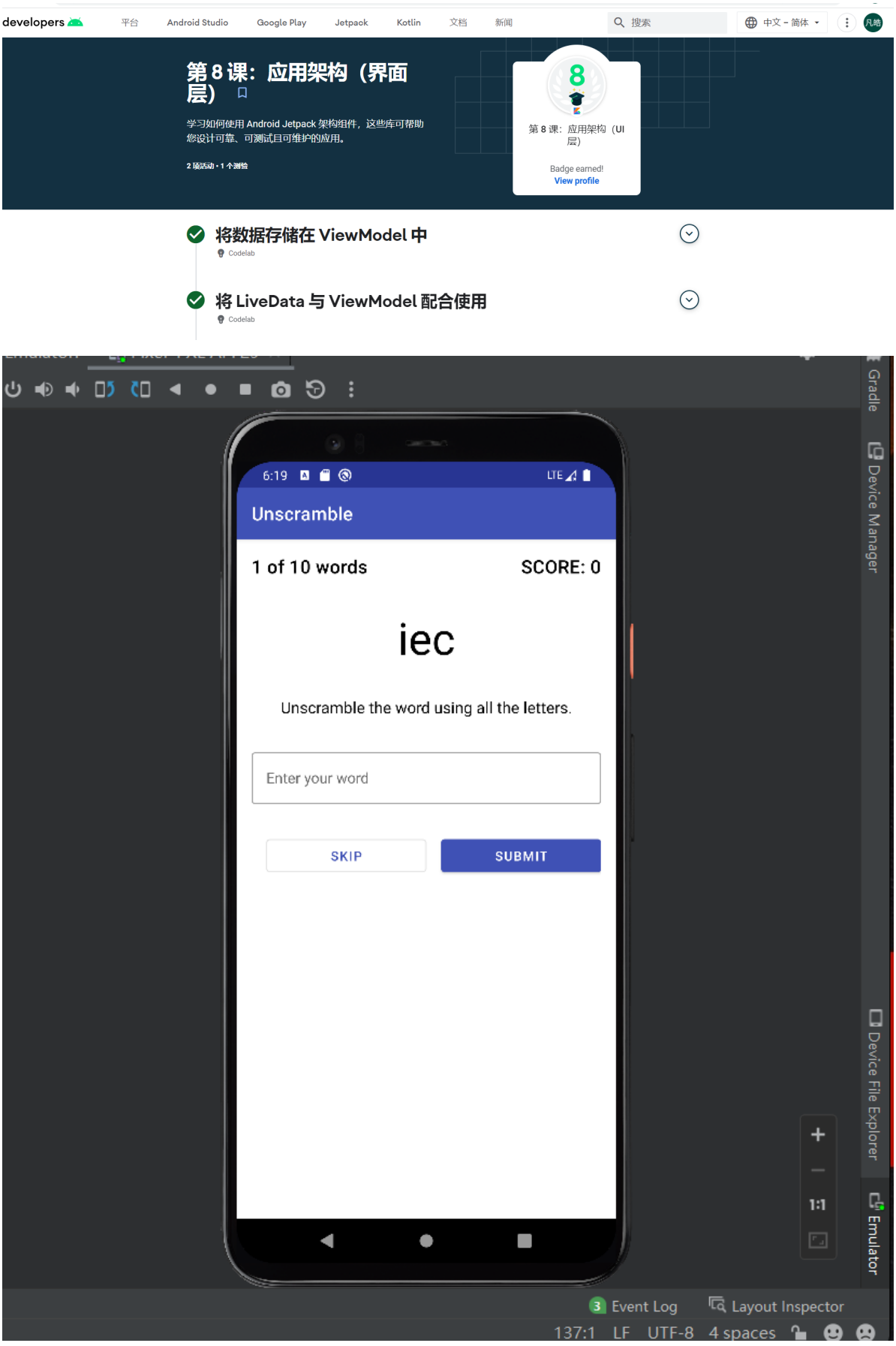
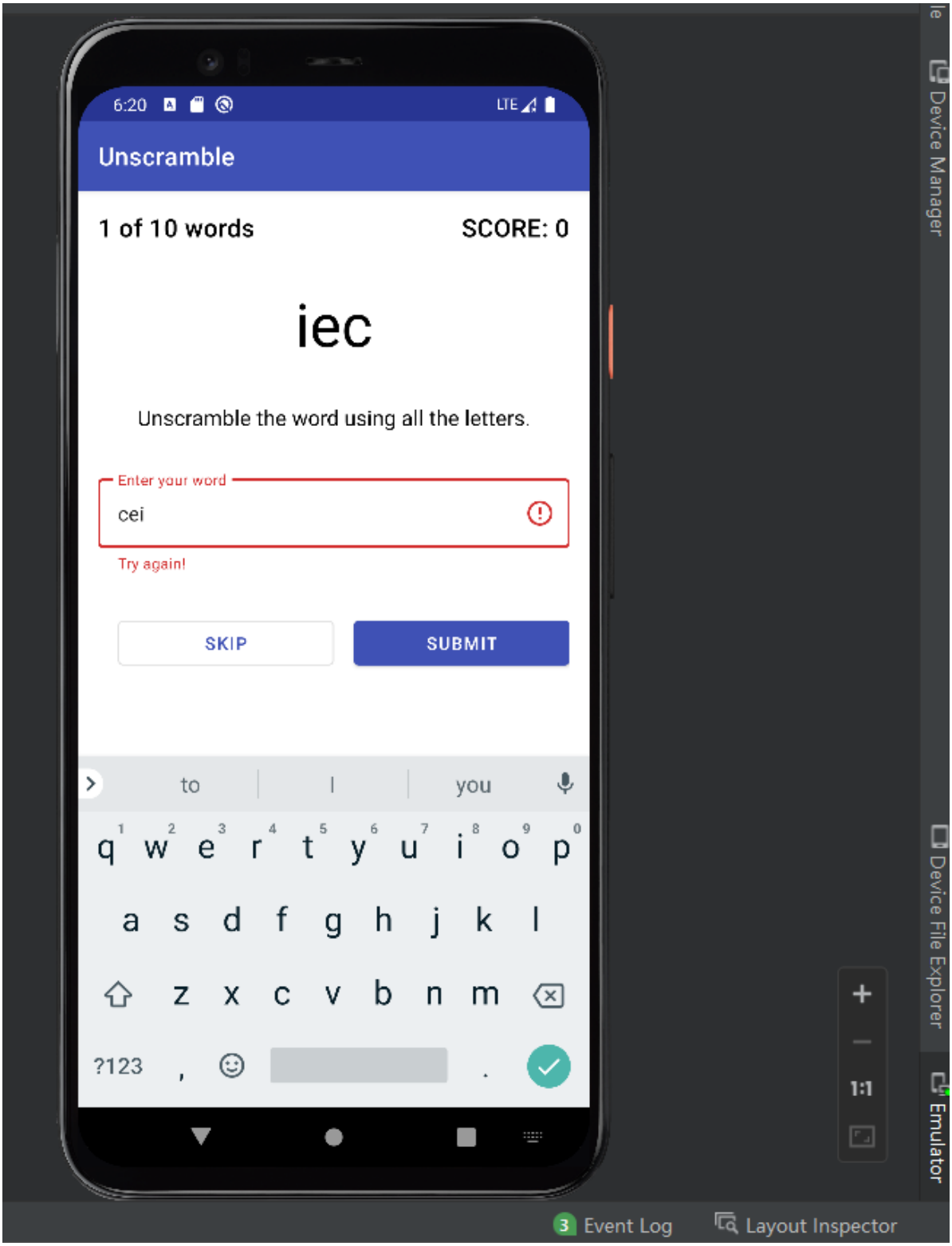
