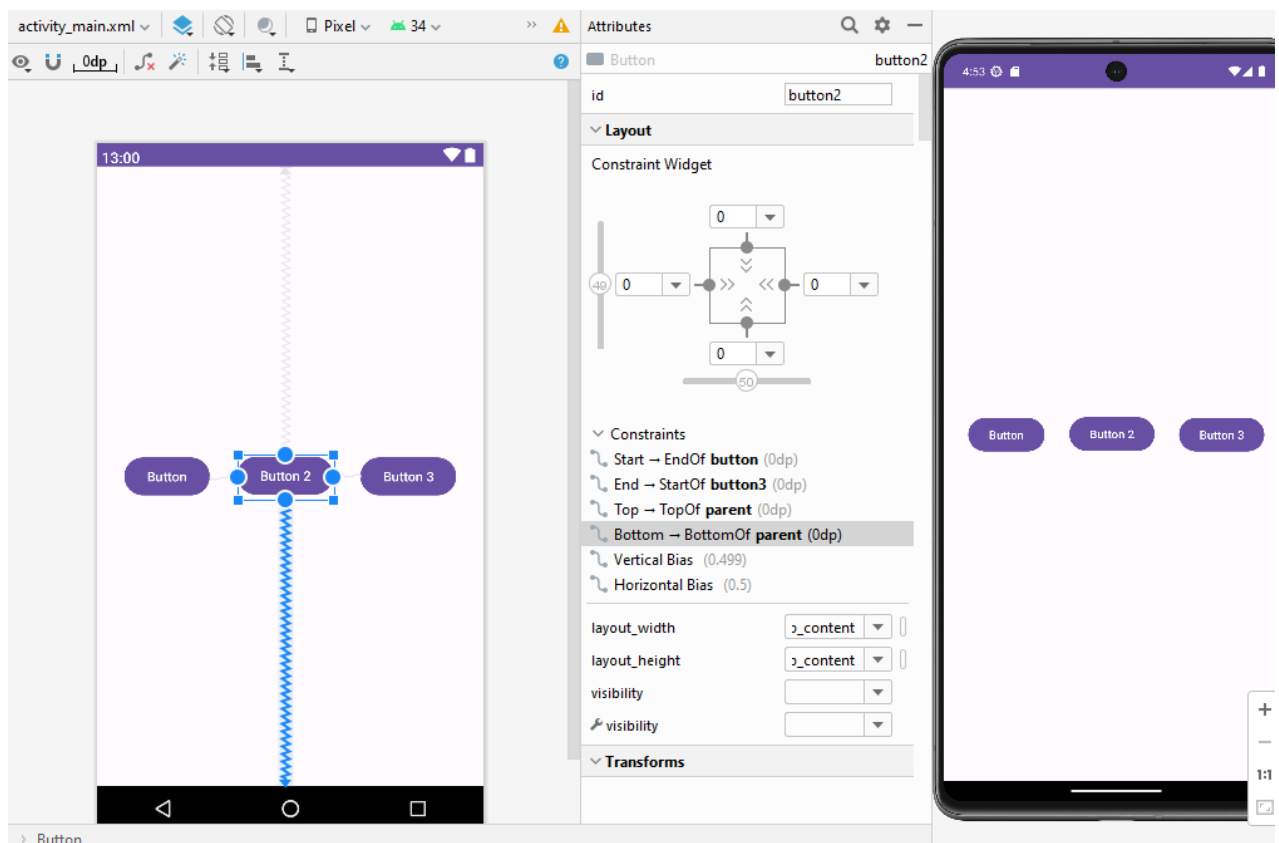




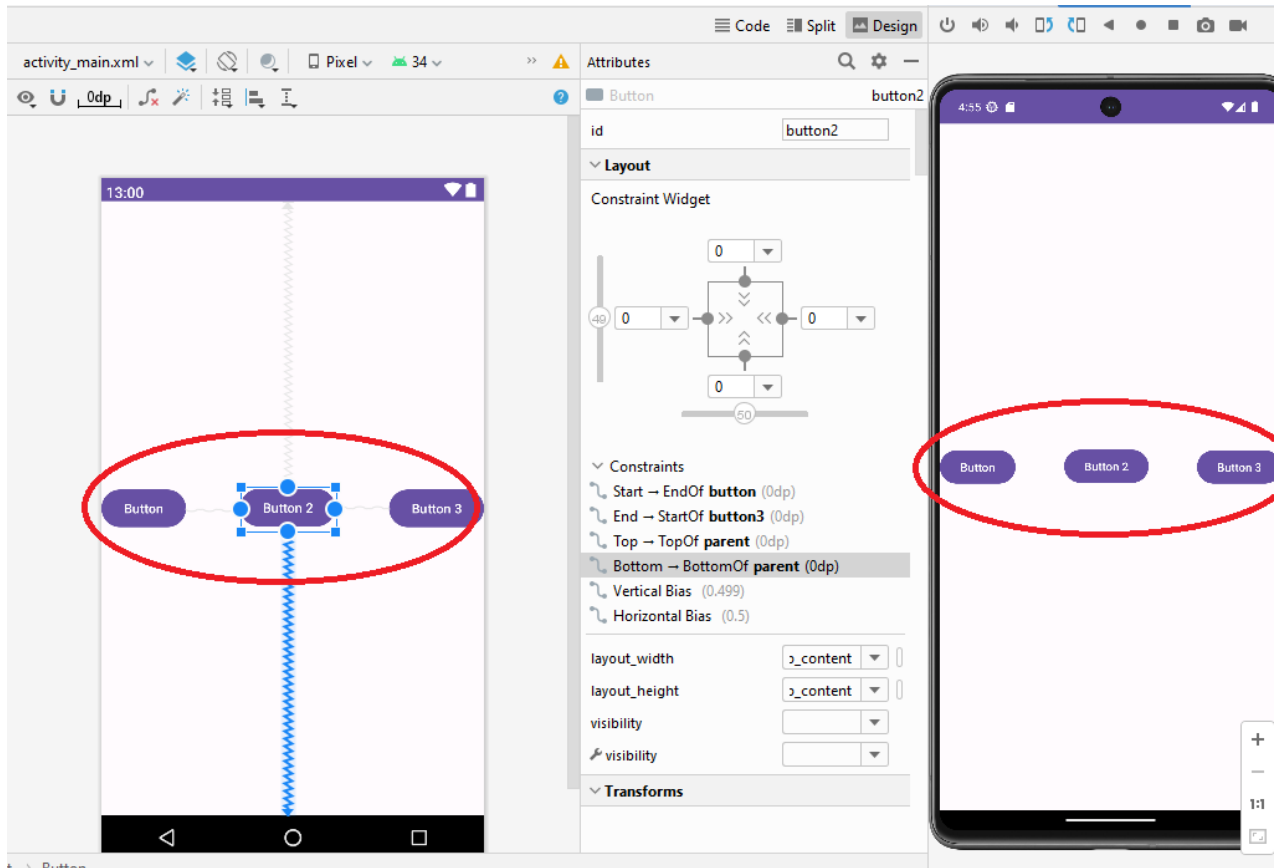
Chains can be styled in one of the following ways:



