import { Dimensions, Platform, Easing, Animated } from 'react-native'

const { width } = Dimensions.get('window')

const IosTransition = (

  index: number,

  position: {

    interpolate: {

      (arg0: { inputRange: any[]; outputRange: number[] }): void

      (arg0: { inputRange: any[]; outputRange: number[] }): void

    }

  }

) => {

  const inputRange = [index - 1, index, index + 0.99, index + 1]

  const outputRange = [width, 0, -10, -10]

  const translateY = 0

  const translateX = position.interpolate({

    inputRange,

    outputRange,

  })

  const opacity = position.interpolate({

    inputRange,

    outputRange: [0, 1, 1, 0],

  })

  return {

    opacity,

    transform: [{ translateX }, { translateY }],

  }

}

const DroidTransition = (

  index: number,

  position: {

    interpolate: {

      (arg0: { inputRange: any[]; outputRange: number[] }): void

      (arg0: { inputRange: any[]; outputRange: number[] }): void

    }

  }

) => {

  const inputRange = [index - 1, index, index + 0.99, index + 1]

  const opacity = position.interpolate({

    inputRange,

    outputRange: [0, 1, 1, 0],

  })

  const translateX = 0

  const translateY = position.interpolate({

    inputRange,

    outputRange: [50, 0, 0, 0],

  })

  return {

    opacity,

    transform: [{ translateX }, { translateY }],

  }

}

const transitionConfig = () => {

  return {

    transitionSpec: {

      useNativeDriver: true,

      easing: Easing.out(Easing.poly(4)),

      timing: Animated.timing,

    },

    screenInterpolator: (sceneProps: { position: any; scene: any }) => {

      const { position, scene } = sceneProps

      const { index } = scene

      if (Platform.OS === 'ios') {

        return IosTransition(index, position)

      }

      return DroidTransition(index, position)

    },

  }

}

export { transitionConfig }