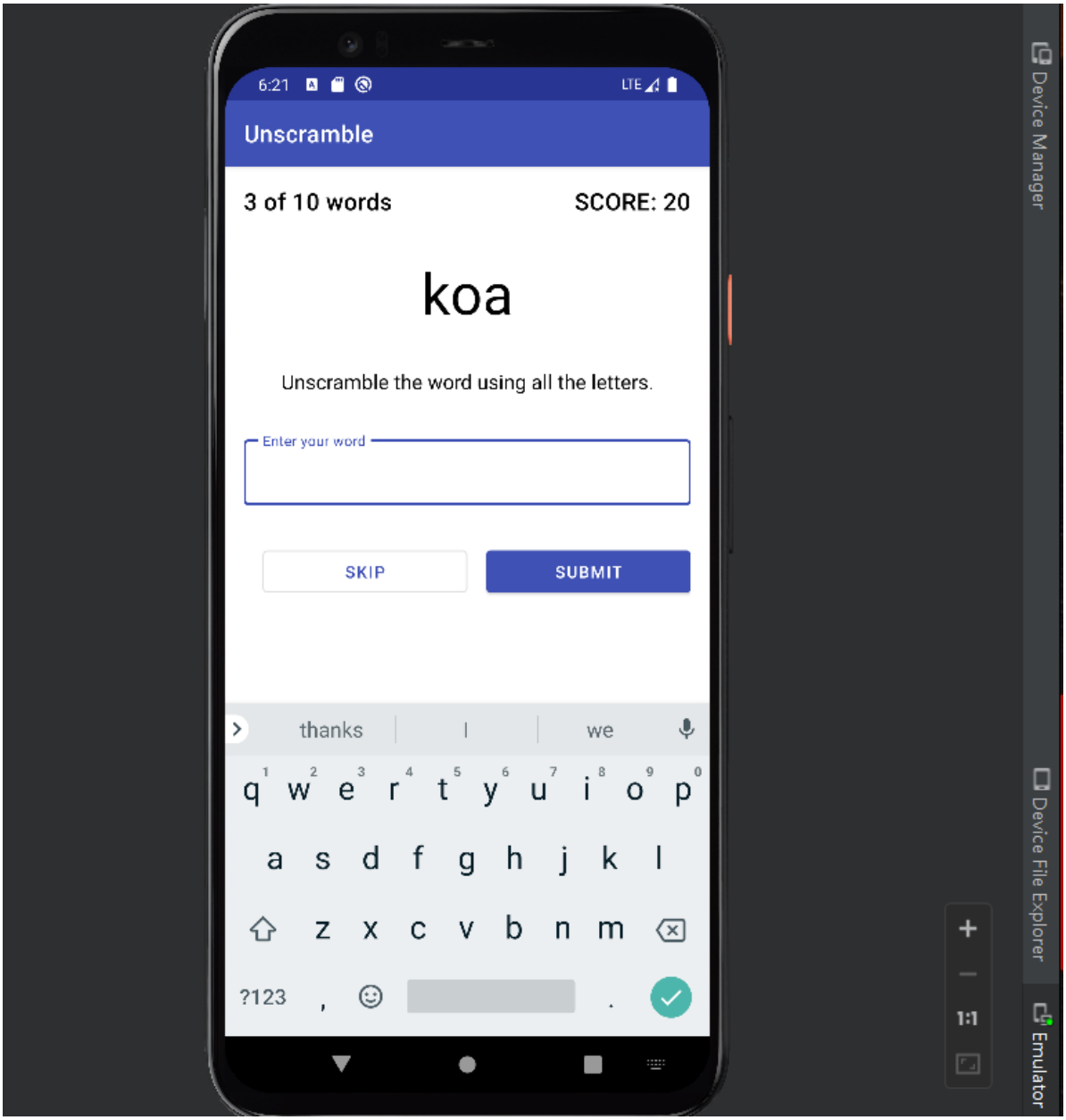
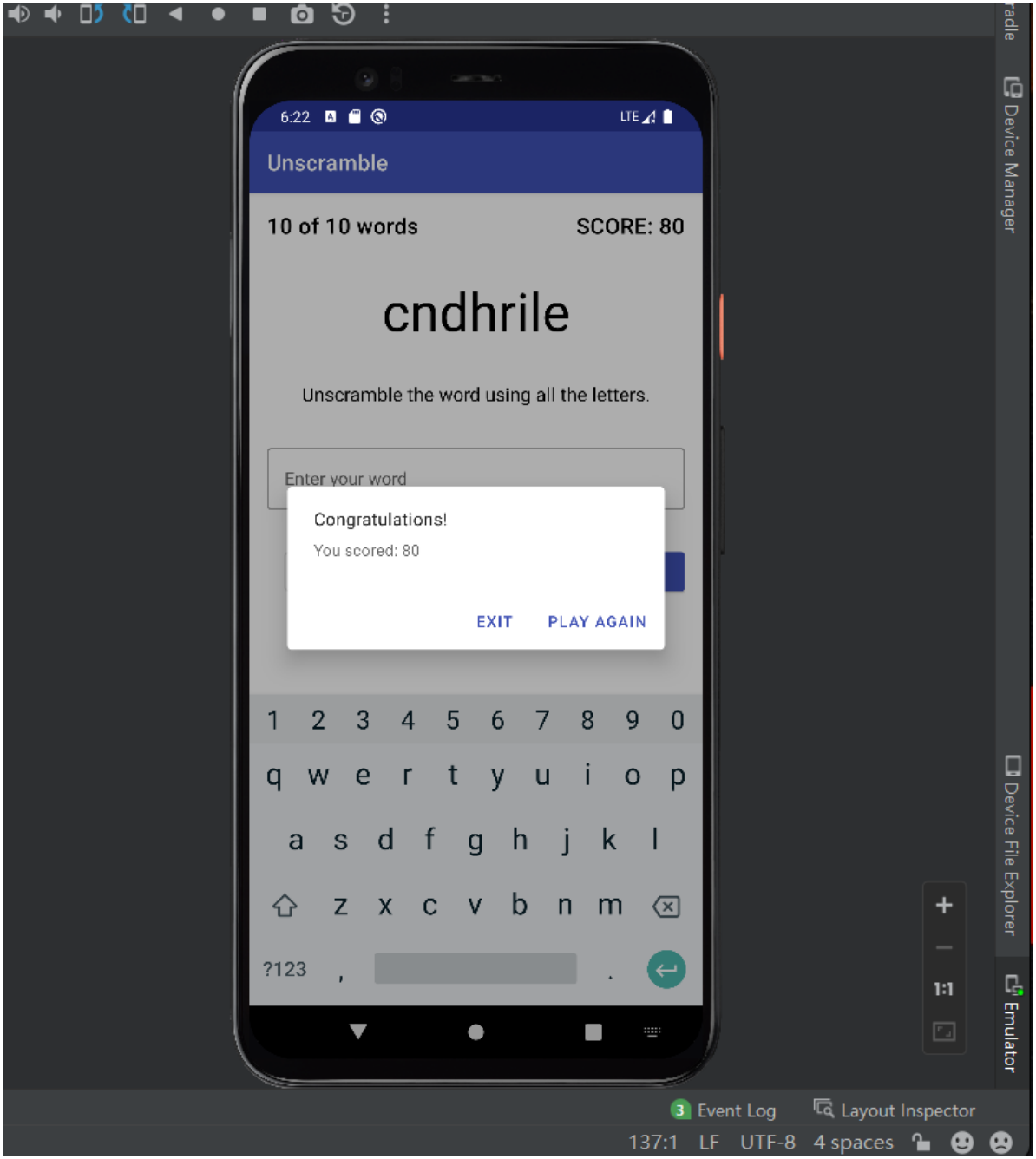
