

Are you sure you want to say
Hey Google in a room full of
Android devices? 😊

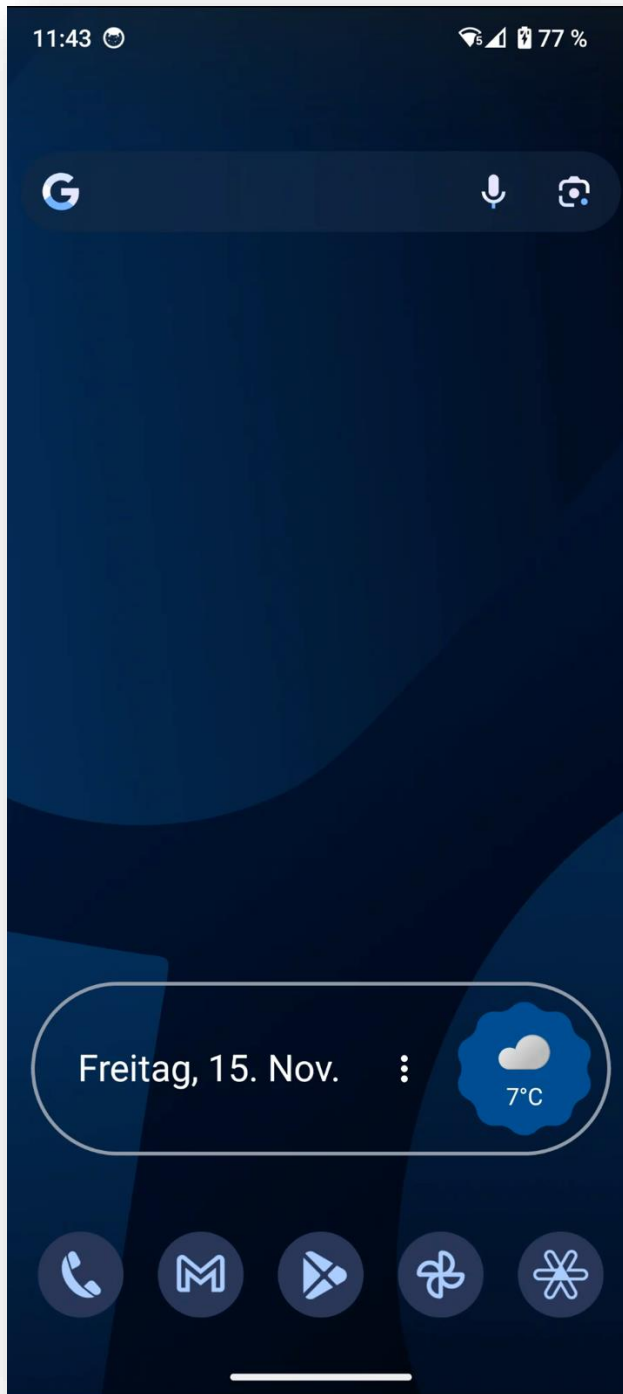
Hey Google, how can I build something like Circle to search?

Thomas Künneth



Invoke the Digital Assistant

Take a picture



Draw on screen

Ask Gemini

Appreciate the answer

Topics

- Taking pictures (sort of)
- On a ... role
- Drawing on screen
- Using Gemini



```
1  override fun onCreate(savedInstanceState: Bundle?) {
2      super.onCreate(savedInstanceState)
3      val executor = ContextCompat.getMainExecutor(this)
4      val previewView = PreviewView(this)
5      val future = ProcessCameraProvider.getInstance(this)
6      enableEdgeToEdge()
7      ...
8      setContent {
9          MaterialTheme(colorScheme = defaultColorScheme()) {
10             ...
11             val hasCameraPermission by cameraPermissionFlow.collectAsState()
12             LaunchedEffect(future) {
13                 setupCamera(
14                     future = future,
15                     lifecycleOwner = this@MainActivity,
16                     previewView = previewView,
17                     executor = executor,
18                     rotation = display.rotation
19                 ) { mainViewModel.setBitmap(it) }
20             }
21             MainScreen(uiState = uiState,
22                 previewView = previewView,
23                 hasCameraPermission = hasCameraPermission,
24                 ...
25                 finish = { finish() })
26         }
27     }
28 }
```

