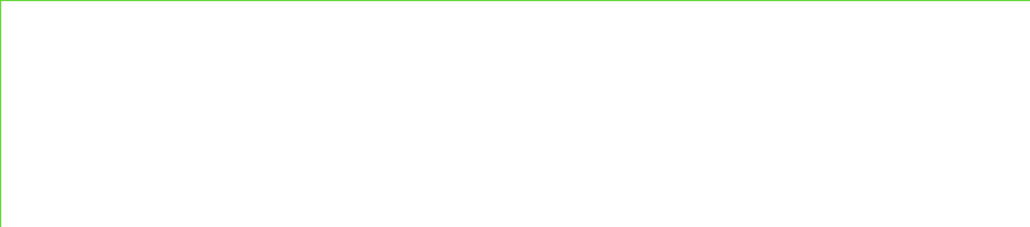



```
val bitmap = Bitmap.createBitmap(  
    /* width = */ width,  
    /* height = */ height,  
    /* config = */ Bitmap.Config.ARGB_8888  
)  
  
// API Level 28  
PixelCopy.request(  
    /* source = */ ...,  
    /* srcRect = */ Rect(rect),  
    /* dest = */ bitmap,  
    /* listener = */ { copyResult →  
        if (copyResult == PixelCopy.SUCCESS) {  
            // use the bitmap  
            ...  
        } else {  
            // Handle failure  
        }  
    },  
    /* listenerThread = */ ...  
)
```

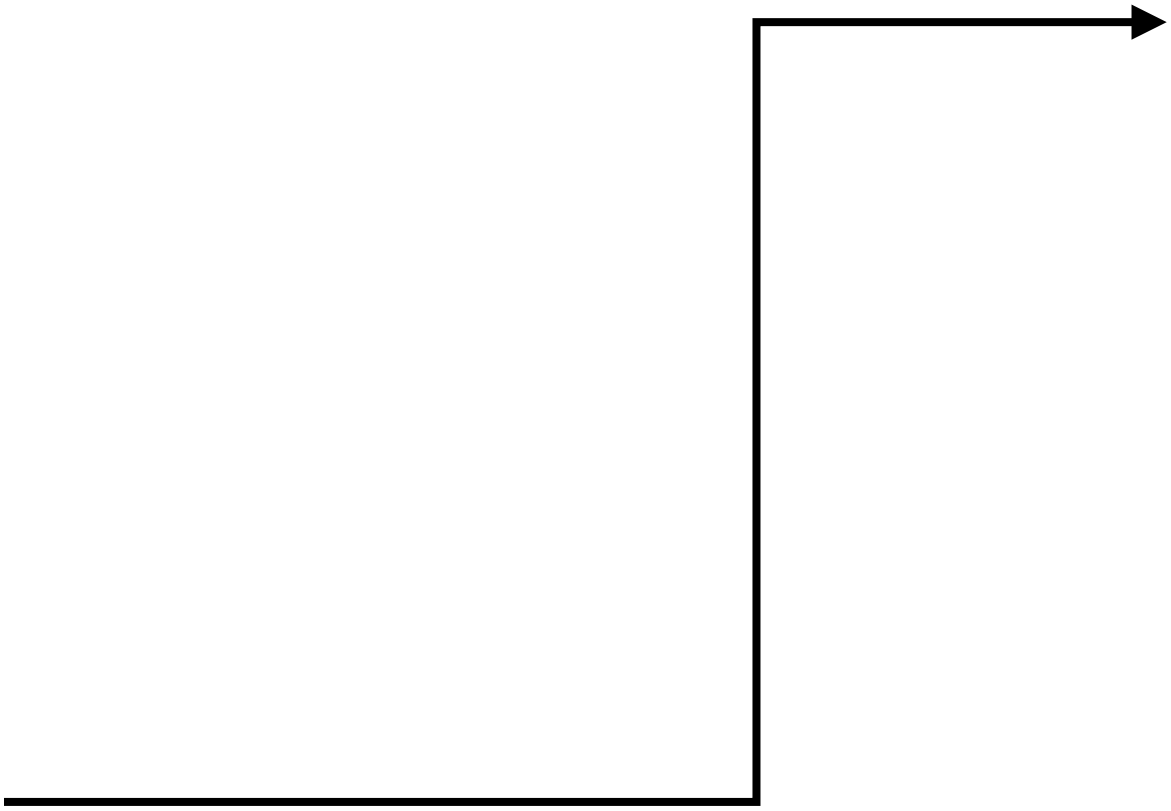


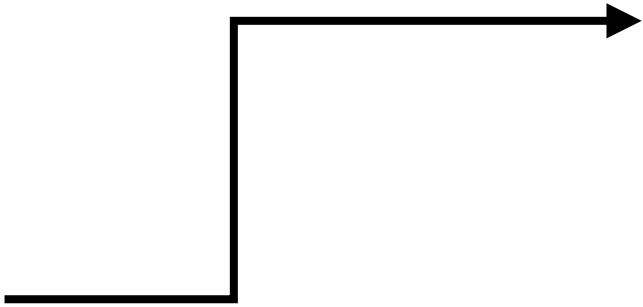


```
Canvas(  
  Modifier  
    .onGloballyPositioned { layoutCoords →  
      // Get size  
      layoutCoordinates.size  
  
      // Get bounds in window for rect  
      layoutCoordinates.boundsInWindow()  
    }  
)
```









```
val bitmap = Bitmap.createBitmap(  
    /* width = */ width,  
    /* height = */ height,  
    /* config = */ Bitmap.Config.ARGB_8888  
)  
  
// API Level 28  
PixelCopy.request(  
    /* source = */ context.getActivityWindow(),  
    /* srcRect = */ Rect(rect),  
    /* dest = */ bitmap,  
    /* listener = */ { copyResult →  
        if (copyResult == PixelCopy.SUCCESS) {  
            // share the bitmap  
            storeScreenShot(context, bitmap)  
        } else {  
            // Handle failure  
        }  
    },  
    /* listenerThread = */ Handler(getMainLooper())  
)
```

```

Canvas(
    Modifier
        .onGloballyPositioned { layoutCoords →
            // Get size
            layoutCoordinates.size

            // Get bounds in window for rect
            layoutCoordinates.boundsInWindow()
        }
)

```

```

val bitmap = Bitmap.createBitmap(
    /* width = */ width,
    /* height = */ height,
    /* config = */ Bitmap.Config.ARGB_8888
)

// API Level 28
PixelCopy.request(
    /* source = */ ...,
    /* srcRect = */ Rect(rect),
    /* dest = */ bitmap,
    /* listener = */ { copyResult →
        if (copyResult == PixelCopy.SUCCESS) {
            // use the bitmap
            ...
        } else {
            // Handle failure
        }
    },
    /* listenerThread = */ ...
)

```

Capturing Images

add a button to capture the grid
composable only