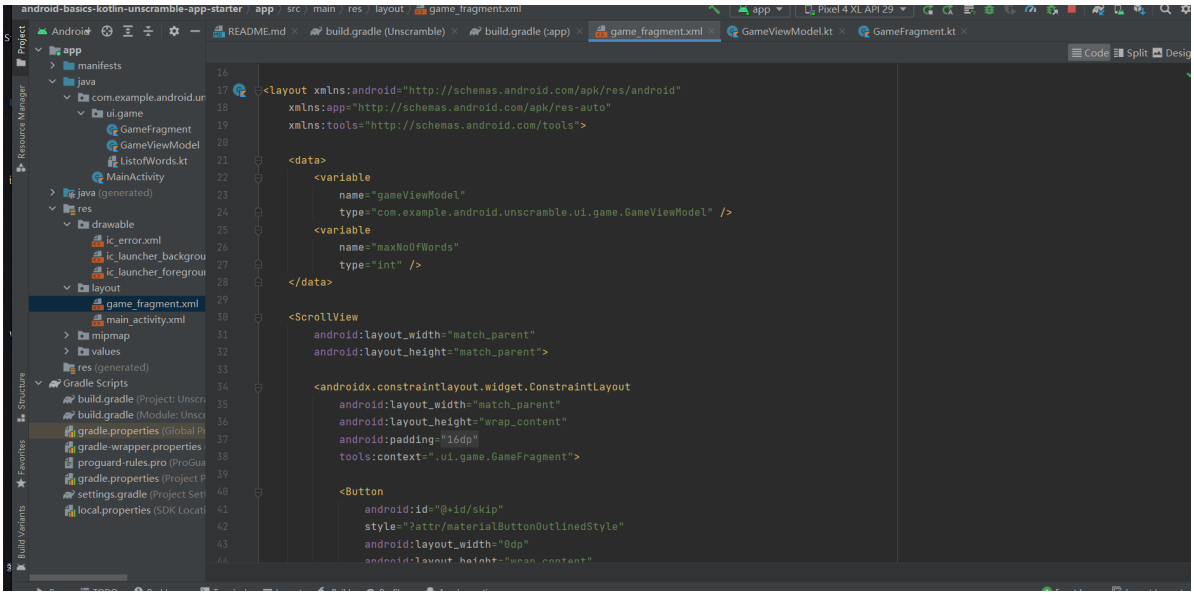
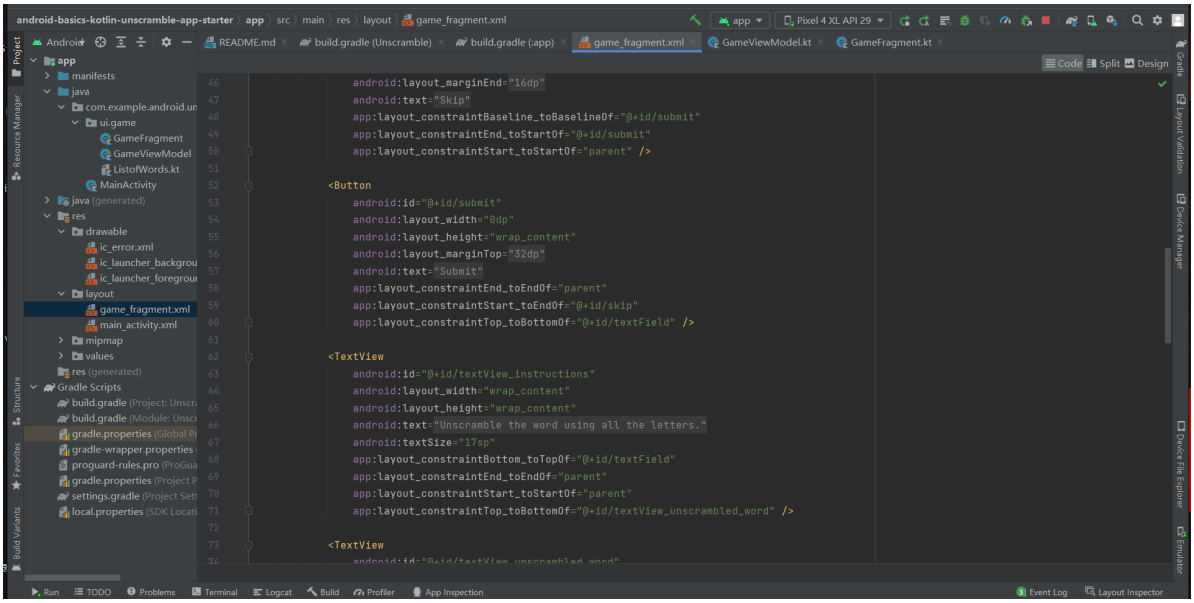