Required for CameraX

Make sure the preview shows something

Main UI including camera preview



```
1 private fun setupCamera(  
2     future: ListenableFuture<ProcessCameraProvider>,  
3     lifecycleOwner: LifecycleOwner,  
4     previewView: PreviewView,  
5     executor: Executor,  
6     rotation: Int,  
7     setBitmap: (Bitmap?) -> Unit  
8 ) {  
9     future.addListener({  
10         val cameraProvider = future.get()  
11         val preview = Preview.Builder().build().also {  
12             it.surfaceProvider = previewView.surfaceProvider  
13         }  
14         val imageAnalyzer = ImageAnalysis.Builder()  
15             .setBackpressureStrategy(ImageAnalysis.STRATEGY_KEEP_ONLY_LATEST).build().also {  
16             it.targetRotation = rotation  
17             it.setAnalyzer(executor) { imageProxy ->  
18                 val matrix = Matrix().also { matrix ->  
19                     matrix.postRotate(imageProxy.imageInfo.rotationDegrees.toFloat())  
20                 }  
21                 val bitmap = imageProxy.toBitmap()  
22                 val rotatedBitmap = Bitmap.createBitmap(  
23                     bitmap, 0, 0, bitmap.width, bitmap.height, matrix, true  
24                 )  
25                 bitmap.recycle()  
26                 setBitmap(rotatedBitmap)  
27                 imageProxy.close()  
28             }  
29         }  
30         try {  
31             cameraProvider.unbindAll()  
32             cameraProvider.bindToLifecycle(  
33                 lifecycleOwner, CameraSelector.DEFAULT_BACK_CAMERA, preview, imageAnalyzer  
34             )  
35         } catch (e: Exception) {  
36             Log.e(TAG, "error during unbindAll() or bindToLifecycle()", e)  
37         }  
38     }, executor)  
39 }
```

Setting up the Preview use case

Setting up the ImageAnalysis use case

Putting all pieces together

```

1 @Composable
2 fun MainScreen(
3     viewModel: MainViewModel = viewModel(),
4     uiState: UiState,
5     previewView: PreviewView,
6     hasCameraPermission: Boolean,
7     roleStatus: RoleStatus,
8     shouldShowMessage: Boolean,
9     requestRole: () -> Unit,
10    hideMessage: () -> Unit,
11    finish: () -> Unit
12 ) {
13     val scope = rememberCoroutineScope()
14     val results: @Composable () -> Unit = {
15         Results(
16             viewModel = viewModel,
17             uiState = uiState,
18             roleStatus = roleStatus,
19             shouldShowMessage = shouldShowMessage,
20             requestRole = requestRole,
21             hideMessage = hideMessage,
22             scope = scope,
23             finish = finish
24         )
25     }
26     Box(modifier = Modifier.fillMaxSize()) {
27         when (uiState) {
28             is UiState.Previewing -> {
29                 CameraPreview(previewView = previewView,
30                     hasCameraPermission = hasCameraPermission,
31                     onClick = { viewModel.startSelecting() })
32             }
33
34             is UiState.Selecting -> { CapturedImageAndDrawArea(viewModel) }
35
36             is UiState.Loading -> {
37                 AnimatedCapturedImage(viewModel)
38                 CircularProgressIndicator(modifier = Modifier.align(Alignment.Center))
39             }
40
41             is UiState.Success -> { results() }
42
43             is UiState.Error -> { results() }
44         }
45     }
46 }

```

```

1 @Composable
2 fun CameraPreview(
3     previewView: PreviewView,
4     hasCameraPermission: Boolean,
5     onClick: () -> Unit
6 ) {
7     if (hasCameraPermission) {
8         Box(modifier = Modifier.fillMaxSize(), contentAlignment = Alignment.BottomEnd) {
9             AndroidView(modifier = Modifier.fillMaxSize(), factory = { previewView })
10            FloatingActionButton(
11                onClick = onClick, modifier = Modifier.safeContentPadding()
12            ) {
13                Image(
14                    painter = painterResource(R.drawable.baseline_camera_24),
15                    contentDescription = stringResource(R.string.shutter)
16                )
17            }
18        }
19    }
20 }

```

- `ImageAnalysis` is great for repeatedly doing something with incoming images
- Not a real capture
- *`viewfinder`* processes the image only upon clicking an icon (so, why is it using `ImageAnalysis`?)

