

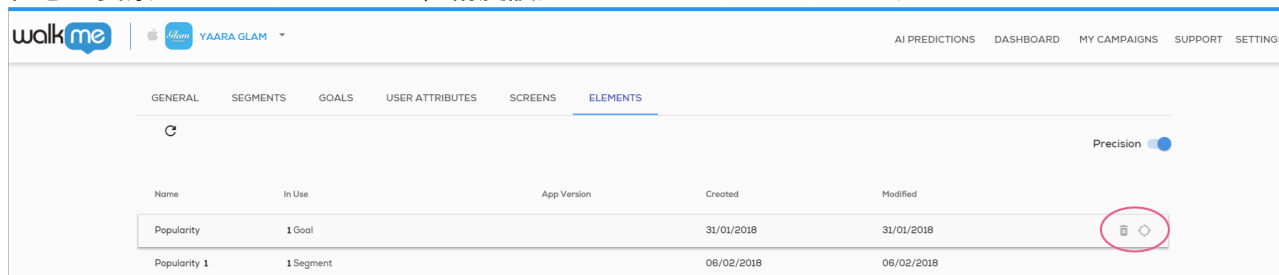
モバイル：エレメント精度設定

WalkMe Mobile SDKバージョン1.7.0から始め、アプリの要素の精度設定を微調整して、アプリで特定の要素を識別するWalkMeの機能を向上します。アルゴリズムの違いによりOSネイティブ、ハイブリッドアプリ間で使用可能な調整が異なります。

ベストプラクティスとして、エンドユーザーのタップリージョンを最大限に活用するために、要素選択を可能な限り最大のリージョンに拡大することを推奨します。このように、要素はキャプチャで可能な限りの詳細が含まれています。

