

Touch Gesture Detection

Advance Web Development

Time: 60 mins

Introduction

In this class, student/s will be introduced to a component in React Native for making the views respond properly to touch gestures on the mobile app screen. Students/s will learn to add the language buttons to the UI of the Language translator App. Student/s will further learn to create an event each time on the press of a language button.

React Native Commands and Components Introduced

- `<TouchableOpacity>` Wraps the components for making them respond to touches
- `map()` function Is used to iterate through each element in an array and execute a set of instructions on each array element.

Vocabulary

- **TouchableOpacity:** A wrapper for making views respond properly to touches. On press down, the opacity of the wrapped view is decreased, dimming it.
- **Array:** An array is a collection of similar data items stored at contiguous memory locations.
- **map() function:** The `map()` function runs a function on each item in the array, returning a new array as the result.

Learning Objectives

Student/s should be able to:

- **Recall** React Native JSX components.
- **Explain** the arrow and `map()` functions used to iterate a function on each array element.
- **Demonstrate** how to use `<TouchableOpacity>` to make any component sensitive to hand touch gesture.
- **Display** the “initial” and “target” language buttons of the Language Translator App on the preview pane of iOS, Android and Web.

Activities

1. Class Narrative: (2 mins)

- Brief the student/s that they will continue with the creation of the UI for the language translator App using Snack Expo. The App created has the basic UI. We now need to include a feature that lets users choose the languages they want to translate from and into.

2. Concept Introduction Activity: (5 mins)

- Let the student/s play the explore-activity and observe the working of the Language Translator App.
- Guide the student/s to view the same App in Web, android, iOS and mobile phone .
- Using the slides, explain how:
 - There are a set of “initial” and “target” language buttons.
 - The language buttons allow to select the languages one wishes to translate from and into.
 - Each language button comprises of an image and the name of the language.
 - When the button is pressed down the opacity of the button reduces.

3. Activity 1: (15 mins)

Teacher Activity: (6 mins)

- Explain that we can make any component of the app touch sensitive using the Touchable components from react-native library.
- Explain to the student/s how to import <TouchableOpacity> component from react-native library.
- Show how to add an image and text for the “initial” language - English and style them..
- Use <TouchableOpacity> to wrap the above image and language name making them responsive to touch gesture.

Student Activity: (9 mins)

- Guide the students to import TouchableOpacity component from react-native library.
- Guide the students to add the “target” language button - French and style it.

4. Activity 2: (8 mins)

Student Activity: (8 mins)

- Guide the student/s to use the onPress() function and print the respective language name in the console upon pressing the language button.

5. Activity 3: (16 mins)

Teacher Activity: (8 mins)

- Explain the syntax of arrow function and map() function.
- Create an array of “initial” languages that the app works on.
- Apply the map() function on each of the languages in the array to make them touch sensitive and store the result in a new array.
- Display the new array of touchable language buttons in view.

Student Activity: (10 mins)

- Guide the students to create an array of “target” languages that the app works on.
- Guide the students to apply the map() function on each of the languages in the array to make them touch sensitive and store the result in a new array.
- Display the new array of touchable language buttons in view.

6. Introduce the Post class project: (2 min)

- Add and style the search button for image search app..

7. Test and Summarize the class learnings: (5 mins)

- Check for understanding through quizzes and summarize learning after respective missions.
- Summarize the overall class learning towards the end of the class.

8. Additional activities:

- Encourage the student/s to add a Go button to the Monkey Chunky App.
- Encourage the student/s to create an and style phonetic buttons on the Monkey Chunky App.

9. State the Next Class Objective: (1 min)

- We will learn to call an API to enable language translation.

U.S. Standards:

CSTA: 2-AP-13, 2-AP-16, 2-AP-19

Links Table		
Activity	Activity Name	Link
Class Presentation	Touch Gesture Detection	https://s3-whjr-curriculum-uploads.whjr.online/d2bf493e-f4df-44c8-b567-11b7eddf9e21.html
Explore Activity	Touch Gesture Detection	https://snack.expo.dev/@procodinggclass/wad-402---50-sa3-add-language-buttons-array---solution
Teacher Activity 1	Add the First Language Button	https://snack.expo.dev/@procodingclass/wad-402---50-ta1-add-the-first-language-button
Teacher Activity 1 Solution	Add the First Language Button - Solution	https://snack.expo.dev/@procodingclass/wad-402---50-ta1-add-the-first-language-button---solution
Student Activity 1	Add a Touchable Button	https://snack.expo.dev/@procodingclass/wad-402---50-sa1-add-a-touchable-button
Teacher Reference: Student Activity 1 Solution	Add a Touchable Button - Solution	https://snack.expo.dev/@procodingclass/wad-402---50-sa1-add-a-touchable-button---solution
Student Activity 2	Add an onPress Event	https://snack.expo.dev/@procodingclass/wad-402---50-sa2-add-an-onpress-event
Teacher Reference: Student Activity 2 Solution	Add an onPress Event - Solution	https://snack.expo.dev/@procodingclass/wad-402---50-sa2-add-an-onpress-event---solution
Teacher Activity 3	Add the Initial Language Buttons	https://snack.expo.dev/@procodingclass/wad-402---50-ta3-add-the-initial-language-buttons

Teacher Activity 3 Solution	Add the Initial Language Buttons - Solution	https://snack.expo.dev/@procodingclass/wad-402---50-ta3-add-the-initial-language-buttons---solution
Student Activity 3	Add Language Buttons Array	https://snack.expo.dev/@procodingclass/wad-402---50-sa3-add-language-buttons-array
Teacher Reference: Student Activity 3 Solution	Add Language Buttons Array - Solution	https://snack.expo.dev/@procodingclass/wad-402---50-sa3-add-language-buttons-array---solution
Student's Additional Activity 1	Add and Customize a Button	https://snack.expo.dev/@procodingclass/wad-402---50-aa1-add-and-customize-a-button
Teacher Reference: Student's Additional Activity 1 Solution	Add and Customize a Button - Solution	https://snack.expo.dev/@procodingclass/wad-402---50-aa1-add-and-customize-a-button---solution
Student's Additional Activity 2	Create and Style Phonetic Buttons	https://snack.expo.dev/@procodingclass/wad-402---50-aa2-create-and-style-phonetic-buttons
Teacher Reference: Student's Additional Activity 2 Solution	Create and Style Phonetic Buttons - Solution	https://snack.expo.dev/@procodingclass/wad-402---50-aa2-create-and-style-phonetic-buttons---solution
Post Class Project	Add a Search Button	https://snack.expo.dev/@procodingclass/wad-402---50-pcp-add-a-search-button
Teacher Reference: Post Class Project Solution	Add a Search Button - Solution	https://snack.expo.dev/@procodingclass/wad-402---50-pcp-add-a-search-button---solution