developers
平台
Android Studio
Google Play
Jetpack
Kotlin
文档
新闻
搜索
中文 – 简体
第 8 课：应用架构（界面层）
学习如何使用 Android Jetpack 架构组件，这些库可帮助您设计可靠、可测试且可维护的应用。
2 项活动 • 1 个测验
8
第 8 课：应用架构（UI 层）
Badge earned!
View profile
将数据存储在 ViewModel 中
Codelab
将 LiveData 与 ViewModel 配合使用
Codelab
6:19
LTE
Unscramble
1 of 10 words
SCORE: 0
iec
Unscramble the word using all the letters.
Enter your word
SKIP
SUBMIT
Gradle
Device Manager
Device File Explorer
Emulator
1:1
Event Log
Layout Inspector
137:1
LF
UTF-8
4 spaces


6:20
LTE
Unscramble
1 of 10 words
SCORE: 0
iec
Unscramble the word using all the letters.
Enter your word
cei
Try again!
SKIP
SUBMIT
to
I
you
Device Manager
Device File Explorer
Emulator
1:1
Event Log
Layout Inspector

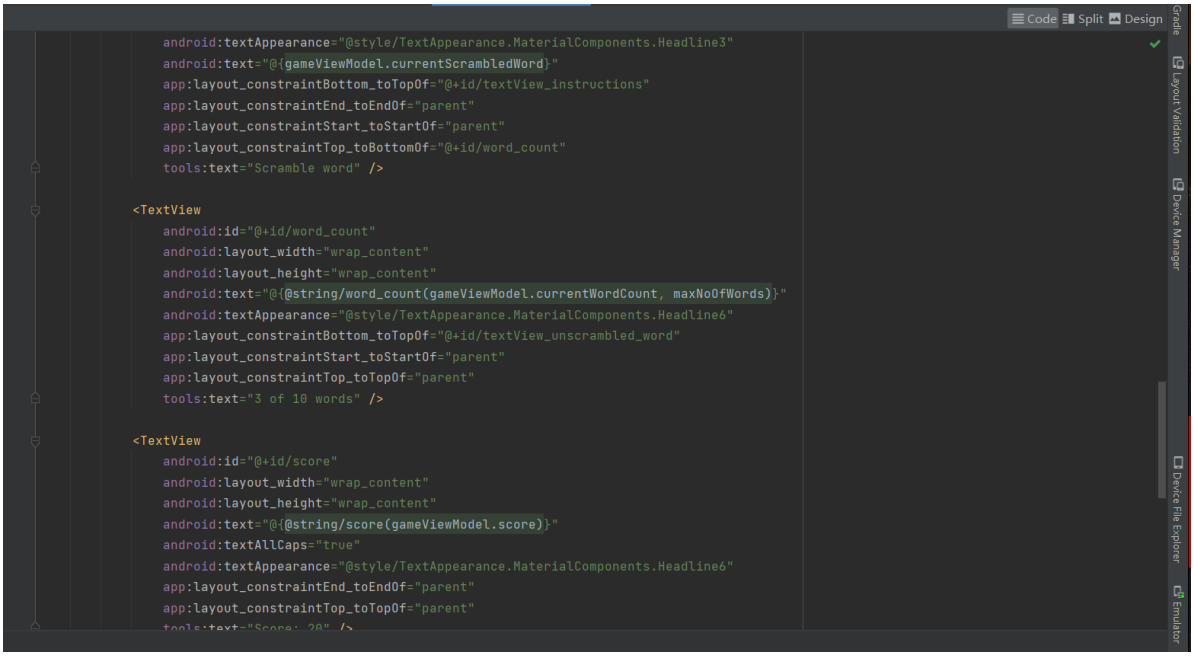

6:21
LTE
Unscramble
3 of 10 words
SCORE: 20
koa
Unscramble the word using all the letters.
Enter your word
SKIP
SUBMIT
thanks
I
we
Device Manager
Device File Explorer
Emulator
1:1


6:22
LTE
Unscramble
10 of 10 words
SCORE: 80
cndhrile
Unscramble the word using all the letters.
Enter your word
Congratulations!
You scored: 80
EXIT
PLAY AGAIN
Device Manager
Device File Explorer
Emulator
Event Log
Layout Inspector
137:1
LF
UTF-8
4 spaces

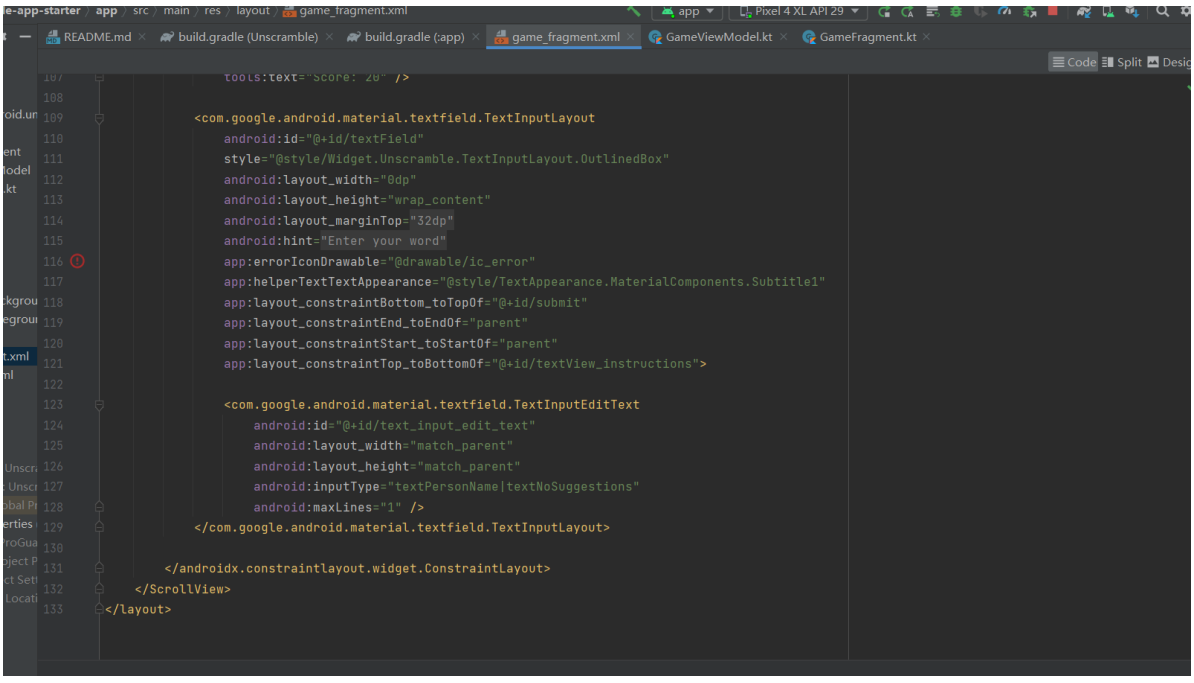

android-basics-kotlin-unscramble-app-starter
app
src
main
res
layout
game_fragment.xml
README.md
build.gradle (Unscramble)
build.gradle (:app)
GameViewModel.kt
GameFragment.kt
<layout xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools">
<data>
<variable
name="gameViewModel"
type="com.example.android.unscramble.ui.game.GameViewModel" />
<variable
name="maxNoOfWords"
type="int" />
</data>
<ScrollView
android:layout_width="match_parent"
android:layout_height="match_parent">
<androidx.constraintlayout.widget.ConstraintLayout
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:padding="16dp"
tools:context=".ui.game.GameFragment">
<Button
android:id="@+id/skip"
style="?attr/materialButtonOutlinedStyle"
android:layout_width="0dp"



