

Safe Zone

Was ist die Safe Zone?

- Was ist die Safe Zone?
 - Safe Zone auf dem gesamten Canvas
 - Safe Zone pro Panel
 - Safe Zone pro Element

Spätestens mit der Ankündigung des iPhone X wurden sogenannte "edge-to-edge" oder randlose Displays zu einem weitverbreiteten Design auf der Mobile-Plattform. Damit die UI nicht von Notches oder Flächen für Gesten (z.B. iPhone Home-Geste) überlagert wird, gibt es die Safe Zone. **Innerhalb der Safe Zone ist garantiert, dass Elemente bedienbar und vollständig sichtbar sind.**

Hier mal ein paar Beispiele aus dem Device Simulator, wo die Safe Zone in gelb drübergelegt ist:



Abbildung 1: iPhone 12 mit symmetrischer Safe Zone



Abbildung 2: iPad 12.9 mit kleiner Safe Zone für den Home Indikator



Abbildung 3: Samsung Galaxy S10e mit asymmetrischer Safe Zone

Auffällig ist, dass Apple seine Safe Zone idR. symmetrisch hält (vgl. Abbildung 1), obwohl dies nicht wirklich notwendig ist. Bei Android wurde sich entsprechend für die asymmetrische Safe Zone entschieden (Abbildung 3). Des Weiteren hält Apple stets Platz für den Home Indikator frei, der sich am unteren Bildschirmrand befindet, selbst wenn es ansonsten keine Safe Zone gibt (Abbildung 2).

Veraltete Apps, die Notches bzw. Safe Zone nicht korrekt unterstützen sehen inzwischen so aus:



Abbildung 4a: Die Safe Zone eines Google Pixel 5



Abbildung 4b: Das Spiel wird entlang der Safe Zone abgeschnitten. Der Rest des Bildschirms bleibt schwarz.

Beim Aufbau und Design der eigenen UI sollte die Safe Zone unbedingt mit einbezogen werden. Damit verhindert man das,

- Elemente der UI durch Rundungen in den Ecken des Displays abgeschnitten werden,
- Bedienelemente von der Notch verschluckt werden oder sonst wie nicht bedienbar sind,
- beim Interagieren ungewollt Systemgesten angesteuert werden.

Ferner noch lohnt es sich über die Safe Zone hinaus Mindestabstände zum Rand zu etablieren, da viele Displays runde Ecken haben. Man kann in Abbildung 4b bereits die Möglichkeit erahnen, dass in den Ecken liegende Buttons von den Rundungen abgeschnitten werden. Für das Design bedeutet die Safe Zone vor allem erst einmal weniger Platz für so gut wie alles. Beim Bau der UI stellt sich die Frage, wie man sich an die Safe Zone anpasst.

Ich habe hier ein Asset aus dem Store, mit samt Artikel zur Erklärung parat. Zusammengefasst erhält man über die Screen-Klasse ein Rect, welches die Maße der Safe Zone darstellt. Dieses Rect kann man dann mit den Maßen der vollen Auflösung verrechnen und so Werte für `anchorMin` bzw. `anchorMax` des anzupassenden `RectTransform`s erhalten, um dieses auf die Dimensionen der Safe Zone zu setzen. Zur Erinnerung: Die Anker Werte sind zwischen 0 und 1, sie beschreiben wie viel des Rects auf einer Achse genutzt wird. Man errechnet also, wie viel die Safe Zone anteilig vom gesamten Bildschirm ausmacht.

Final stellt sich damit noch die Frage, auf welcher Ebene ein Safe Zone Script liegen sollte:

- Canvas
- Panel
- Element

Die Antwort darauf hängt ganz vom Design ab – vor allem, ob das/die Panel/s einen fullscreen Hintergrund haben.

Safe Zone auf dem gesamten Canvas



Abbildung 5a: Canvas in Größe der Safe Zone

Stumpf ist Trumpf – wir passen einfach den gesamten Canvas auf die Safe Zone an. In Abbildung 5a erhalten wir jetzt ein ähnliches Bild, wie zuvor in Abbildung 4. Es ergeben sich also folgende Probleme:

1. Was ist mit fullscreen Hintergründen bzw. Panels?
2. Was ist, wenn ich Elemente bündig zum Displayrand platzieren muss?



Abbildung 5b: Hintergrund mit negativem Padding nach links – Simulator View



Abbildung 5c: Padding nach links – Scene View

In beiden Szenarien muss man die Safe Zone Änderung quasi wieder zurückrechnen. Bei Hintergründen von Panels könnte man einfach ein großzügiges Maß an Padding hinzufügen (Abbildungen 5b und c), das ist jedoch ein wenig hacky und kann in Zukunft nicht mehr ausreichen, wenn wieder andere Display Dimensionen auf den Markt kommen. Am besten platziert man also eine Safe Zone auf Canvas Ebene, wenn man weiß, das Panels darunter sowieso keine Fullscreen-Grafiken oder Elemente am Displayrand haben werden.

Safe Zone pro Panel



Abbildung 6a: Auf die Safe Zone angepasstes Panel mit Buttons



Abbildung 6b: Angepasstes Panel - Simulator View

Eine flexiblere Lösung kann es sein pro Panel zu entscheiden, ob dieses eine Safe Zone benötigt oder nicht. Auf diese Art und Weise können Safe Zone-konforme und nicht-konforme Panel auf dem gleichen Canvas existieren, was Hierarchie und Layering vereinfacht. Für die Abbildungen 6 a und b habe ich den Hintergrund in ein eigenes Panel umgewandelt, was sich nicht an die Safe Zone hält. Die Buttons hingegen sind auf einem Safe Zone konformen Panel.

Man kann bereits erkennen, warum eine Asymmetrisches Safe Zone Probleme bereiten kann: Die Buttons sind nun nicht mehr korrekt zentriert. Design oder Safe Zone Script müssen sich an diesem Umstand anpassen. Es könnte beispielsweise im Script festgelegt werden, dass die Safe Zone immer symmetrisch ist.

Safe Zone pro Element

Final könnte man innerhalb eines Panels in Safe Zone-konforme Elemente und nicht-konforme Elemente unterteilen. Eine beispielhafte Hierarchie wäre Panel Hintergrund Inhalt mit Safe Zone. Das dritte Beispiel aus dem Artikel verdeutlicht die Notwendigkeit einer solchen Aufteilung:



Abbildung 7a: Panel mit Safe Zone auf der obersten Ebene



Abbildung 7a: Panel mit Safe Zone unabhängigen Streifen

Die horizontalen blauen Streifen sollen fullscreen sein und würden mit der "Safe Zone pro Panel"-Variante, entsprechend abgeschnitten werden. Währenddessen müssen sich die obere und untere Zeile des Panels bzw. die Zeile mit Score und die mit Buttons für Leaderboard, Neustart und Einstellungen, an die Safe Zone halten. Jede Zeile ist also ein Element, mit oder ohne Safe Zone, das sich unter dem Panel befindet. Damit sich die blauen Balken über die gesamte Breite erstrecken können, verwenden wir drei mal die Eingangs beschriebene Struktur Element (zuvor Panel; jetzt Element unterm Panel) Balken Inhalt mit Safe Zone.