MainActivity.java

package com.example.myapplication;  
  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.constraintlayout.widget.ConstraintLayout;  
  
import android.annotation.SuppressLint;  
import android.os.AsyncTask;  
import android.os.Bundle;  
import android.util.Log;  
import android.view.MotionEvent;  
import android.view.View;  
import android.widget.ImageButton;  
import android.widget.TextView;  
  
import com.google.firebase.database.DataSnapshot;  
import com.google.firebase.database.DatabaseError;  
import com.google.firebase.database.DatabaseReference;  
import com.google.firebase.database.FirebaseDatabase;  
import com.google.firebase.database.ValueEventListener;  
  
public class MainActivity extends AppCompatActivity {  
  
 ImageButton forward\_button, backward\_button, right\_button, left\_button;  
 TextView throttle, acceleration, direction;  
 boolean isForward = false, isBackward = false, isLeft = false, isRight = false;  
 Integer curr\_speed = 0;  
 FirebaseDatabase database = FirebaseDatabase.*getInstance*();  
 DatabaseReference commandRef = database.getReference("command");  
 DatabaseReference speedref = database.getReference("speed");  
 DatabaseReference accref = database.getReference("acceleration");  
  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 forward\_button = findViewById(R.id.*forward\_button*);  
 backward\_button = findViewById(R.id.*backward\_button*);  
 left\_button = findViewById(R.id.*left\_button*);  
 right\_button = findViewById(R.id.*right\_button*);  
  
  
 throttle = findViewById(R.id.*throttle*);  
 acceleration = findViewById(R.id.*acceleration*);  
 direction = findViewById(R.id.*direction*);  
  
 speedref.setValue(curr\_speed);  
  
  
 //setting onTouchListener on buttons  
 forward\_button.setOnTouchListener(new View.OnTouchListener(){  
 @Override  
 public boolean onTouch(View v, MotionEvent event){  
 if(event.getAction() == MotionEvent.*ACTION\_DOWN*){  
 isForward = true;  
 ConstraintLayout.LayoutParams params = (ConstraintLayout.LayoutParams)forward\_button.getLayoutParams();  
 final float scale = forward\_button.getContext().getResources().getDisplayMetrics().density;  
 params.width = (int)(75\* scale + 0.5f);  
 params.height = (int)(135\* scale + 0.5f);  
 new SendVolumeUpTask().execute('F');  
 forward\_button.setLayoutParams(params);  
 return true;  
 }  
 if(event.getAction() == MotionEvent.*ACTION\_UP*){  
 isForward = false;  
 ConstraintLayout.LayoutParams params = (ConstraintLayout.LayoutParams)forward\_button.getLayoutParams();  
 final float scale = forward\_button.getContext().getResources().getDisplayMetrics().density;  
 params.width = (int)(80\* scale + 0.5f);  
 params.height = (int)(142\* scale + 0.5f);  
  
 forward\_button.setLayoutParams(params);  
 return true;  
 }  
 return false;  
 }  
 });  
  
  
 backward\_button.setOnTouchListener(new View.OnTouchListener(){  
 @Override  
 public boolean onTouch(View v, MotionEvent event){  
 if(event.getAction() == MotionEvent.*ACTION\_DOWN*){  
 isBackward = true;  
 ConstraintLayout.LayoutParams params = (ConstraintLayout.LayoutParams)backward\_button.getLayoutParams();  
 final float scale = backward\_button.getContext().getResources().getDisplayMetrics().density;  
 params.width = (int)(95\* scale + 0.5f);  
 params.height = (int)(78\* scale + 0.5f);  
 new SendVolumeUpTask().execute('B');  
 backward\_button.setLayoutParams(params);  
 return true;  
 }  
 if(event.getAction() == MotionEvent.*ACTION\_UP*){  
 isBackward = false;  
 ConstraintLayout.LayoutParams params = (ConstraintLayout.LayoutParams)backward\_button.getLayoutParams();  
 final float scale = backward\_button.getContext().getResources().getDisplayMetrics().density;  
 params.width = (int)(100\* scale + 0.5f);  
 params.height = (int)(84\* scale + 0.5f);  
  
 backward\_button.setLayoutParams(params);  
 return true;  
 }  
 return false;  
 }  
 });  
  