Topics

- Taking pictures (sort of)
- **On a ... role**
- Drawing on screen
- Using Gemini

What is a role?

- Unique name within the system associated with certain privileges / capabilities
- Some existing roles:
ROLE_ASSISTANT, ROLE_BROWSER,
ROLE_CALL_REDIRECTION, ROLE_CALL_SCREENING,
ROLE_DIALER, ROLE_EMERGENCY, ROLE_HOME, ROLE_NOTES,
ROLE_SMS, ROLE_WALLET
- Check availability of a role with `isRoleAvailable`

- While multiple applications may qualify for a role, only a subset can become role holders
- To qualify for a role, apps must meet certain requirements, for example defining certain components in the manifest
- Requirements can be found in `androidx.core.role.RoleManagerCompat`

Jetpack Core-Role

- Name and documentation for roles that might be available in the system
- Latest update: December 15, 2021
- Release Candidate: 1.1.0-rc01 (`androidx.core:core-role:1.1.0-rc01`)
- Package: `androidx.core.role`



Files

androidx-main

Go to file

core-performance-testing

core-performance

core-remoteviews

core-role

api

src/main/java/androidx

RoleManagerCompat.java

OWNERS

README.md

build.gradle

core-splashscreen

core-telecom

core-testing

core

androidx / core / core-role / src / main / java / androidx / core / role / RoleManagerCompat.java

Code

Blame

224 lines (212 loc) · 9.36 KB ·

Raw



↑ Top



One of the smallest Jetpack libraries

<http://www.apache.org/licenses/LICENSE-2.0>

```
*  
* Unless required by applicable law or agreed to in writing, software  
* distributed under the License is distributed on an "AS IS" BASIS,  
* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
* See the License for the specific language governing permissions and  
* limitations under the License.  
*/
```

```
/**  
 * This class contains the name and documentation for roles that might be available in the system.  
 * <p>  
 * The list of available roles might change with a system app update, so apps should not make  
 * assumption about the availability of roles. Instead, they should always check if the role is  
 * available using {@link android.app.role.RoleManager#isRoleAvailable(String)} before trying to d  
 * anything with it.  
 *  
 * @see android.app.role.RoleManager  
 */  
public final class RoleManagerCompat {
```

/**

© `androidx.core.role.RoleManagerCompat`

```
public static final String ROLE_BROWSER = "android.app.role.BROWSER"
```

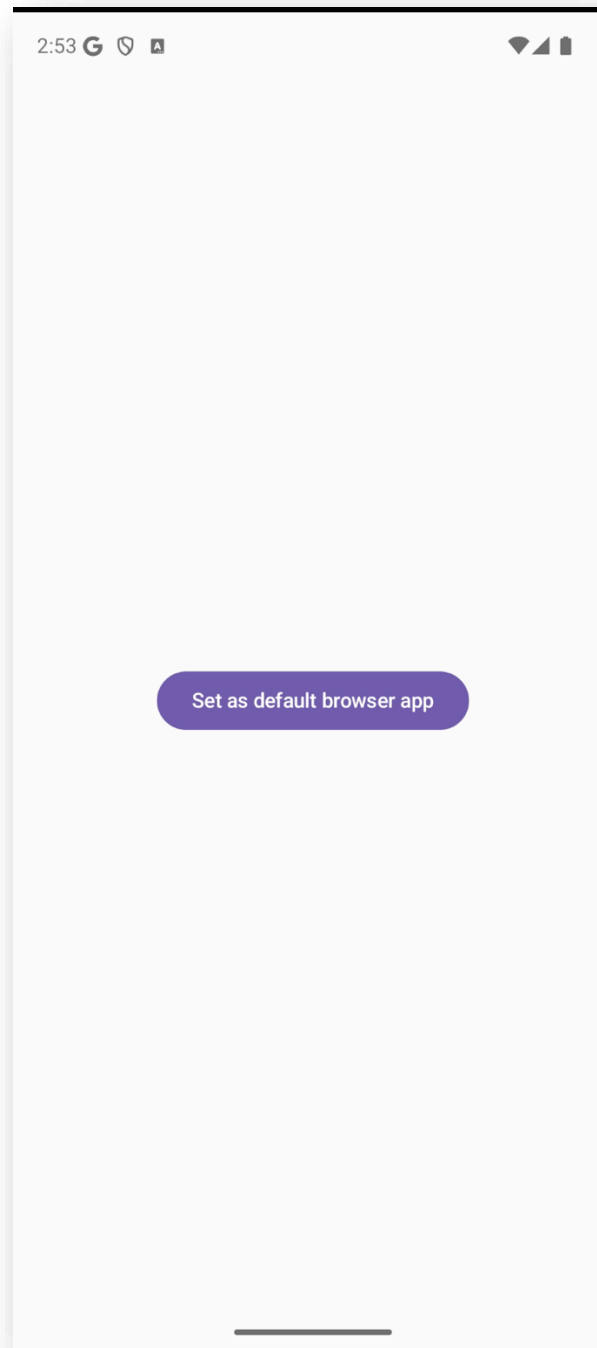
The name of the browser role.