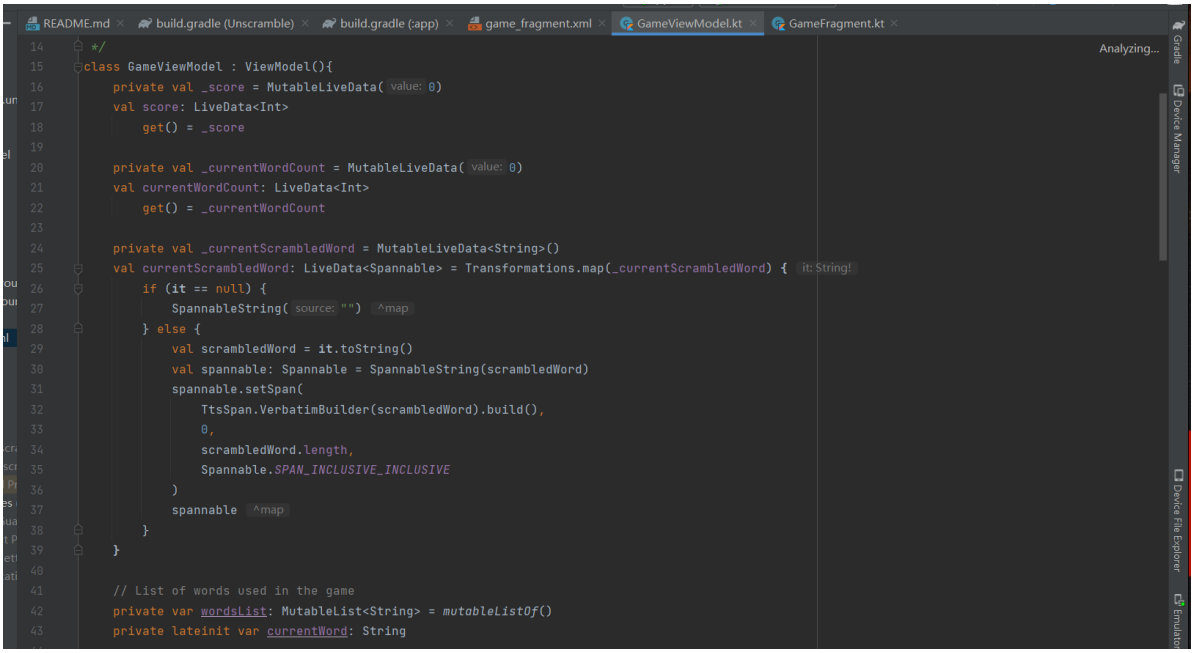
```
                android:layout_marginEnd="16dp"
                android:text="Skip"
                app:layout_constraintBaseline_toBaselineOf="@+id/submit"
                app:layout_constraintEnd_toStartOf="@+id/submit"
                app:layout_constraintStart_toStartOf="parent" />

            <Button
                android:id="@+id/submit"
                android:layout_width="0dp"
                android:layout_height="wrap_content"
                android:layout_marginTop="32dp"
                android:text="Submit"
                app:layout_constraintEnd_toEndOf="parent"
                app:layout_constraintStart_toEndOf="@+id/skip"
                app:layout_constraintTop_toBottomOf="@+id/textField" />

            <TextView
                android:id="@+id/textView_instructions"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="Unscramble the word using all the letters."
                android:textSize="17sp"
                app:layout_constraintBottom_toTopOf="@+id/textField"
                app:layout_constraintEnd_toEndOf="parent"
                app:layout_constraintStart_toStartOf="parent"
                app:layout_constraintTop_toBottomOf="@+id/textView_unscrambled_word" />

            <TextView
```

