MY ACTIVITY .JAVA

package com.example.autonomouscar;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.bluetooth.BluetoothAdapter;  
import android.bluetooth.BluetoothDevice;  
import android.content.Context;  
import android.content.Intent;  
import android.content.SharedPreferences;  
import android.content.pm.ActivityInfo;  
import android.graphics.Color;  
import android.os.AsyncTask;  
import android.os.Bundle;  
import android.preference.PreferenceManager;  
import android.util.Log;  
import android.view.LayoutInflater;  
import android.view.Menu;  
import android.view.MenuItem;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.AdapterView;  
import android.widget.ArrayAdapter;  
import android.widget.Button;  
import android.widget.ListView;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Set;  
import java.util.UUID;  
  
public class MainActivity extends AppCompatActivity {  
  
 private Button search;  
 private Button connect;  
 private ListView listView;  
 private BluetoothAdapter mBTAdapter;  
 private static final int *BT\_ENABLE\_REQUEST* = 10; // This is the code we use for BT Enable  
 private static final int *SETTINGS* = 20;  
 private UUID mDeviceUUID = UUID.*fromString*("00001101-0000-1000-8000-00805F9B34FB");  
 private int mBufferSize = 50000; //Default  
 public static final String *DEVICE\_EXTRA* = "com.example.lightcontrol.SOCKET";  
 public static final String *DEVICE\_UUID* = "com.example.lightcontrol.uuid";  
 private static final String *DEVICE\_LIST* = "com.example.lightcontrol.devicelist";  
 private static final String *DEVICE\_LIST\_SELECTED* = "com.example.lightcontrol.devicelistselected";  
 public static final String *BUFFER\_SIZE* = "com.example.lightcontrol.buffersize";  
 private static final String *TAG* = "BlueTest5-MainActivity";  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
 search = (Button) findViewById(R.id.*search*);  
 connect = (Button) findViewById(R.id.*connect*);  
 listView = (ListView) findViewById(R.id.*listView*);  
  
 if (savedInstanceState != null) {  
 ArrayList<BluetoothDevice> list = savedInstanceState.getParcelableArrayList(*DEVICE\_LIST*);  
 if (list != null) {  
 initList(list);  
 MyAdapter adapter = (MyAdapter) listView.getAdapter();  
 int selectedIndex = savedInstanceState.getInt(*DEVICE\_LIST\_SELECTED*);  
 if (selectedIndex != -1) {  
 adapter.setSelectedIndex(selectedIndex);  
 connect.setEnabled(true);  
 }  
 } else {  
 initList(new ArrayList<BluetoothDevice>());  
 }  
  
 } else {  
 initList(new ArrayList<BluetoothDevice>());  
 }  
  
 search.setOnClickListener(new View.OnClickListener() {  
  
 @Override  
 public void onClick(View arg0) {  
 mBTAdapter = BluetoothAdapter.*getDefaultAdapter*();  
  
 if (mBTAdapter == null) {  
 Toast.*makeText*(getApplicationContext(), "Bluetooth not found", Toast.*LENGTH\_SHORT*).show();  
 } else if (!mBTAdapter.isEnabled()) {  
 Intent enableBT = new Intent(BluetoothAdapter.*ACTION\_REQUEST\_ENABLE*);  
 startActivityForResult(enableBT, *BT\_ENABLE\_REQUEST*);  
 } else {  
 new SearchDevices().execute();  
 }  
 }  
 });  
  
 connect.setOnClickListener(new View.OnClickListener() {  
  
 @Override  
 public void onClick(View arg0) {  
 BluetoothDevice device = ((MyAdapter) (listView.getAdapter())).getSelectedItem();  
 Intent intent = new Intent(getApplicationContext(), Controlling.class);  
 intent.putExtra(*DEVICE\_EXTRA*, device);  
 intent.putExtra(*DEVICE\_UUID*, mDeviceUUID.toString());  
 intent.putExtra(*BUFFER\_SIZE*, mBufferSize);  
 startActivity(intent);  
 }  
 });  
  
 }  
  
 protected void onPause() {  
// *TODO Auto-generated method stub* super.onPause();  
 }  
  