To qualify for this role, an application needs to handle the intent to browse the Internet:

```
<activity>
  <intent-filter>
    <action android:name="android.intent.action.VIEW" />
    <category android:name="android.intent.category.BROWSABLE" />
    <category android:name="android.intent.category.DEFAULT" />
    <data android:scheme="http" />
  </intent-filter>
</activity>
```

The application will be able to handle that intent by default.

Apps that hold this role are allowed to start activities in response to notification clicks or notification action clicks when targeting `android.os.Build.VERSION_CODES.S` to give browsers time to adapt. This is temporary and browsers will be subjected to the same trampoline restrictions at some point in future releases. For more details on those restrictions see `android.app.Notification.Builder.setContentIntent(PendingIntent)` and `android.app.Notification.Action.Builder.Builder(android.graphics.drawable.Icon, CharSequence, android.app.PendingIntent)`.



[https://github.com/tkuenneth/
unclutter-your-jetpack](https://github.com/tkuenneth/unclutter-your-jetpack)



```
1 class RoleDemoActivity : ComponentActivity() {
2
3     private val roleMessage: MutableStateFlow<String> = MutableStateFlow("")
4     private val launcher =
5         registerForActivityResult(ActivityResultContracts.StartActivityForResult()) {
6             when (it.resultCode) {
7                 RESULT_OK -> roleMessage.update { getString(R.string.role_acquired) }
8                 RESULT_CANCELED -> roleMessage.update { getString(R.string.cancelled) }
9             }
10        }
11
12    override fun onCreate(savedInstanceState: Bundle?) {
13        super.onCreate(savedInstanceState)
14        enableEdgeToEdge()
15        val manager: RoleManager? = getSystemService(RoleManager::class.java)?.run {
16            if (isRoleAvailable(RoleManagerCompat.ROLE_BROWSER)) {
17                roleMessage.update {
18                    if (isRoleHeld(RoleManagerCompat.ROLE_BROWSER)) {
19                        getString(R.string.has_role)
20                    } else { "" }
21                }
22            } else {
23                roleMessage.update { getString(R.string.role_not_available) }
24            }
25            this
26        }
27        val requestRole = {
28            val intent =
29                manager?.createRequestRoleIntent(RoleManagerCompat.ROLE_BROWSER) ?: Intent()
30            launcher.launch(intent)
31        }
32        setContent { ... }
33    }
34 }
```

Handle the result of role requests

- Get a RoleManager instance
- Check role availability (isRoleAvailable)
- Check if role is held (isRoleHeld)