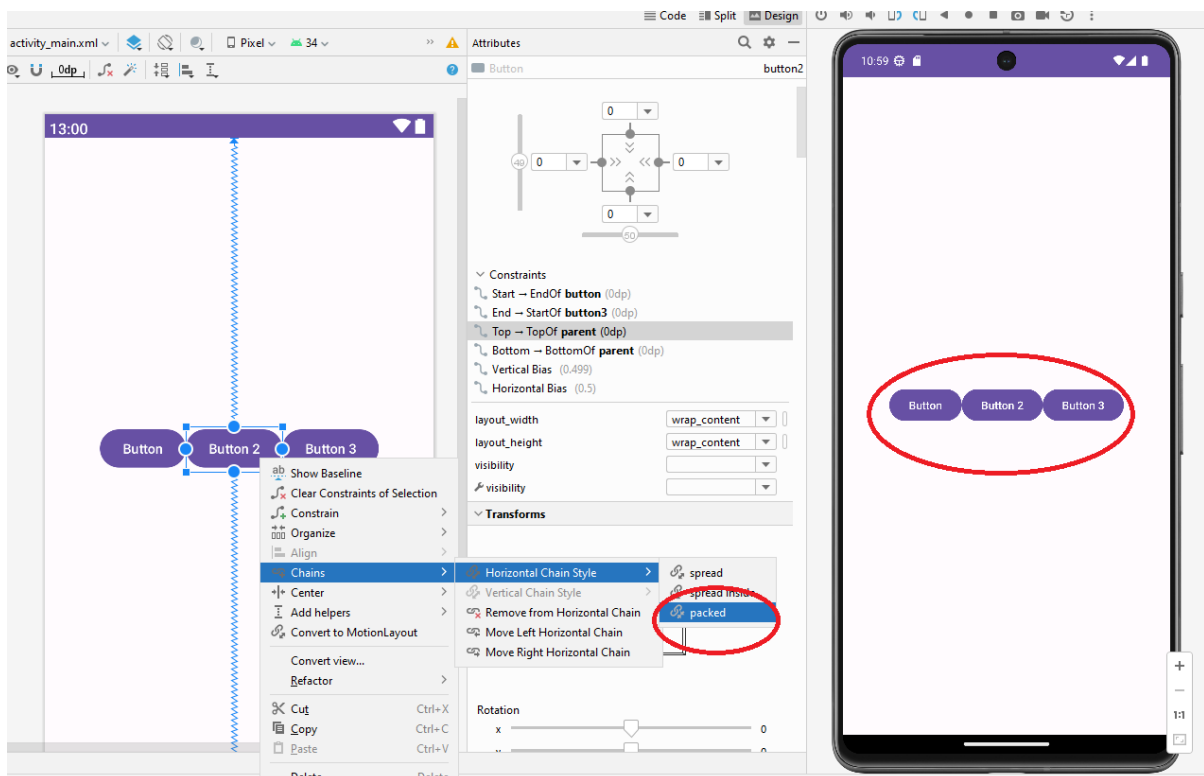
Spread: the views are evenly distributed after margins are accounted for. This is the default.



Spread inside: the first and last views are affixed to the constraints on each end of the chain, and the rest are evenly distributed.



Packed: the views are packed together after margins are accounted for.



Weighted: when the chain is set to **spread** or **spread inside**, you can fill the remaining space by setting one or more views to "match constraints" (0dp). By default, the space is evenly distributed between each view that's set to "match constraints," but you can assign a weight of importance to each view using the `layout_constraintHorizontal_weight` and `layout_constraintVertical_weight` attributes. This works the same way as `layout_weight` in a **linear layout**: the view with the highest weight value gets the most space, and views that have the same weight get the same amount of space.