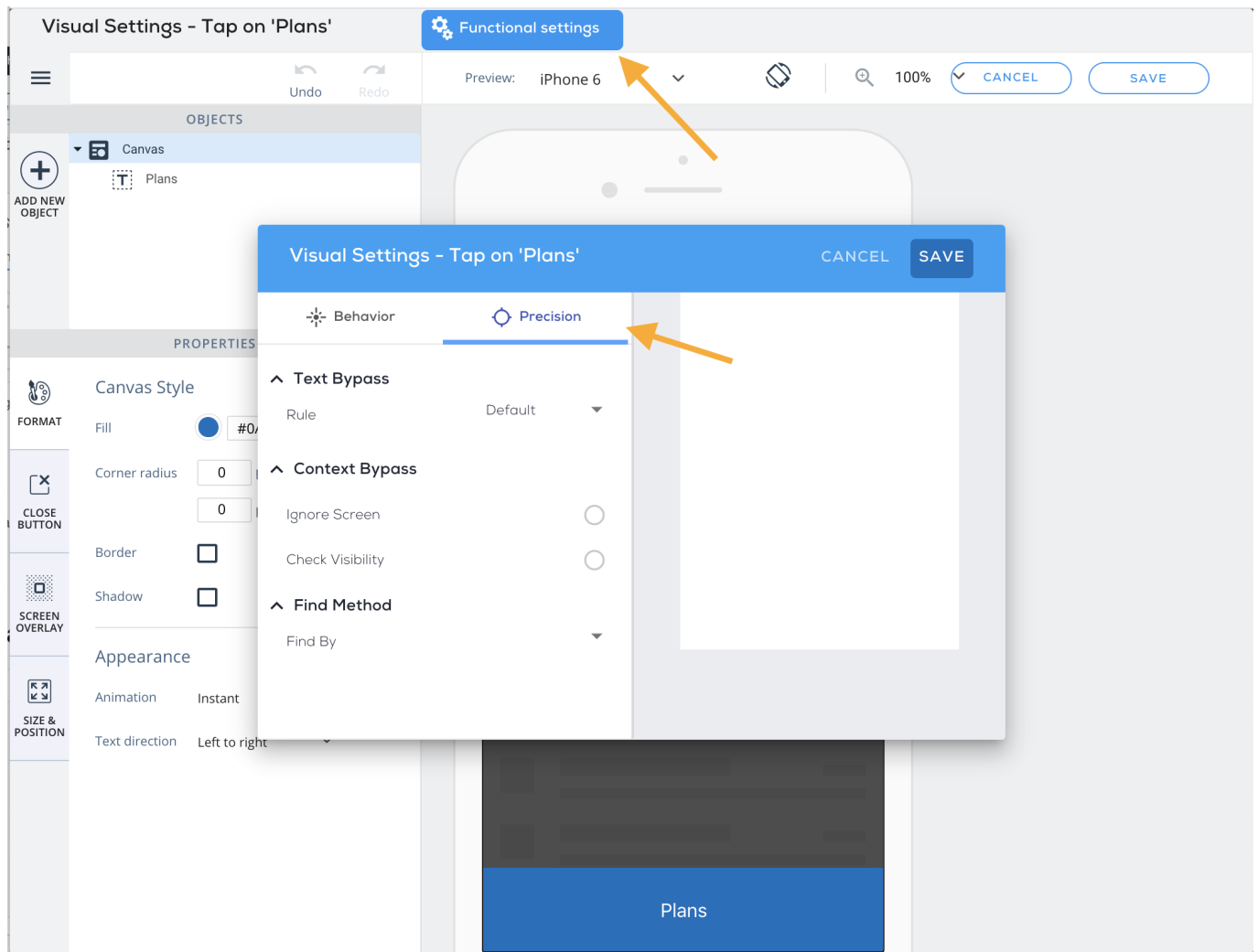
精度設定を追加する手順

1. Mobile Consoleで、**設定>アプリの設定>要素**に移動します。
2. 任意の要素にカーソルをあわせ、精度設定アイコンをクリックします：



3. 希望の精度設定を選択します。
4. 保存をクリックします。

以下のすべての情報は、スマートウォークスルーステップにも適用されます：



精度設定マトリックス

iOSネイティブ	iOSハイブリッド	Androidネイティブ	Androidハイブリッド
デフォルト テキストを無視する テキストの一致 画面を無視する アクセシビリティIDを 無視する 表示を確認する 階層[xpath]で検索す る タイプで探す	デフォルト テキストを無視する テキストで識別する 画面を無視する 階層[xpath]で検索す る タイプで探す	デフォルト テキストを無視する テキストの一致 画面を無視する クラス名を無視する IDを無視する アクセシビリティIDを 無視する 階層[xpath]で検索す る タイプで探す	デフォルト テキストを無視する テキストで識別する 画面を無視する クラス名を無視する IDを無視する 階層[xpath]で検索す る タイプで探す

精度変更-タイミング

精度に対するすべての更新が即時であるとは限りません。これは、サーバー側でcampaign自体に変更を移行する方法によるものです。ここでは、精度を更新する2つの方法に対してコンテンツを再生する4つの方法を簡潔に紹介しています：

	プレビュー	シミュレート	キャプチャーされたアイテム	公開済み
SWTステップ	自動化	〜1〜2分	該当なし	〜1〜2分
キャプチャされた要素	〜1〜2分	〜1〜2分	自動化	〜1〜2分

精度設定定義

デフォルト

WalkMeアルゴリズムは、要素の存在を評価するために、要素の様々な側面を考慮します。完全な要素の説明は、コンソールでキャプチャした要素をクリックすることで確認できます。

テキストを無視する

キャプチャにテキストが含まれている場合、キャプチャされた要素を確認しようとするWalkMeアルゴリズムはそのテキストを無視します。テキストを変更する要素（たとえば、ボタン）が存在する場合、または動的テキストでテーブル/リストをキャプチャしようとしている場合。キャプチャされたコードにテキスト値がない場合、テキストを無視する効果はありません`"text": "[here]"`

```
{
  "uiKitType": "UITabBarButton",
  "frame": "{227, 730}, {71, 48}",
  "isInteractable": true,
  "screen": "Glip.TabBarController\\Glip.RCUIPopGestureNavigationController\\Glip.GlipUIRecentChatViewController\\Glip.GlipUIRecentChatAllTableViewCellViewController",
  "isInput": 0,
  "windowBounds": "{0, 0}, {375, 812}",
  "type": "UITabBarButton",
  "propertyName": "",
  "xpath": "UIWindow()\\UILayoutContainerView(0)\\UITabBar(1)\\UITabBarButton(4)",
  "accessabilityInfo": {
    "accessabilityLabel": "",
    "accessabilityHint": "",
    "accessabilityValue": "",
    "text": "Contacts",
    "containerVC": "Glip.TabBarController",
    "imageFingerprint": 7551607122347454464,
    "referenceVC": "Glip.RCUIPopGestureNavigationController",
    "sdkVersion": "2.3.0",
    "uiControlArray": [
      {
        "target": "UITabBar",
        "actions": [
          "_buttonDown:",
          "_buttonUp:",
          "_buttonCancel:",
          "_sendAction:withEvent:"
        ]
      }
    ]
  }
}
```

テキストの一致

キャプチャされたテキストが存在する場合（上記の画像で見たように）、テキストの一致はアルゴリズムを狭め、テキストに焦点を当てることができます。Regexは使用できますが、必要はありません。注意：これはネイティブ要素（ハイブリッドではない）にのみ適用されます。