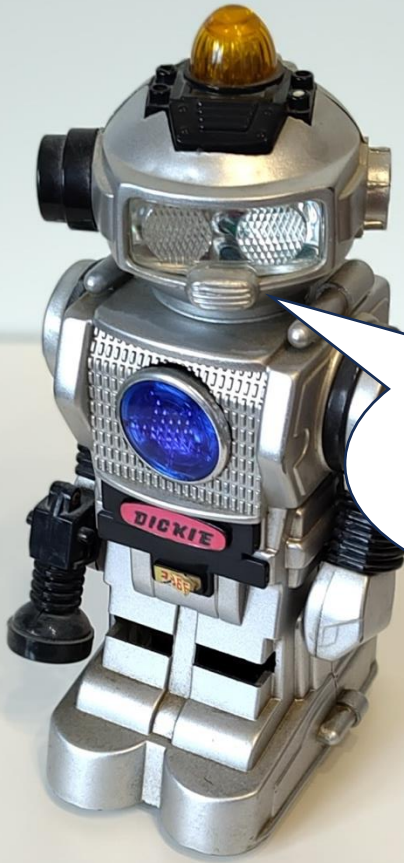
Request the role



```
1 setContent {
2     val scope = rememberCoroutineScope()
3     val message by roleMessage.collectAsState()
4     Box(
5         contentAlignment = Alignment.Center,
6         modifier = Modifier
7             .fillMaxSize()
8             .safeContentPadding()
9     ) {
10         Button(onClick = {
11             scope.launch { requestRole() }
12         }) {
13             Text(stringResource(id = R.string.acquire_role))
14         }
15         Text(
16             text = message,
17             style = MaterialTheme.typography.displayMedium,
18             textAlign = TextAlign.Center,
19             color = MaterialTheme.colorScheme.primary,
20             modifier = Modifier
21                 .align(Alignment.BottomCenter)
22                 .safeContentPadding()
23         )
24     }
25 }
```

Request the role

Update upon state changes



Why did you explain this using ROLE_BROWSER?

Browser window showing the Android Code Search page for the commit `3512ea1`. The page displays the commit message "Don't allow requesting the assistant role." and the commit details, including the bug number `130735926` and the change ID `166b4e203f2922645e3a357ca0f628faefadfe344`.

The commit message is: "Don't allow requesting the assistant role." By adding a requestable attribute for roles and set it to false for assistant.

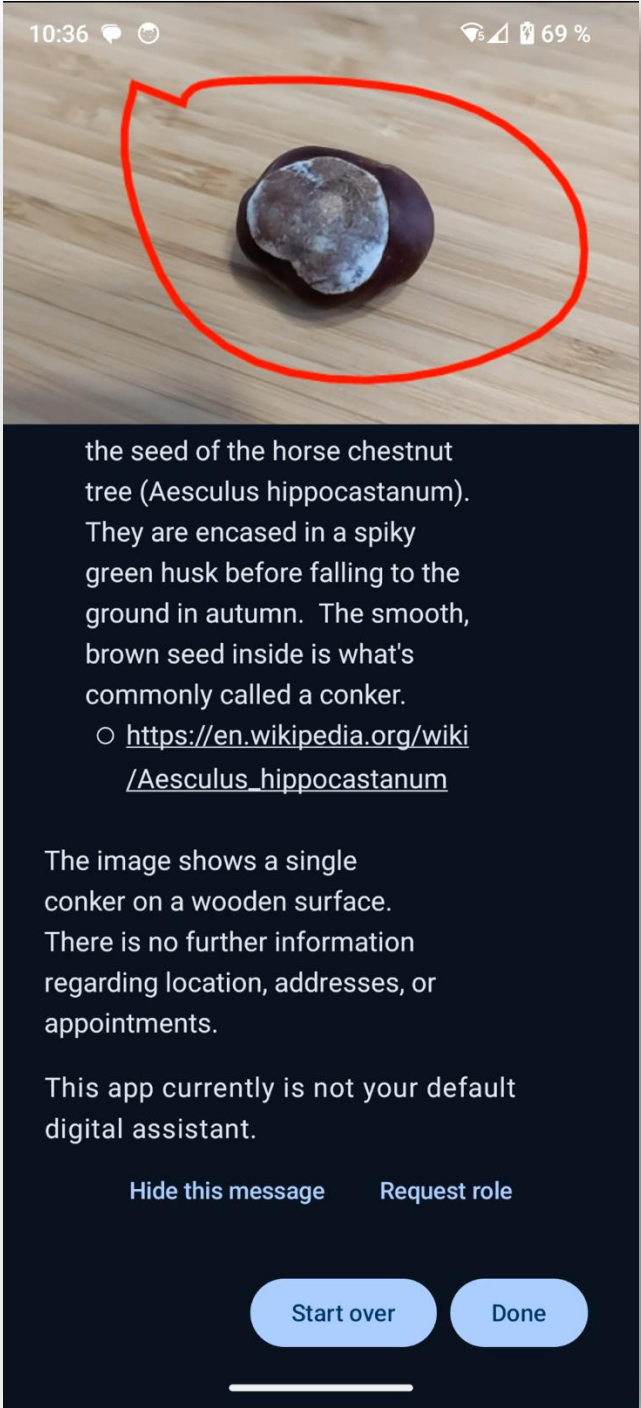
Bug: `130735926`
Test: `atest RoleManagerTest#requestAssistantRoleThenDeniedAutomatically`
Change-Id: `166b4e203f2922645e3a357ca0f628faefadfe344`

