# ЛАБОРАТОРНА РОБОТА №1

# НАЛАШТУВАННЯ СЕРЕДОВИЩА РОЗРОБКИ МОБІЛЬНИХ ДОДАТКІВ ДЛЯ ОС ANDROID

**Мета роботи:**

* Налаштувати середовище розробки додатків для операційної системи Android.
* Ознайомитись з типовою структурою Android-проекту.
* Ознайомитись з системою автоматичного збирання проектів Gradle.
* Навчитись розробляти прості Android-додатки з використанням стандартної бібліотеки візуальних компонентів.

## Хід виконання роботи

### Завдання згідно з варіантом №8

Створити Android-додаток «Віднови слово». Додаток складається з екранів головного меню, налаштувань та екрану гри. - Головне меню містить інформацію про розробника, кнопки «Налаштування», «Розпочати гру». - Екран налаштувань дозволяє задати мінімальний (4,5,6) та максимальний (7,8,9) розмір слів у символах та час зворотного відліку (1 хв, 2 хв, 5 хв). - Екран гри показує час, який залишивсь до закінчення гри, кількість вгаданих слів. Гравцеві показується слово з перемішаними випадковим чином буквами. Гравець може натискати на букви, складаючи з них слово. Наприклад, програма показує букви «ідроа». Натискаючи послідовно на букви «р», «а», «д», «і», «о» гравець складає слово «радіо»:

і д р о а і д - о а і д - о - і - - о - - - - о - - - - - - - - - - - р - - - - р а - - - р а д - - р а д і - р а д і о

У випадку помилки, гравець може вернути букву назад вгору, натиснувши на неї. Слово заповнюється послідовно, зліва направо. Якщо гравець вгадав слово, програма показує Toast-повідомлення «Вгадано» та генерує наступне слово. Гра закінчується, коли час вичерпано. Гравцеві показується діалог з результатами: кількість вгаданих слів. Навігація має бути побудована на базі Activity. Компонування інтерфейсу виконати на базі ConstraintLayout, LinearLayout.

### Результати виконання зі скріншотом екрану результатів розрахунку варіанту завдання

Дану лабораторну роботу було розміщено на безкоштовному вебсервісі GitHub за посиланням: [github.com/hellstaff-valera/lab1](https://github.com/hellstaff-valera/ProgrammingOfMobileDevices/tree/master/lab1).

Для Android-додатку «Віднови слово» було розроблено 3 Activity, 1 Class,1 Adapter та 4 layout’и.

Код класу MainActivity наведено в лістингу 1.1 – це клас який відображає за допомогою layout activity\_main наведено в лістингу 1.2 головний екран на рисунку 1.1.

Код класу GameActivity наведено в лістингу 1.3 – це клас який відображає за допомогою layout activity\_game наведено в лістингу 1.4 екран гри на рисунку 1.2.

Код класу SettingsActivity наведено в лістингу 1.5 – це клас який відображає за допомогою layout settings\_activity наведено в лістингу 1.6 екран налаштувань на рисунку 1.3.

Лістинг 1.1 – MainActivity

package com.hellstaff.lab1  
import android.content.Intent  
import androidx.appcompat.app.AppCompatActivity  
import android.os.Bundle  
import com.hellstaff.lab1.databinding.ActivityMainBinding  
  
class MainActivity : AppCompatActivity() {  
 private lateinit var binding: ActivityMainBinding  
  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
  
 binding = ActivityMainBinding.inflate(*layoutInflater*)  
 setContentView(binding.*root*)  
  
 binding.settings.setOnClickListener **{** startActivity(Intent(this, SettingsActivity::class.*java*))  
 **}** binding.startGame.setOnClickListener **{** startActivity(Intent(this, GameActivity::class.*java*))  
 **}** }  
}

Лістинг 1.2 – activity\_main

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout 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"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".MainActivity">  
  