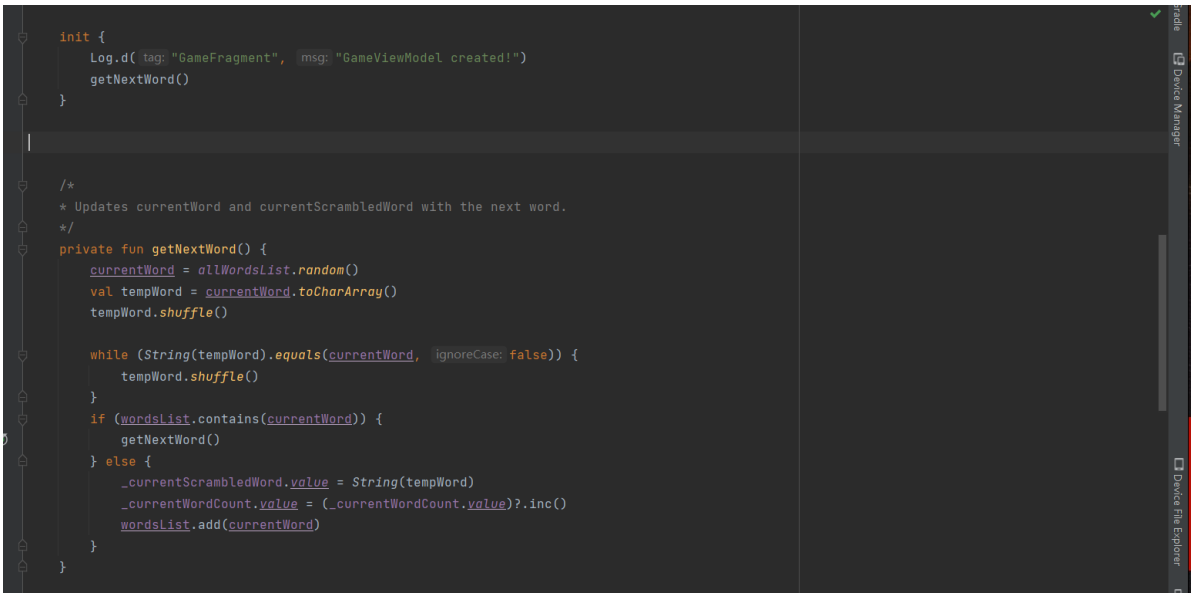


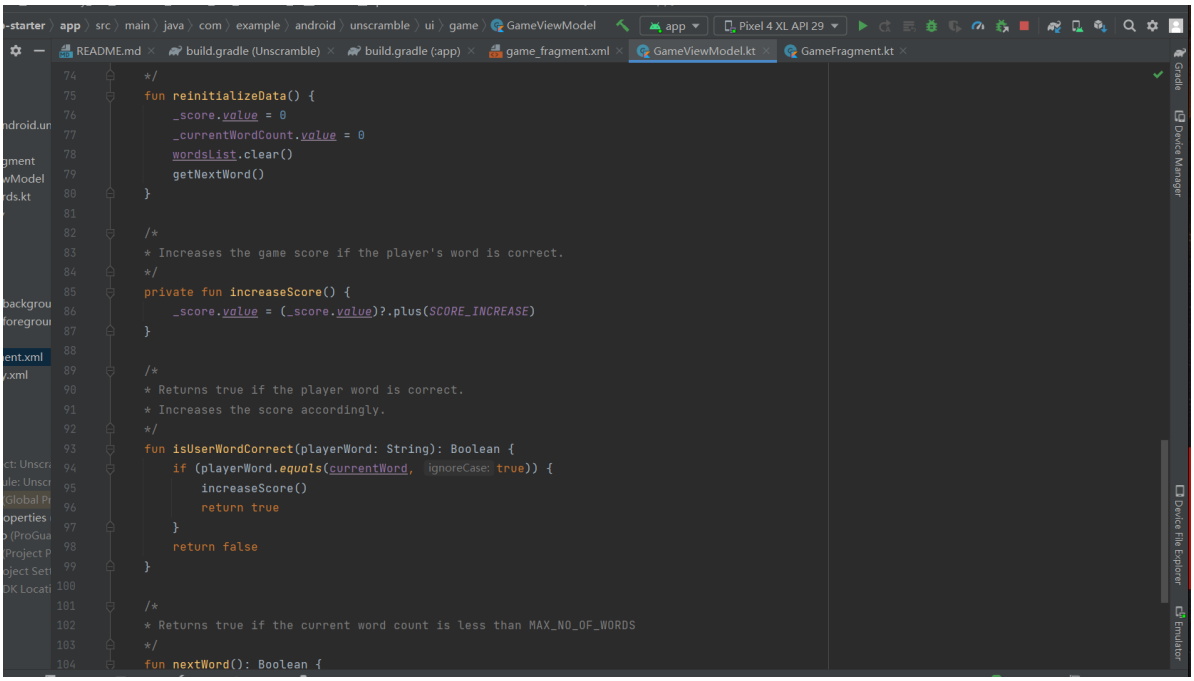
```
        android:textAppearance="@style/TextAppearance.MaterialComponents.Headline3"
        android:text="@{gameViewModel.currentScrambledWord}"
        app:layout_constraintBottom_toTopOf="@+id/textView_instructions"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/word_count"
        tools:text="Scramble word" />

    <TextView
        android:id="@+id/word_count"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@{@string/word_count(gameViewModel.currentWordCount, maxNoOfWords)}"
        android:textAppearance="@style/TextAppearance.MaterialComponents.Headline6"
        app:layout_constraintBottom_toTopOf="@+id/textView_unscrambled_word"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        tools:text="3 of 10 words" />

    <TextView
        android:id="@+id/score"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@{@string/score(gameViewModel.score)}"
        android:textAllCaps="true"
        android:textAppearance="@style/TextAppearance.MaterialComponents.Headline6"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintTop_toTopOf="parent"
```

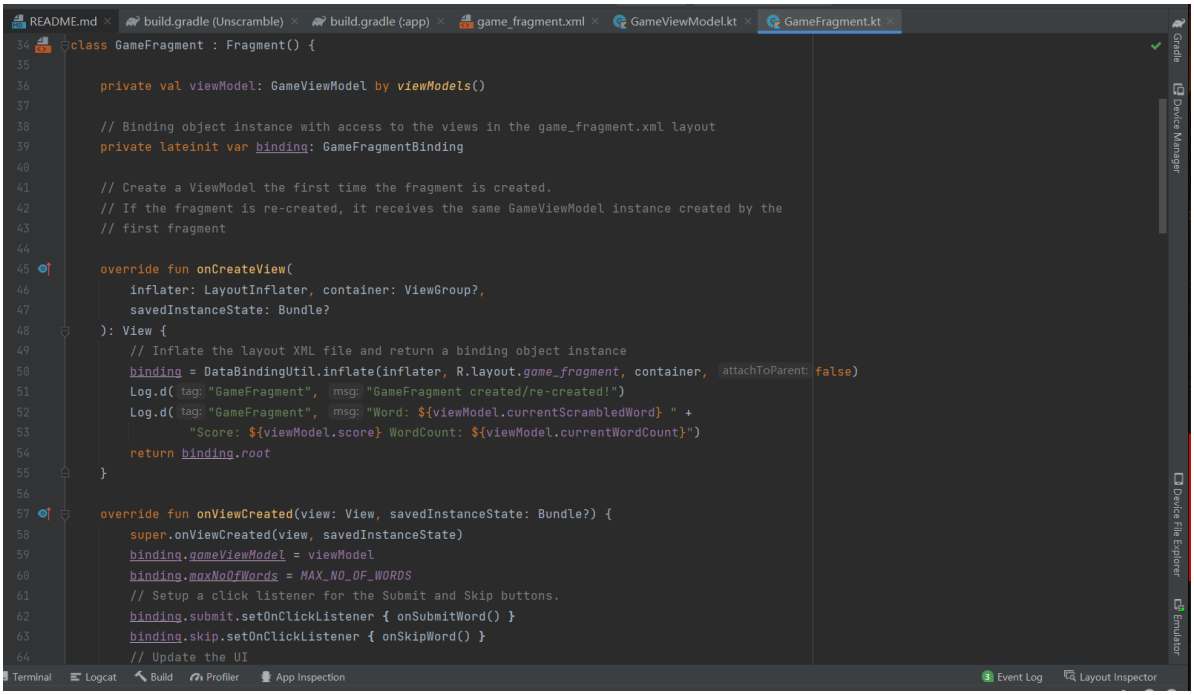


```
            tools:text="Score: 20" />

        <com.google.android.material.textfield.TextInputLayout
            android:id="@+id/textField"
            style="@style/Widget.Unscramble.TextInputLayout.OutlinedBox"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_marginTop="32dp"
            android:hint="Enter your word"
            app:errorIconDrawable="@drawable/ic_error"
            app:helperTextTextAppearance="@style/TextAppearance.MaterialComponents.Subtitle1"
            app:layout_constraintBottom_toTopOf="@+id/submit"
            app:layout_constraintEnd_toEndOf="parent"
            app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintTop_toBottomOf="@+id/textView_instructions">

            <com.google.android.material.textfield.TextInputEditText
                android:id="@+id/text_input_edit_text"
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:inputType="textPersonName|textNoSuggestions"
                android:maxLines="1" />
        </com.google.android.material.textfield.TextInputLayout>

    </androidx.constraintlayout.widget.ConstraintLayout>
</ScrollView>
</layout>
```