Author: zhanghai@google.com (author) Committed on 2019-04-18 7:40 AM

Parent: `d920cbf`

Files (4):

Type	Path	Change Summary
File	<code>PermissionController/res/xml/roles.xml</code>	<code>-1, 1, +0</code>
File	<code>PermissionController/src/com/android/packageinstaller/role/model/Role.java</code>	<code>-0, 3, +0</code>
File	<code>PermissionController/src/com/android/packageinstaller/role/model/Roles.java</code>	<code>-0, 1, +0</code>
File	<code>PermissionController/src/com/android/packageinstaller/role/ui/RequestRoleActivity.java</code>	<code>-0, 1, +0</code>





```
1 enum class RoleStatus {
2     NOT_HELD, HELD, UNAVAILABLE
3 }
4
5 fun getRoleStatus(held: Boolean): RoleStatus = if (held) RoleStatus.HELD else RoleStatus.NOT_HELD
6
7 private val roleFlow: MutableStateFlow<RoleStatus> = MutableStateFlow(RoleStatus.NOT_HELD)
8 private val roleLauncher =
9     registerForActivityResult(ActivityResultContracts.StartActivityForResult()) { _ ->
10         roleFlow.update { getRoleStatus(manager.isRoleHeld(RoleManagerCompat.ROLE_ASSISTANT)) }
11     }
12
13 manager.run {
14     if (isRoleAvailable(RoleManagerCompat.ROLE_ASSISTANT)) {
15         roleFlow.update {
16             getRoleStatus(isRoleHeld(RoleManagerCompat.ROLE_ASSISTANT))
17         }
18     } else roleFlow.update { RoleStatus.UNAVAILABLE }
19 }
20 val requestRole = {
21     val intent = Intent(Settings.ACTION_VOICE_INPUT_SETTINGS)
22     roleLauncher.launch(intent)
23 }
```

Update the role status

Initial setup

Jump into system settings

```
public final class RoleManagerCompat {
```

The name of the assistant role.

To qualify for this role, an application needs to either implement `android.service.voice.VoiceInteractionService` or handle `android.content.Intent.ACTION_ASSIST`. The application will be able to access call log and SMS for its functionality.

See Also: `android.service.voice.VoiceInteractionService`,
`android.content.Intent.ACTION_ASSIST`

