# Exercise 2 - Keyboard application

R Abhinav

205001001

CSE-A

## Aim

To implement a keyboard application using Android Studio

## Code

### activity\_main.xml

<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout 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"

android:orientation="vertical"

tools:context=".MainActivity">

<TextView

android:id="@+id/text\_bar"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginLeft="6dp"

android:layout\_marginTop="24dp"

android:layout\_marginRight="6dp"

android:hint="Enter text..."

android:text=""

android:textSize="24sp" />

<GridLayout

android:id="@+id/keyboard"

android:layout\_width="match\_parent"

android:layout\_height="200dp"

android:layout\_alignParentBottom="true"

android:background="#202020"

android:padding="1dp" />

</RelativeLayout>

### MainActivity.java

package com.example.qwerty;

import androidx.appcompat.app.AppCompatActivity;

import androidx.core.content.res.ResourcesCompat;

import android.os.Bundle;

import android.util.DisplayMetrics;

import android.view.Gravity;

import android.widget.Button;

import android.widget.GridLayout;

import android.widget.TextView;

import android.graphics.Typeface;

public class MainActivity extends AppCompatActivity {

private final String[][] letterKeys = {

{"Q", "W", "E", "R", "T", "Y", "U", "I", "O", "P"},

{"A", "S", "D", "F", "G", "H", "J", "K", "L"},

{"↑", "Z", "X", "C", "V", "B", "N", "M", "←"},

{"?123", ",", " ", ".", "⮐"}

};

private final String[][] numberKeys = {

{"1", "2", "3", "4", "5", "6", "7", "8", "9", "0"},

{"@", "#", "₹", "\_", "&", "-", "+", "(", ")", "/"},

{"\*", "\"", "'", ":", ";", "!", "?", "<", ">", "=", "←"},

{"ABC", ",", " ", ".", "⮐"}

};

private TextView textBar;

private GridLayout keyboard;

private DisplayMetrics displayMetrics;

private Boolean isCaps = false;

private Typeface customFont;

private void init() {

textBar = findViewById(R.id.text\_bar);

textBar.setText("|");

keyboard = findViewById(R.id.keyboard);

displayMetrics = new DisplayMetrics();

getWindowManager().getDefaultDisplay().getMetrics(displayMetrics);

customFont = ResourcesCompat.getFont(this, R.font.oswald);

}

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

init();

initKeyboard(letterKeys);

}

private void initKeyboard(String[][] keys) {

keyboard.removeAllViews();

int rowIndex = 0;

for (String[] row : keys) {

int colIndex = 0;

int width = displayMetrics.widthPixels / (row.length + 4);

for (String letter : row) {

addButtonToKeyboard(letter, width, rowIndex, colIndex);

colIndex++;

}

rowIndex++;

}

}

private boolean isChar(String key) {

return key.length() == 1 && Character.isLetter(key.charAt(0));

}

private void addButtonToKeyboard(String key, int width, int row, int col) {

Button button = new Button(this);

if (isChar(key)) button.setText(key.toLowerCase());

else {

button.setText(key);

button.setTypeface(customFont);

button.setWidth(0);

button.setPadding(0, 0, 0, 0);

}

GridLayout.LayoutParams params = new GridLayout.LayoutParams();

params.width = width;

params.height = GridLayout.LayoutParams.WRAP\_CONTENT;

params.rowSpec = GridLayout.spec(row);

params.columnSpec = GridLayout.spec(col);

params.setGravity(Gravity.FILL);

button.setLayoutParams(params);

button.setClickable(true);

setHandler(button, key);

keyboard.addView(button);

}

private void setHandler(Button button, String key) {

if (key.length() == 1 && Character.isLetter(key.charAt(0))) {

button.setOnClickListener(view -> {

if (isCaps) textBar.append(key.toUpperCase());

else textBar.append(key.toLowerCase());

});

return;

}

switch (key) {

case "↑":

button.setOnClickListener(view -> isCaps = !isCaps);

break;

case "←":

button.setOnClickListener(view -> {

CharSequence currentText = textBar.getText();

if(currentText.length()!=0) {

CharSequence newText = currentText.subSequence(0, currentText.length() - 1);

textBar.setText(newText);

}

});

break;

case "?123":

button.setOnClickListener(view -> initKeyboard(numberKeys));

break;

case "ABC":

button.setOnClickListener(view -> initKeyboard(letterKeys));

break;

case "⮐":

button.setOnClickListener(view -> textBar.append("\n"));

break;

default:

button.setOnClickListener(view -> {

CharSequence currentText = textBar.getText();

CharSequence newText = currentText.subSequence(0, currentText.length() - 1);

textBar.setText(newText);

textBar.append(String.valueOf(key.charAt(0)));

textBar.append("|");

}

);

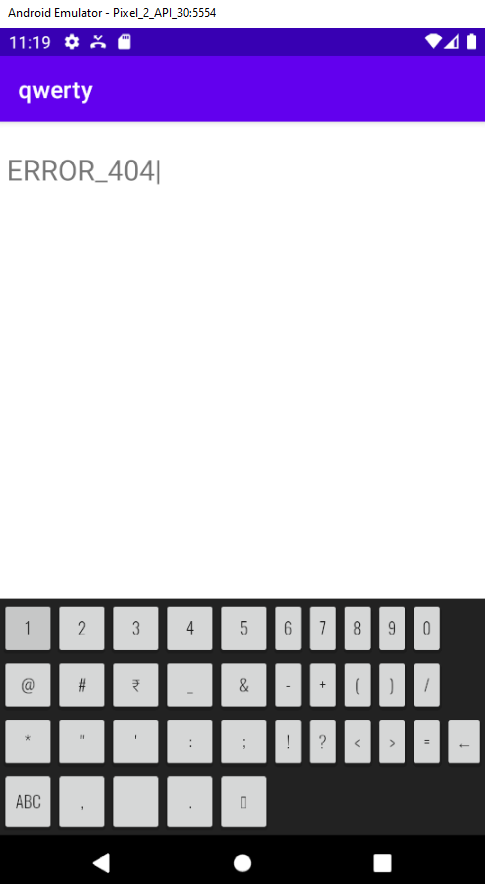
break;

}

}

}

## Output



## Learning outcomes

Thus a keyboard was implemented using Android Studio