テキストで識別する

ハイブリッド要素でキャプチャされたテキストに焦点を当てることができます。注意：キャプチャされたテキストが複数の文字列に区切られている場合WalkMeは最初の文字列でのみ識別されます。

画面を無視する

画面の値の中には、動的であるか、または十分に説明的ではないものがあります。この設定により、これらの場合の画面の値を無視することができます。注意：ウォークスルーの場合、最初のステップのみが画面を考慮しているため、最初のステップの画面の値が動的である場合は、画面を無視するを追加する必要があります。その後のすべてのステップは、画面の値を考慮しません。

クラス名を無視する

Androidでは、要素にクラス名を持つことができます。これは、キャプチャで「class_name」として書かれています。選択しようとしている要素に動的クラスがある場合、クラス名を無視することが重要です。注意：これはAndroidにのみ適用されます

```
{
  "screen": "HomeActivity#HomeMeetingsPageFragment",
  "activity": "com.glip.ui.home.HomeActivity",
  "isInput": false,
  "indexOfListInPath": 0,
  "viewPositionInList": -1,
  "text": "Message",
  "isOnDialog": false,
  "sdk_ver": "2.5.3",
  "version": 2,
  "id": -1,
  "accessibilityId": "Message",
  "class_name": "com.google.android.material.bottomnavigation.BottomNavigationView",
  "index_on_parent": 0,
  "xpath": {
    "nodes": [
      {
        "id": -1,
        "className": "android.widget.LinearLayout",
        "index": 0
      },
      {
        "id": -1,
        "className": "android.widget.FrameLayout",
        "index": 0
      },
      {
        "id": 2131296368,
        "className": "androidx.appcompat.widget.FitWindowsFrameLayout",
        "index": 0
      },
      {
        "id": 16908290,
        "className": "androidx.appcompat.widget.ContentFrameLayout",
        "index": 0
      }
    ]
  }
}
```

IDを無視する

class_nameと同様に、IDはAndroid独自のものです。すべての要素キャプチャにIDがあるわけではありませんがIDが一意でない場合、トラブルシューティング中にそれを無視するのに便利かもしれません。

```
{
  "screen": "HomeActivity#HomeGroupsPageFragment",
  "activity": "com.glip.ui.home.HomeActivity",
  "isInput": false,
  "indexOfListInPath": 0,
  "viewPositionInList": -1,
  "text": "Message",
  "isOnDialog": false,
  "sdk_ver": "2.5.3",
  "version": 2,
  "id": -1,
  "accessibilityId": "Message",
  "class_name": "com.google.android.material.bottomnavigation.BottomNavigationView",
  "index_on_parent": 0,
  "xpath": {
    "nodes": [
      {
        "id": -1,
        "className": "android.widget.LinearLayout",
        "index": 0
      }
    ]
  }
}
```

アクセシビリティIDを無視する

WalkMeは、この記事で説明されているように、さまざまな要因に基づいて要素に焦点を当て、要素を識別することができます。アプリ開発者の中には、アクセシビリティIDを使用して要素をコード化している方がいます – 画像、テキスト、言語などの他の動的特性に関係なく、一意の識別子を使用します。要素でアクセシビリティIDが利用可能である場合WalkMeはそれに焦点を当てます。この設定を使用して、必要性が生じた場合にアクセシビリティIDを無視できます。

階層[xpath]で検索する

デフォルトではWalkMeはアルゴリズムの階層に重点を置きます。階層は、選択された要素の「パス」を見ます。たとえば、ページ>リスト>行1>セル1 コンソールでは、これは[x-path]です。iOS 13以上と12以下間で異なる階層がある可能性がある既知の問題があります。可能であればiOS 13とiOS 12を使用したデバイスをテストします。人々がiOS 13+に移行するにつれて、これは一般的になります。

タイプで探す

Webと比較すると、これは「要素クラス」です。たとえば、画面にある各リストアイテムまたは各保存ボタンは、おそらく同じクラスを持っています。これがコンソールの[タイプ]です。タイプを使用するための最も一般的なユースケースは、上記で説明したiOS12/iOS13階層の問題を排除することです。

表示を確認します(iOS SWTステップのみに適用されます)

アプリ画面は、階層と一部の要素で開発されることがありますが、上部の階層の下に非表示となっている間、技術的に存在します。WalkMeは、非表示の要素を指すステップにつながるこれらの要素を確認することができます。解決するには、この設定を有効にしてWalkMeが本当に表示されているときにこれらの要素を確認するだけです。