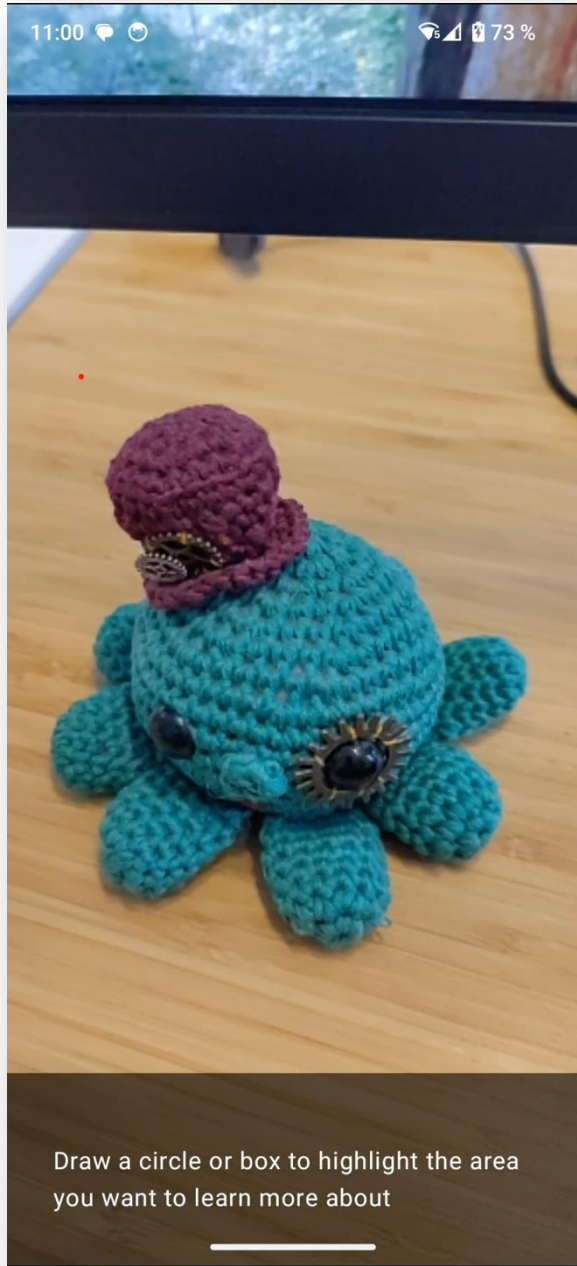
```
public static final String ROLE_ASSISTANT = "android.app.role.ASSISTANT";
```

```
1 <activity
2     android:name=".MainActivity"
3     android:exported="true">
4     <intent-filter>
5         <action android:name="android.intent.action.MAIN" />
6         <category android:name="android.intent.category.LAUNCHER" />
7     </intent-filter>
8     <intent-filter>
9         <action android:name="android.intent.action.ASSIST" />
10        <category android:name="android.intent.category.DEFAULT" />
11    </intent-filter>
12 </activity>
```

- Since the last Jetpack library update, the framework class `RoleManager` received additions, for example `ROLE_NOTES` (API level 34)
- `RoleManagerCompat` lacks factory functions that ensure a consistent behavior on older Android versions (`RoleManager` was added with API level 29)
- In its current form Core-Role looks more than some sort of documentation than a component

Topics

- Taking pictures (sort of)
- On a ... role
- **Drawing on screen**
- Using Gemini





```
1 const val STROKE_WIDTH = 12F
2 val DRAWING_COLOR = Color.Red
3
4 @Composable
5 fun DrawingArea(drawComplete: (IntSize, List<Offset>) -> Unit) {
6     val points = remember { mutableStateListOf<Offset>() }
7     Canvas(modifier = Modifier.fillMaxSize())
8         .pointerInput(Unit) {
9             awaitPointerEventScope {
10                 while (true) {
11                     val event = awaitPointerEvent()
12                     val touch = event.changes.first()
13                     points.add(touch.position)
14                     if (!touch.pressed) {
15                         if (points.size > 2) {
16                             drawComplete(size, points.toList())
17                         }
18                         points.clear()
19                     }
20                 }
21             }
22             if (points.size > 2) {
23                 drawPath(
24                     path = Path().apply {
25                         moveTo(points[0].x, points[0].y)
26                         for (i in 1..points.lastIndex) {
27                             lineTo(points[i].x, points[i].y)
28                         }
29                         close()
30                     }, color = DRAWING_COLOR, style = Stroke(width = STROKE_WIDTH)
31                 )
32             } else {
33                 points.forEach { point ->
34                     drawCircle(
35                         color = DRAWING_COLOR, center = point, radius = STROKE_WIDTH / 2F
36                     )
37                 }
38             }
39 }
```