 left\_button.setOnTouchListener(new View.OnTouchListener(){  
 @Override  
 public boolean onTouch(View v, MotionEvent event){  
 if(event.getAction() == MotionEvent.*ACTION\_DOWN*){  
 isLeft = true;  
 direction.setText("Left");  
 ConstraintLayout.LayoutParams params = (ConstraintLayout.LayoutParams)left\_button.getLayoutParams();  
 final float scale = forward\_button.getContext().getResources().getDisplayMetrics().density;  
 params.width = (int)(95\* scale + 0.5f);  
 params.height = (int)(95\* scale + 0.5f);  
 new SendVolumeUpTask().execute('L');  
 left\_button.setLayoutParams(params);  
 return true;  
 }  
 if(event.getAction() == MotionEvent.*ACTION\_UP*){  
 isLeft = false;  
 direction.setText("");  
 ConstraintLayout.LayoutParams params = (ConstraintLayout.LayoutParams)left\_button.getLayoutParams();  
 final float scale = left\_button.getContext().getResources().getDisplayMetrics().density;  
 params.width = (int)(100\* scale + 0.5f);  
 params.height = (int)(100\* scale + 0.5f);  
  
 left\_button.setLayoutParams(params);  
 return true;  
 }  
 return false;  
 }  
 });  
  
 right\_button.setOnTouchListener(new View.OnTouchListener(){  
 @Override  
 public boolean onTouch(View v, MotionEvent event){  
 if(event.getAction() == MotionEvent.*ACTION\_DOWN*){  
 isRight= true;  
 direction.setText("Right");  
 ConstraintLayout.LayoutParams params = (ConstraintLayout.LayoutParams)right\_button.getLayoutParams();  
 final float scale = right\_button.getContext().getResources().getDisplayMetrics().density;  
 params.width = (int)(95\* scale + 0.5f);  
 params.height = (int)(95\* scale + 0.5f);  
 new SendVolumeUpTask().execute('R');  
 right\_button.setLayoutParams(params);  
 return true;  
 }  
 if(event.getAction() == MotionEvent.*ACTION\_UP*){  
 isRight = false;  
 direction.setText("");  
 ConstraintLayout.LayoutParams params = (ConstraintLayout.LayoutParams)right\_button.getLayoutParams();  
 final float scale = right\_button.getContext().getResources().getDisplayMetrics().density;  
 params.width = (int)(100\* scale + 0.5f);  
 params.height = (int)(100\* scale + 0.5f);  
  
 right\_button.setLayoutParams(params);  
 return true;  
 }  
 return false;  
 }  
 });  
  
 ValueEventListener accListener = new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot snapshot) {  
 double acc = 0;  
 acc = snapshot.getValue(double.class);  
 acc = round(acc, 2);  
 acceleration.setText("acc: " + String.*valueOf*(acc) + "m/s2");  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError error) {  
 //empty  
 }  
  
 private double round(double value, int places) {  
 if(places < 0) throw new IllegalArgumentException();  
 long factor = (long)Math.*pow*(10, places);  
 value= value\*factor;  
 long tmp = Math.*round*(value);  
 return (double) tmp/factor;  
 }  
 };  
  
 accref.addValueEventListener(accListener);  
 }  
  
  
 class SendVolumeUpTask extends AsyncTask<Character, Integer, Void>{  
  
 @Override  
 protected Void doInBackground(Character... command){  
 long curr\_millis = System.*currentTimeMillis*();  
 long prev\_millis = 0;  
  
 while(isForward || isBackward || isLeft || isRight){  
 curr\_millis = System.*currentTimeMillis*();  
  
 if(curr\_millis - prev\_millis >= 15){  
 updateDirection(command[0]);  
  
 if(curr\_speed < 1025 && curr\_speed >= 0 && isForward)  
 curr\_speed += 5;  
 if(curr\_speed > -1025 && curr\_speed < 0 && isBackward)  
 curr\_speed -= 5;  
 if(curr\_speed <= 1025 && curr\_speed >= 0 && isBackward)  
 curr\_speed -= 5;  
 if(curr\_speed >= -1025 && curr\_speed < 0 && isForward)  
 curr\_speed += 5;  
  
 prev\_millis = curr\_millis;  
  
 int throttle\_percent = (int)(((double)curr\_speed/1025)\*100);  
 publishProgress(throttle\_percent);  
 updateSpeed();  
  
 }  
 }  
 return null;  
 }  
  
 @Override  
 protected void onProgressUpdate(Integer... values){  
 throttle.setText("throttle: " + String.*valueOf*(values[0]) + "%");  
 }  
 private void updateSpeed() {  
 speedref.setValue(curr\_speed);  
 }  
  
 private void updateDirection(Character command) {  
 Log.*e*("msg", String.*valueOf*(command));  
 commandRef.setValue(String.*valueOf*(command));  
 }  
 }  
  
}