 @Override  
 protected void onStop() {  
// *TODO Auto-generated method stub* super.onStop();  
 }  
  
 protected void onActivityResult(int requestCode, int resultCode, Intent data) {  
 switch (requestCode) {  
 case *BT\_ENABLE\_REQUEST*:  
 if (resultCode == *RESULT\_OK*) {  
 msg("Bluetooth Enabled successfully");  
 new SearchDevices().execute();  
 } else {  
 msg("Bluetooth couldn't be enabled");  
 }  
  
 break;  
 case *SETTINGS*: //If the settings have been updated  
 SharedPreferences prefs = PreferenceManager.*getDefaultSharedPreferences*(this);  
 String uuid = prefs.getString("prefUuid", "Null");  
 mDeviceUUID = UUID.*fromString*(uuid);  
 Log.*d*(*TAG*, "UUID: " + uuid);  
 String bufSize = prefs.getString("prefTextBuffer", "Null");  
 mBufferSize = Integer.*parseInt*(bufSize);  
  
 String orientation = prefs.getString("prefOrientation", "Null");  
 Log.*d*(*TAG*, "Orientation: " + orientation);  
 if (orientation.equals("Landscape")) {  
 setRequestedOrientation(ActivityInfo.*SCREEN\_ORIENTATION\_LANDSCAPE*);  
 } else if (orientation.equals("Portrait")) {  
 setRequestedOrientation(ActivityInfo.*SCREEN\_ORIENTATION\_PORTRAIT*);  
 } else if (orientation.equals("Auto")) {  
 setRequestedOrientation(ActivityInfo.*SCREEN\_ORIENTATION\_FULL\_SENSOR*);  
 }  
 break;  
 default:  
 break;  
 }  
 super.onActivityResult(requestCode, resultCode, data);  
 }  
  
 private void msg(String str) {  
 Toast.*makeText*(getApplicationContext(), str, Toast.*LENGTH\_SHORT*).show();  
 }  
  
 */\*\*  
 \* Initialize the List adapter  
 \** ***@param*** *objects  
 \*/* private void initList(List<BluetoothDevice> objects) {  
 final MyAdapter adapter = new MyAdapter(getApplicationContext(), R.layout.*list\_item*, R.id.*lstContent*, objects);  
 listView.setAdapter(adapter);  
 listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {  
  
 @Override  
 public void onItemClick(AdapterView<?> parent, View view, int position, long id) {  
 adapter.setSelectedIndex(position);  
 connect.setEnabled(true);  
 }  
 });  
 }  
  
 private class SearchDevices extends AsyncTask<Void, Void, List<BluetoothDevice>> {  
  
 @Override  
 protected List<BluetoothDevice> doInBackground(Void... params) {  
 Set<BluetoothDevice> pairedDevices = mBTAdapter.getBondedDevices();  
 List<BluetoothDevice> listDevices = new ArrayList<BluetoothDevice>();  
 for (BluetoothDevice device : pairedDevices) {  
 listDevices.add(device);  
 }  
 return listDevices;  
  
 }  
  
 @Override  
 protected void onPostExecute(List<BluetoothDevice> listDevices) {  
 super.onPostExecute(listDevices);  
 if (listDevices.size() > 0) {  
 MyAdapter adapter = (MyAdapter) listView.getAdapter();  
 adapter.replaceItems(listDevices);  
 } else {  
 msg("No paired devices found, please pair your serial BT device and try again");  
 }  
 }  
 }  
  
 private class MyAdapter extends ArrayAdapter<BluetoothDevice> {  
 private int selectedIndex;  
 private Context context;  
 private int selectedColor = Color.*parseColor*("#abcdef");  
 private List<BluetoothDevice> myList;  
 public MyAdapter(Context ctx, int resource, int textViewResourceId, List<BluetoothDevice> objects) {  
 super(ctx, resource, textViewResourceId, objects);  
 context = ctx;  
 myList = objects;  
 selectedIndex = -1;  
 }  
  
 public void setSelectedIndex(int position) {  
 selectedIndex = position;  
 notifyDataSetChanged();  
 }  
  
 public BluetoothDevice getSelectedItem() {  
 return myList.get(selectedIndex);  
 }  
  
 public int getCount() {  
 return myList.size();  
 }  
  
 @Override  
 public BluetoothDevice getItem(int position) {  
 return myList.get(position);  
 }  
  