 <TextView  
 android:id="@+id/infoDeveloper"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:elegantTextHeight="false"  
 android:text="@string/info"  
 android:textAlignment="center"  
 android:textSize="34sp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
 <Button  
 android:id="@+id/settings"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="@string/settings"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="1.0"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
 <Button  
 android:id="@+id/startGame"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="@string/startGame"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/infoDeveloper" />  
  
</androidx.constraintlayout.widget.ConstraintLayout>

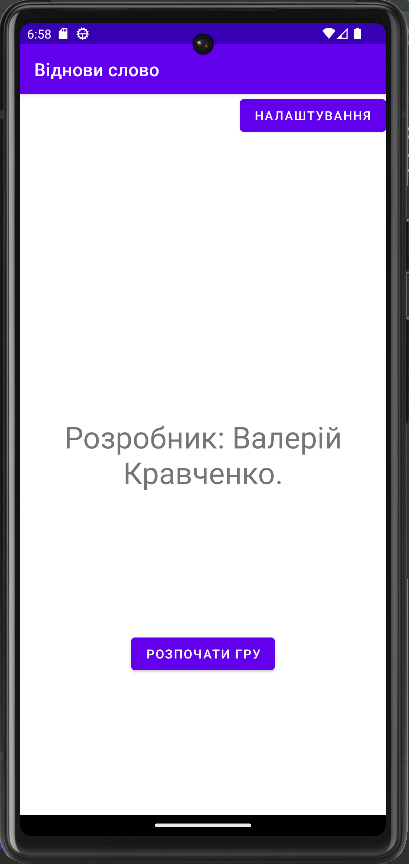


Рисунок 1.1 – головний екран.

Лістинг 1.3 – GameActivity

package com.hellstaff.lab1  
  
import android.annotation.SuppressLint  
import android.app.AlertDialog  
import android.os.Bundle  
import android.os.CountDownTimer  
import android.util.Log  
import android.widget.Toast  
import androidx.appcompat.app.AppCompatActivity  
import androidx.recyclerview.widget.GridLayoutManager  
import com.hellstaff.lab1.databinding.ActivityGameBinding  
import java.io.BufferedReader  
import java.io.File  
import java.io.FileInputStream  
import java.io.InputStreamReader  
  
class GameActivity : AppCompatActivity(), Adaptor.CheckWord {  
 private lateinit var binding: ActivityGameBinding  
 private val filename = "settings"  
 private var length: Int = 0  
 private var time: Long = 0  
 private var counterTrueWord: Short = 0  
 private var timerWork: Boolean = false  
 private var timer: CountDownTimer? = null  
 private var word: String = ""  
  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 binding = ActivityGameBinding.inflate(layoutInflater)  
 setContentView(binding.root)  
 if (!File(filename).exists() && filename.trim() != "") {  
 val fileInputStream: FileInputStream? = openFileInput(filename)  
 val inputStreamReader = InputStreamReader(fileInputStream)  
 val bufferedReader = BufferedReader(inputStreamReader)  
 val stringBuilder: StringBuilder = StringBuilder()  
 var text: String?  
 while (run **{** text = bufferedReader.readLine()  
 text  
 **}** != null) {  
 stringBuilder.append(text)  
 }  
 Log.d("onCreate", "File -> $stringBuilder")  
 length = stringBuilder.split(' ')[2].toInt()  
 time = stringBuilder.split(' ')[3].toLong() \* 60 \* 1000  
 }  
 }  
  
 override fun onStart() {  
 super.onStart()  
 counterTrueWord = 0  
 startCountDownTimer(time)  
 timerWork = true  
 }  
  
 override fun onResume() {  
 super.onResume()  
 val letters = fetchDate()  
 letters.shuffle()  
 val adaptor = Adaptor(this, letters, fetchDate("-"))  
 binding.word.layoutManager = GridLayoutManager(this, length)  
 binding.word.adapter = adaptor  
 }  
  
