Version: 2.6.0 – 2.12.0

GestureDetector

`GestureDetector` is the main component of the RNGH2. It is responsible for creating and updating native gesture handlers based on the config of provided gesture. The most significant difference between it and old gesture handlers is that the `GestureDetector` can recognize more than one gesture at the time thanks to gesture composition. Keep in mind that `GestureDetector` is not compatible with the Animated API, nor with Reanimated 1.

CAUTION

Gesture Detector will use first native view in its subtree to recognize gestures, however if this view is used only to group its children it may get automatically collapsed. Consider this example:

```
export default function Example() {
  const tap = Gesture.Tap().onStart(() => {
    console.log('tap');
  });

  return (
    <GestureDetector gesture={tap}>
      <FunctionalComponent>
        <View style={styles.box} />
      </FunctionalComponent>
    </GestureDetector>
  );
}

function FunctionalComponent(props) {
  return <View collapsable={false}>{props.children}</View>;
}
```

If we were to remove the collapsable prop from the View, the gesture would stop working because it would be attached to a view that is not present in the view hierarchy. Gesture Detector adds this prop automatically to its direct child but it's impossible to do automatically for more complex view trees.

Properties

`gesture`

A gesture object containing the configuration and callbacks. Can be any of the base gestures (`Tap`, `Pan`, `LongPress`, `Fling`, `Pinch`, `Rotation`, `ForceTouch`) or any `ComposedGesture` (`Race`, `Simultaneous`, `Exclusive`).

> INFO
>
> GestureDetector will decide whether to use Reanimated to process provided gestures based on callbacks they have. If any of the callbacks is a worklet, tools provided by the Reanimated will be utilized bringing ability to handle gestures synchrously.
>
> Starting with Reanimated-2.3.0-beta.4 Gesture Handler will provide a StateManager in the touch events that allows for managing the state of the gesture.

`userSelect` (web only)

This parameter allows to specify which `userSelect` property should be applied to underlying view. Possible values are `"none" | "auto" | "text"`. Defaults to `"none"`. **Available since version 2.8.0**