Handle pointer events

Draw a closed path

Draw individual circles



```
1 @Composable
2 fun CapturedImageAndDrawArea(viewModel: MainViewModel) {
3     Box(modifier = Modifier.fillMaxSize()) {
4         CapturedImage(viewModel)
5         Info()
6         DrawingArea { size, offsets ->
7             viewModel.capturedImage.value?.let { bitmap ->
8                 val xRatio = bitmap.width.toFloat() / size.width.toFloat()
9                 val yRatio = bitmap.height.toFloat() / size.height.toFloat()
10                val scaledOffsets = offsets.map { point ->
11                    PointF(point.x * xRatio, point.y * yRatio)
12                }
13                val canvas = Canvas(bitmap)
14                val path = android.graphics.Path()
15                if (scaledOffsets.isNotEmpty()) {
16                    path.moveTo(scaledOffsets[0].x, scaledOffsets[0].y)
17                    for (i in 1 until scaledOffsets.size) {
18                        path.lineTo(scaledOffsets[i].x, scaledOffsets[i].y)
19                    }
20                    path.close()
21                }
22                canvas.drawPath(path, Paint().apply {
23                    style = Paint.Style.STROKE
24                    strokeWidth = STROKE_WIDTH / 2F
25                    color = DRAWING_COLOR.toArgb()
26                })
27                viewModel.askGemini(bitmap)
28            }
29        }
30    }
31 }
```

Convert points to match image size

Create a Canvas based on a Bitmap

Create a Path from the scaled points

Draw the Path onto the Canvas

Ask Gemini

Topics

- Taking pictures (sort of)
- On a ... role
- Drawing on screen
- **Using Gemini**

```
1 [versions]
2 generativeai = "0.9.0"
3
4 [libraries]
5 generativeai = { group = "com.google.ai.client.generativeai", name = "generativeai", version.ref = "generativeai" }
```

```
1 private const val modelName = "gemini-1.5-pro"
2
3 class MainViewModel : ViewModel() {
4
5     private val _bitmap: MutableStateFlow<Bitmap?> = MutableStateFlow(null)
6     private val bitmap = _bitmap.asStateFlow()
7
8     private val generativeModel = GenerativeModel(
9         modelName = modelName, apiKey = BuildConfig.apiKey
10    )
11
12     fun askGemini(bitmap: Bitmap) {
13         sendPrompt(bitmap = bitmap)
14     }
15     ...
}
```

```

1 private fun sendPrompt(bitmap: Bitmap) {
2     _uiState.update { UiState.Loading }
3     viewModelScope.launch(Dispatchers.IO) {
4         try {
5             val actions = mutableListOf<Pair<Action, String>>()
6             val description = generativeModel.generateContent(content {
7                 image(bitmap)
8                 text(prompt_01)
9             }).text ?: ""
10            ...

```

```

1 private val prompt_01 = """
2     Describe what is contained inside the thick red line inside the image.
3     Give a short description, followed by a bullet point list with all important details.
4     Add web links with additional information for each bullet point items when available.
5     Choose Wikipedia if possible.
6     If there are details related to appointments, locations, addresses,
7     mention these explicitly
8     """.trimIndent()

```



```

1 private val prompt_02 = """
2     Does the following text contain information that looks like
3     a business card? Please answer only with yes or no.
4     Here is the text: %s
5 """.trimIndent()
6
7 private val prompt_03 = """
8     Please create a data structure in VCARD format.
9     Do not add any explanations. Instead, make sure that your answer only
10    contains the VCARD data structure, nothing else.
11    Use the information that follows after the colon: %s
12 """.trimIndent()

1 // Second step: Does the description contain appointment info?
2 with(generativeModel.generateContent(content {
3     text(String.format(prompt_02, description))
4 })) {
5     if (text?.toLowerCase(Locale.current)?.contains("yes") == true) {
6         with(generativeModel.generateContent(content {
7             text(String.format(prompt_03, description))
8         }).text) {
9             val data = this?.replace("`vcard", "").replace("`", "") ?: ""
10            if (data.isNotEmpty()) {
11                actions.add(Pair(Action.VCARD, data))
12            }
13        }
14    }
15 }

```

- Default format of Gemini answers is Markdown
- Can be changed, but is nice if the answers are shown on screen
- Cool Open-Source library by Jeziel Lago: `compose-markdown`
(<https://github.com/jeziellago/compose-markdown>)



Pretty cool, but ...

- Why not taking advantage of the continuous image stream?
- How about using on-device LLMs?



[https://github.com/tkuenneth/
viewfinder](https://github.com/tkuenneth/viewfinder)



[https://github.com/tkuenneth/
unclutter-your-jetpack](https://github.com/tkuenneth/unclutter-your-jetpack)

Thank you

 @tkuenneth

 @tkuenneth

 @tkuenneth.bsky.social