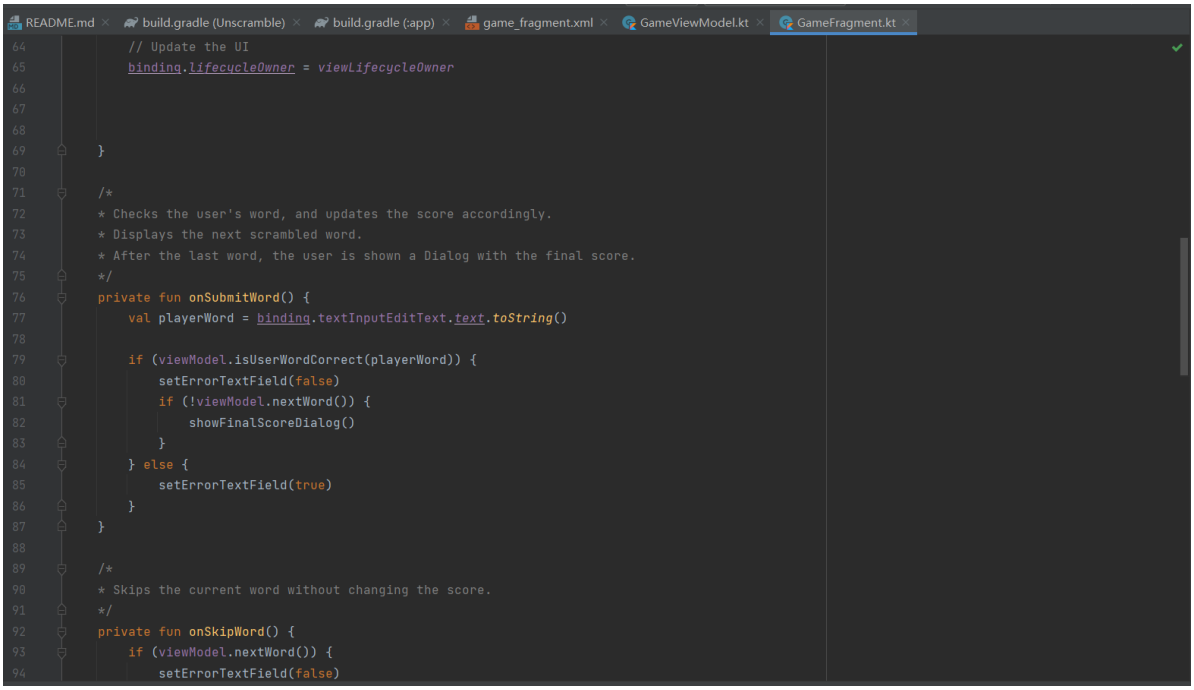

README.md
build.gradle (Unscramble)
build.gradle (:app)
game_fragment.xml
GameViewModel.kt
GameFragment.kt
Analyzing...
*/
class GameViewModel : ViewModel(){
private val _score = MutableLiveData(value: 0)
val score: LiveData<Int>
get() = _score
private val _currentWordCount = MutableLiveData(value: 0)
val currentWordCount: LiveData<Int>
get() = _currentWordCount
private val _currentScrambledWord = MutableLiveData<String>()
val currentScrambledWord: LiveData<Spannable> = Transformations.map(_currentScrambledWord) { it: String!
if (it == null) {
SpannableString(source: "") ^map
} else {
val scrambledWord = it.toString()
val spannable: Spannable = SpannableString(scrambledWord)
spannable.setSpan(
TtsSpan.VerbatimBuilder(scrambledWord).build(),
0,
scrambledWord.length,
Spannable.SPAN_INCLUSIVE_INCLUSIVE
)
spannable ^map
}
}
// List of words used in the game
private var wordsList: MutableList<String> = mutableListOf()
private lateinit var currentWord: String


init {
Log.d(tag: "GameFragment", msg: "GameViewModel created!")
getNextWord()
}
/*
* Updates currentWord and currentScrambledWord with the next word.
*/
private fun getNextWord() {
currentWord = allWordsList.random()
val tempWord = currentWord.toCharArray()
tempWord.shuffle()
while (String(tempWord).equals(currentWord, ignoreCase: false)) {
tempWord.shuffle()
}
if (wordsList.contains(currentWord)) {
getNextWord()
} else {
_currentScrambledWord.value = String(tempWord)
_currentWordCount.value = (_currentWordCount.value)?.inc()
wordsList.add(currentWord)
}
}