 private fun fetchDate(): ArrayList<String> {  
 var letters = ""  
 val wordTrue = ArrayList<String>()  
 when (length) {  
 4 -> {  
 val number = (0..Word().wordChar4.size).random()  
 letters = Word().wordChar4[number]  
 }  
 5 -> {  
 val number = (0..Word().wordChar5.size).random()  
 letters = Word().wordChar5[number]  
 }  
 6 -> {  
 val number = (0..Word().wordChar6.size).random()  
 letters = Word().wordChar6[number]  
 }  
 7 -> {  
 val number = (0..Word().wordChar7.size).random()  
 letters = Word().wordChar7[number]  
 }  
 8 -> {  
 val number = (0..Word().wordChar8.size).random()  
 letters = Word().wordChar8[number]  
 }  
 9 -> {  
 val number = (0..Word().wordChar9.size).random()  
 letters = Word().wordChar9[number]  
 }  
 }  
 word = letters  
 letters.forEach **{** element **->** wordTrue.add(element.toString()) **}** return wordTrue  
 }  
  
 private fun fetchDate(symbol: String): ArrayList<String> {  
 val item = ArrayList<String>()  
 for (i in 0 until length) {  
 item.add(symbol)  
 }  
 return item  
 }  
  
 private fun startCountDownTimer(timeMillis: Long) {  
 timer?.cancel()  
 timer = object : CountDownTimer(timeMillis, 1000) {  
 @SuppressLint("SetTextI18n")  
 override fun onTick(timeMili: Long) {  
 if ((timeMili / 1000 % 60).toString().length == 1) binding.textTimer.text =  
 "0" + (timeMili / 60000).toString() + ":0" + (timeMili / 1000 % 60).toString()  
 else binding.textTimer.text =  
 "0" + (timeMili / 60000).toString() + ":" + (timeMili / 1000 % 60).toString()  
 }  
  
 override fun onFinish() {  
 timerWork = false  
 binding.textTimer.text = "Час вийшов!"  
 onCreateDialog()  
 }  
 }.start()  
 }  
  
 override fun checkWord(word: ArrayList<String>) {  
 Log.d("checkWord", word.joinToString(""))  
 if (word.joinToString("") == this.word) {  
 counterTrueWord++  
 Toast.makeText(this, "Вгадано! $counterTrueWord", Toast.LENGTH\_LONG).show()  
 onResume()  
 }  
 }  
 private fun onCreateDialog() {  
 val builder = AlertDialog.Builder(this)  
 builder.setTitle("Гру завершено")  
 builder.setMessage("Ваш результат: $counterTrueWord")  
 builder.setNeutralButton("Почати ще раз") **{** \_, \_ **->** onStart() **}** builder.show()  
 }  
}

Лістинг 1.4 – activity\_game

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 xmlns:app="http://schemas.android.com/apk/res-auto">  
  
 <TextView  
 android:id="@+id/textTimer"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="@string/timer"  
 android:textSize="34sp"  
 app:layout\_constraintBottom\_toTopOf="@+id/word"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
 <androidx.recyclerview.widget.RecyclerView  
 android:id="@+id/word"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
  
</androidx.constraintlayout.widget.ConstraintLayout>

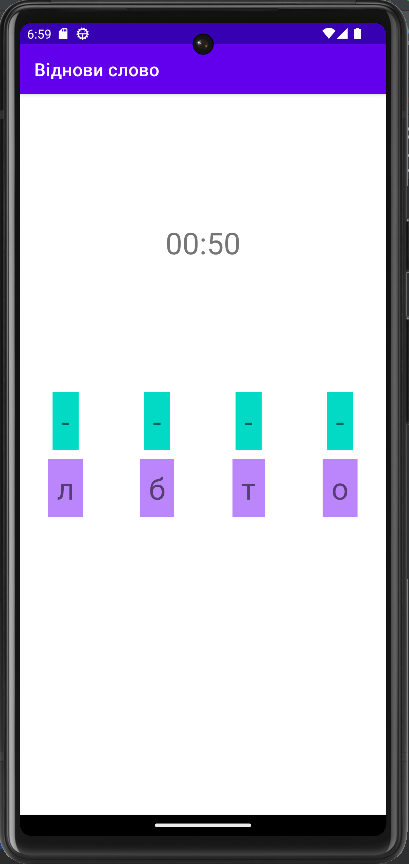


Рисунок 1.2 – екран гри.