 @Override  
 public long getItemId(int position) {  
 return position;  
 }  
  
 private class ViewHolder {  
 TextView tv;  
 }  
  
 public void replaceItems(List<BluetoothDevice> list) {  
 myList = list;  
 notifyDataSetChanged();  
 }  
  
 public List<BluetoothDevice> getEntireList() {  
 return myList;  
 }  
  
 @Override  
 public View getView(int position, View convertView, ViewGroup parent) {  
 View vi = convertView;  
 ViewHolder holder;  
 if (convertView == null) {  
 vi = LayoutInflater.*from*(context).inflate(R.layout.*list\_item*, null);  
 holder = new ViewHolder();  
  
 holder.tv = (TextView) vi.findViewById(R.id.*lstContent*);  
  
 vi.setTag(holder);  
 } else {  
 holder = (ViewHolder) vi.getTag();  
 }  
  
 if (selectedIndex != -1 && position == selectedIndex) {  
 holder.tv.setBackgroundColor(selectedColor);  
 } else {  
 holder.tv.setBackgroundColor(Color.*WHITE*);  
 }  
 BluetoothDevice device = myList.get(position);  
 holder.tv.setText(device.getName() + "\n " + device.getAddress());  
  
 return vi;  
 }  
  
 }  
  
  
  
}

ACTIVITY.XML

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout 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:orientation="vertical"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".MainActivity"  
 tools:ignore="ExtraText">  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center"  
 android:layout\_marginTop="20dp">  
  
 <Button  
 android:id="@+id/search"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="match\_parent"  
 android:layout\_marginLeft="20pt"  
  
 android:text="Search" />  
  
 <Button  
 android:id="@+id/connect"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="match\_parent"  
 android:layout\_marginLeft="50pt"  
 android:text="Connect" />  
  
 </LinearLayout>  
 <ListView  
 android:id="@+id/listView"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent">  
  
 </ListView>  
  
</LinearLayout>

CONTROLLING.JAVA

package com.example.autonomouscar;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.app.ProgressDialog;  
import android.bluetooth.BluetoothAdapter;  
import android.bluetooth.BluetoothDevice;  
import android.bluetooth.BluetoothSocket;  
import android.content.Intent;  
import android.os.AsyncTask;  
import android.os.Bundle;  
import android.util.Log;  
import android.view.View;  
import android.widget.Button;  
import android.widget.Toast;  
  
import java.io.IOException;  
import java.io.InputStream;  
import java.nio.charset.StandardCharsets;  
import java.util.UUID;  
  
public class Controlling extends AppCompatActivity {  
  
 private static final String *TAG* = "BlueTest5-Controlling";  
 private int mMaxChars = 50000;//Default//change this to string..........  
 private UUID mDeviceUUID;  
 private BluetoothSocket mBTSocket;  
 private ReadInput mReadThread = null;  
  
 private boolean mIsUserInitiatedDisconnect = false;  
 private boolean mIsBluetoothConnected = false;  
  
  
 private Button mBtnDisconnect;  
 private BluetoothDevice mDevice;  
  
 final static String *Obsticle\_Avoiding*="1";//Obsticle\_Avoiding  
 final static String *Bluetooth*="2";//Bluetooth  
 final static String *IR\_Remote*="3";//IR\_Remote  
 final static String *Line\_Following*="4";//Line\_Following  
  
 private ProgressDialog progressDialog;  
 Button btn1,btn2,btn3,btn4;  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_controlling*);  
  
 ActivityHelper.*initialize*(this);  
 // mBtnDisconnect = (Button) findViewById(R.id.btnDisconnect);  
 btn1=(Button)findViewById(R.id.*Obsticle\_Avoiding*);  
 btn2=(Button)findViewById(R.id.*Bluetooth*);  
 btn3=(Button)findViewById(R.id.*IR\_Remote*);  
 btn4=(Button)findViewById(R.id.*Line\_Following*);  
  