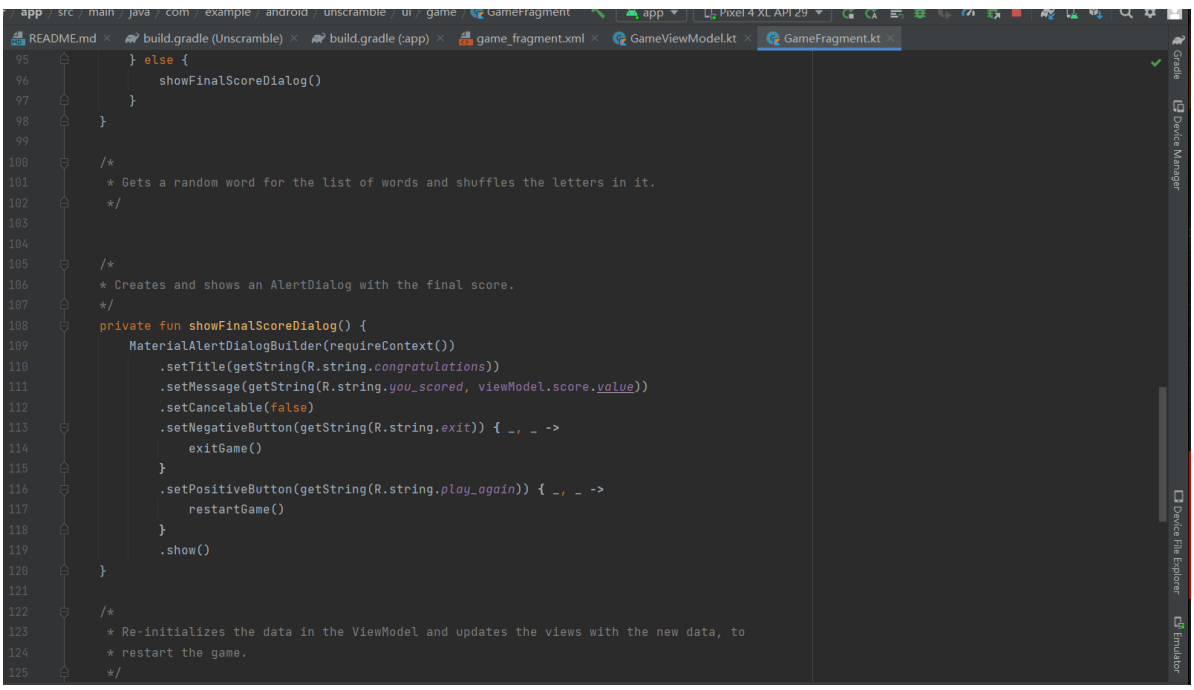

app
src
main
java
com
example
android
unscramble
ui
game
GameViewModel
Pixel 4 XL API 29
README.md
build.gradle (Unscramble)
build.gradle (:app)
game_fragment.xml
GameViewModel.kt
GameFragment.kt
*/
fun reinitializeData() {
_score.value = 0
_currentWordCount.value = 0
wordsList.clear()
getNextWord()
}
/*
* Increases the game score if the player's word is correct.
*/
private fun increaseScore() {
_score.value = (_score.value)?.plus(SCORE_INCREASE)
}
/*
* Returns true if the player word is correct.
* Increases the score accordingly.
*/
fun isUserWordCorrect(playerWord: String): Boolean {
if (playerWord.equals(currentWord, ignoreCase: true)) {
increaseScore()
return true
}
return false
}
/*
* Returns true if the current word count is less than MAX_NO_OF_WORDS
*/
fun nextWord(): Boolean {



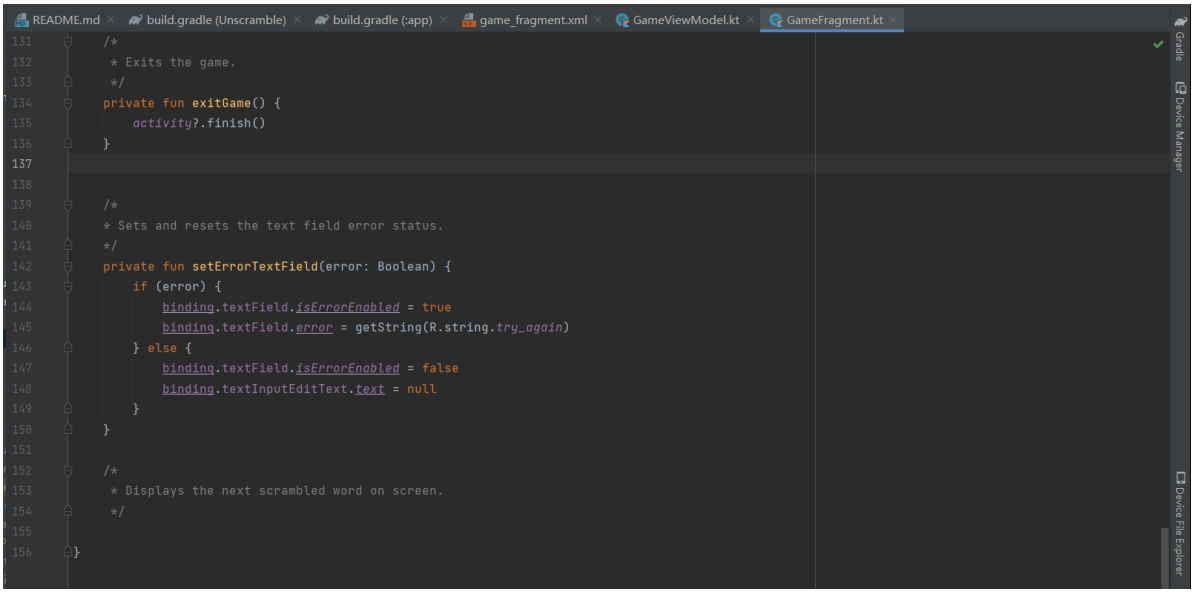
```
class GameFragment : Fragment() {

    private val viewModel: GameViewModel by viewModels()

    // Binding object instance with access to the views in the game_fragment.xml layout
    private lateinit var binding: GameFragmentBinding

    // Create a ViewModel the first time the fragment is created.
    // If the fragment is re-created, it receives the same GameViewModel instance created by the
    // first fragment

    override fun onCreateView(
        inflater: LayoutInflater, container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View {
        // Inflate the layout XML file and return a binding object instance
        binding = DataBindingUtil.inflate(inflater, R.layout.game_fragment, container, attachToParent: false)
        Log.d( tag: "GameFragment",  msg: "GameFragment created/re-created!")
        Log.d( tag: "GameFragment",  msg: "Word: ${viewModel.currentScrambledWord} " +
                "Score: ${viewModel.score} WordCount: ${viewModel.currentWordCount}")
        return binding.root
    }

    override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
        super.onViewCreated(view, savedInstanceState)
        binding.gameViewModel = viewModel
        binding.maxNoOfWords = MAX_NO_OF_WORDS
        // Setup a click listener for the Submit and Skip buttons.
        binding.submit.setOnClickListener { onSubmitWord() }
        binding.skip.setOnClickListener { onSkipWord() }
        // Update the UI
```



```
        // Update the UI
        binding.lifecycleOwner = viewLifecycleOwner




    }

    /*
    * Checks the user's word, and updates the score accordingly.
    * Displays the next scrambled word.
    * After the last word, the user is shown a Dialog with the final score.
    */
    private fun onSubmitWord() {
        val playerWord = binding.textInputEditText.text.toString()

        if (viewModel.isUserWordCorrect(playerWord)) {
            setErrorTextField(false)
            if (!viewModel.nextWord()) {
                showFinalScoreDialog()
            }
        } else {
            setErrorTextField(true)
        }
    }

    /*
    * Skips the current word without changing the score.
    */
    private fun onSkipWord() {
        if (viewModel.nextWord()) {
            setErrorTextField(false)
```



```
        } else {
            showFinalScoreDialog()
        }
    }

    /*
     * Gets a random word for the list of words and shuffles the letters in it.
     */


    /*
    * Creates and shows an AlertDialog with the final score.
    */
    private fun showFinalScoreDialog() {
        MaterialAlertDialogBuilder(requireContext())
            .setTitle(getString(R.string.congratulations))
            .setMessage(getString(R.string.you_scored, viewModel.score.value))
            .setCancelable(false)
            .setNegativeButton(getString(R.string.exit)) { _, _ ->
                exitGame()
            }
            .setPositiveButton(getString(R.string.play_again)) { _, _ ->
                restartGame()
            }
            .show()
    }

    /*
     * Re-initializes the data in the ViewModel and updates the views with the new data, to
     * restart the game.
     */
```


README.md
build.gradle (Unscramble)
build.gradle (:app)
game_fragment.xml
GameViewModel.kt
GameFragment.kt
Gradle
Device Manager
Device File Explorer
/*
* Exits the game.
*/
private fun exitGame() {
activity?.finish()
}
/*
* Sets and resets the text field error status.
*/
private fun setErrorTextField(error: Boolean) {
if (error) {
binding.textField.isErrorEnabled = true
binding.textField.error = getString(R.string.try_again)
} else {
binding.textField.isErrorEnabled = false
binding.textInputEditText.text = null
}
}
/*
* Displays the next scrambled word on screen.
*/
}