Лістинг 1.5 – SettingsActivity

package com.hellstaff.lab1  
  
import android.content.Context  
import android.os.Bundle  
import android.view.View  
import android.widget.AdapterView  
import androidx.appcompat.app.AppCompatActivity  
import com.hellstaff.lab1.databinding.ActivitySettingsBinding  
import java.io.\*  
  
class SettingsActivity : AppCompatActivity() {  
 private lateinit var binding: ActivitySettingsBinding  
 private val filename = "settings"  
  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 binding = ActivitySettingsBinding.inflate(*layoutInflater*)  
 setContentView(binding.*root*)  
  
 if (File(filename).exists()) {  
 val fileInputStream: FileInputStream? = openFileInput(filename)  
 val inputStreamReader = InputStreamReader(fileInputStream)  
 val bufferedReader = BufferedReader(inputStreamReader)  
 val stringBuilder: StringBuilder = StringBuilder()  
 var text: String?  
 while (*run* **{** text = bufferedReader.readLine()  
 text  
 **}** != null) {  
 stringBuilder.append(text)  
 }  
 //Log.d("onCreate", "File -> $stringBuilder")  
 binding.words.setSelection(stringBuilder.*split*(' ')[0].*toInt*())  
 binding.timers.setSelection(stringBuilder.*split*(' ')[1].*toInt*())  
 }  
  
 binding.words.*onItemSelectedListener* = object : AdapterView.OnItemSelectedListener {  
 override fun onItemSelected(  
 parent: AdapterView<\*>?,  
 view: View?,  
 position: Int,  
 id: Long  
 ) {  
 }  
  
 override fun onNothingSelected(parent: AdapterView<\*>?) {}  
 }  
 binding.timers.*onItemSelectedListener* = object : AdapterView.OnItemSelectedListener {  
 override fun onItemSelected(  
 parent: AdapterView<\*>?,  
 view: View?,  
 position: Int,  
 id: Long  
 ) {  
 }  
  
 override fun onNothingSelected(parent: AdapterView<\*>?) {}  
 }  
 }  
  
 override fun onPause() {  
 super.onPause()  
 val data =  
 binding.words.*selectedItemPosition*.toString() + ' ' + binding.timers.*selectedItemPosition*.toString() + ' ' + binding.words.*selectedItem*.toString()  
 .*split*(' ')[0] + ' ' + binding.timers.*selectedItem*.toString().*split*(' ')[0]  
 val fileOutputStream: FileOutputStream  
 try {  
 fileOutputStream = openFileOutput(filename, Context.*MODE\_PRIVATE*)  
 fileOutputStream.write(data.*toByteArray*())  
 } catch (e: FileNotFoundException) {  
 e.printStackTrace()  
 } catch (e: NumberFormatException) {  
 e.printStackTrace()  
 } catch (e: IOException) {  
 e.printStackTrace()  
 } catch (e: Exception) {  
 e.printStackTrace()  
 }  
 }  
}

Лістинг 1.6 – settings\_activity

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".SettingsActivity"  
 android:gravity="center"  
 android:orientation="vertical"  
 android:padding="20dp">  
  
 <View  
 android:layout\_width="wrap\_content"  
 android:layout\_height="3dp"  
 android:background="@color/teal\_700" />  
  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="@string/word\_header"  
 android:textColor="@color/teal\_200"  
 android:padding="10dp"  
 android:textSize="20sp"/>  
  
 <LinearLayout  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:gravity="center\_horizontal">  
  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="@string/word\_title"  
 android:textColor="@color/black"  
 android:textSize="16sp"/>  
  
 <Spinner  
 android:id="@+id/words"  
 android:layout\_width="150dp"  
 android:layout\_height="wrap\_content"  
 android:entries="@array/words"  
 android:padding="10dp"  
 android:prompt="@string/word\_title"/>  
  
 </LinearLayout>  
  
  
  
 <View  
 android:layout\_width="wrap\_content"  
 android:layout\_height="3dp"  
 android:background="@color/teal\_700" />  
  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="@string/timer\_header"  
 android:textColor="@color/teal\_200"  
 android:padding="10dp"  
 android:textSize="20sp"/>  
  
 <LinearLayout  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:gravity="center\_horizontal">  
  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="@string/timer\_title"  
 android:textColor="@color/black"  
 android:textSize="16sp"/>  
  
 <Spinner  
 android:id="@+id/timers"  
 android:layout\_width="150dp"  
 android:layout\_height="wrap\_content"  
 android:entries="@array/timers"  
 android:padding="10dp"  
 android:prompt="@string/word\_title"/>  
  
 </LinearLayout>  
  
 <View  
 android:layout\_width="wrap\_content"  
 android:layout\_height="3dp"  
 android:background="@color/teal\_700" />  
  
</LinearLayout>

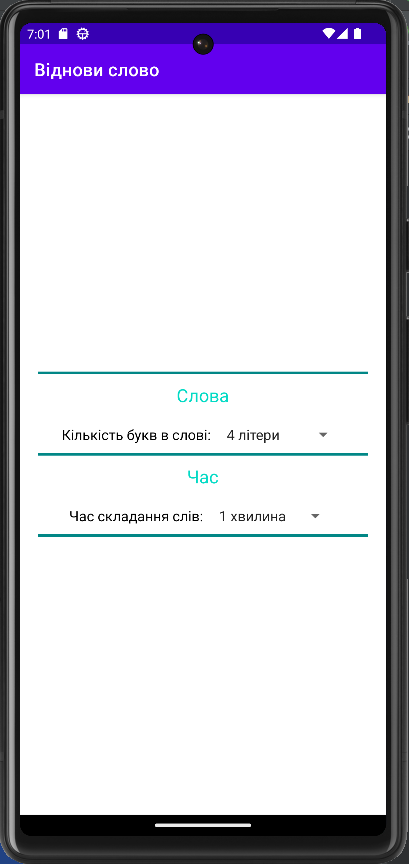


Рисунок 1.3 – екран налаштувань.

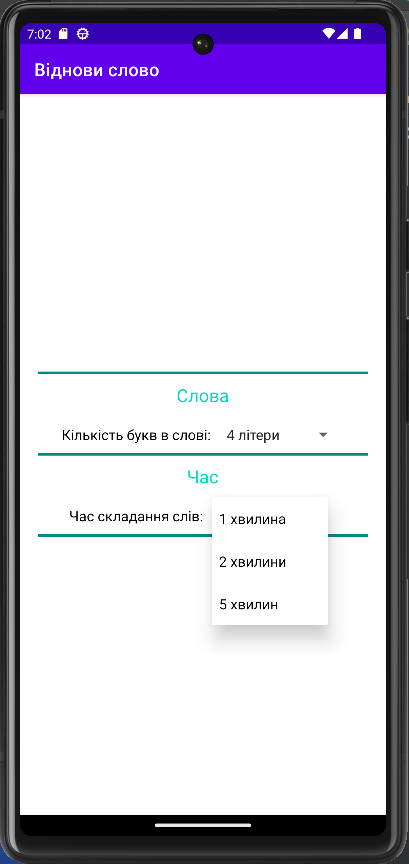


Рисунок 1.4 – діалог вибору часу.

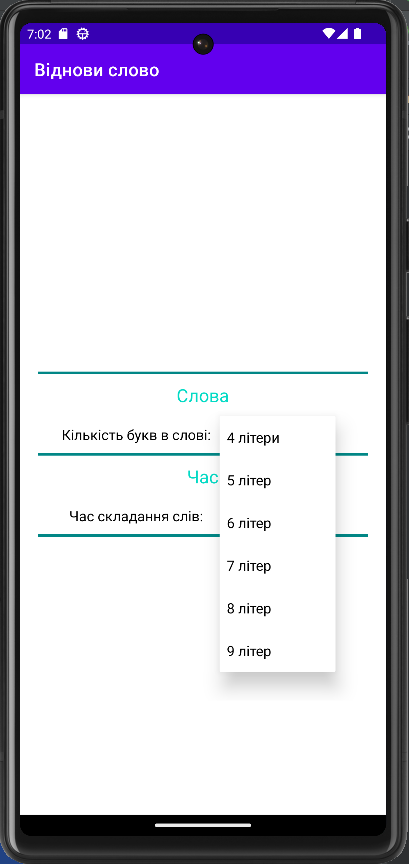


Рисунок 1.5 – діалог вибору кількості букв в слові

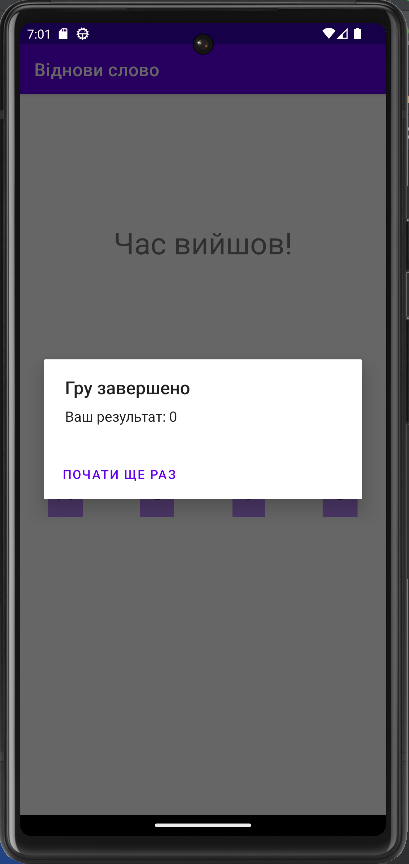


Рисунок 1.6 – діалог про закінчення гри.

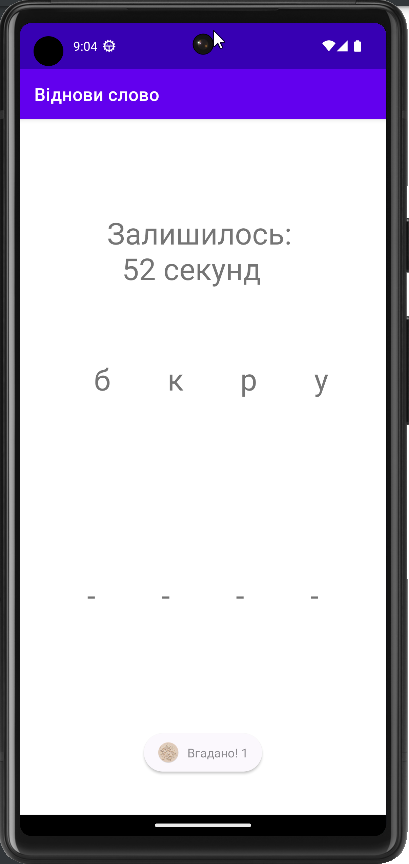


Рисунок 1.7 – Toast-повідомлення

## Опис виявлених проблем при розробці додатку та шляхи їх вирішення.

Основною проблемою в розробці додатку було те, що було не зрозуміло як зробити так щоб налаштування зберігалися, а не змінювалися при закритті програми. Цю проблему було вирішено створенням файлу налаштувань і запису туди потрібних даних.

Другорядною проблемою було те, що не виходило реалізувати сам алгоритм переміщення букв між даним словом та місцем правильного слова. Але цю проблему вдалося дуже швидко вирішити за допомогою доволі простої лінійно-функціонального програмування та Adapter RecyclerView.

## Висновки

На даній лабораторній роботі вдалося налаштувати середовище розробки додатків для операційної системи Android. Ознайомитися з типовою структурою Android-проекту з системою автоматичного збирання проектів Gradle. Навчилися розробляти прості Android-додатки з використанням стандартної бібліотеки візуальних компонентів, а також ознайомилися з Adapter RecyclerView.