 Intent intent = getIntent();  
 Bundle b = intent.getExtras();  
 mDevice = b.getParcelable(MainActivity.*DEVICE\_EXTRA*);  
 mDeviceUUID = UUID.*fromString*(b.getString(MainActivity.*DEVICE\_UUID*));  
 mMaxChars = b.getInt(MainActivity.*BUFFER\_SIZE*);  
  
 Log.*d*(*TAG*, "Ready");  
  
 btn1.setOnClickListener(new View.OnClickListener()  
 {  
  
 @Override  
 public void onClick(View v) {  
// *TODO Auto-generated method stub* try {  
 mBTSocket.getOutputStream().write(*Obsticle\_Avoiding*.getBytes());  
  
 } catch (IOException e) {  
 // *TODO Auto-generated catch block* e.printStackTrace();  
 }  
 }});  
  
 btn2.setOnClickListener(new View.OnClickListener()  
 {  
  
 @Override  
 public void onClick(View v) {  
 try {  
 mBTSocket.getOutputStream().write(*Bluetooth*.getBytes());  
 } catch (IOException e) {  
 // *TODO Auto-generated catch block* e.printStackTrace();  
 }  
  
  
 }});  
  
  
 btn3.setOnClickListener(new View.OnClickListener()  
 {  
 @Override  
 public void onClick(View v) {  
 try {  
 mBTSocket.getOutputStream().write(*IR\_Remote*.getBytes());  
 }catch (IOException e){  
 e.printStackTrace();  
 }  
 }  
 });  
  
 btn4.setOnClickListener(new View.OnClickListener()  
 {  
 @Override  
 public void onClick(View v) {  
 try {  
 mBTSocket.getOutputStream().write(*Line\_Following*.getBytes());  
 }catch (IOException e){  
 e.printStackTrace();  
 }  
 }  
 });  
  
  
 }  
  
 private class ReadInput implements Runnable {  
  
 private boolean bStop = false;  
 private Thread t;  
  
 public ReadInput() {  
 t = new Thread(this, "Input Thread");  
 t.start();  
 }  
  
 public boolean isRunning() {  
 return t.isAlive();  
 }  
  
 @Override  
 public void run() {  
 InputStream inputStream;  
  
 try {  
 inputStream = mBTSocket.getInputStream();  
 while (!bStop) {  
 byte[] buffer = new byte[256];  
 if (inputStream.available() > 0) {  
 inputStream.read(buffer);  
 int i = 0;  
 /\*  
 \* This is needed because new String(buffer) is taking the entire buffer i.e. 256 chars on Android 2.3.4 http://stackoverflow.com/a/8843462/1287554  
 \*/  
 for (i = 0; i < buffer.length && buffer[i] != 0; i++) {  
 }  
 final String strInput = new String(buffer, 0, i);  
  
 /\*  
 \* If checked then receive text, better design would probably be to stop thread if unchecked and free resources, but this is a quick fix  
 \*/  
  
  
  
 }  
 Thread.*sleep*(500);  
 }  
 } catch (IOException e) {  
// *TODO Auto-generated catch block* e.printStackTrace();  
 } catch (InterruptedException e) {  
// *TODO Auto-generated catch block* e.printStackTrace();  
 }  
  
 }  
  
 public void stop() {  
 bStop = true;  
 }  
 }  
  
 private class DisConnectBT extends AsyncTask<Void, Void, Void> {  
  
 @Override  
 protected void onPreExecute() {  
 }  
  
 @Override  
 protected Void doInBackground(Void... params) {//cant inderstand these dotss  
  
 if (mReadThread != null) {  
 mReadThread.stop();  
 while (mReadThread.isRunning())  
 ; // Wait until it stops  
 mReadThread = null;  
  
 }  
  
 try {  
 mBTSocket.close();  
 } catch (IOException e) {  
// *TODO Auto-generated catch block* e.printStackTrace();  
 }  
  
 return null;  
 }  
  
 @Override  
 protected void onPostExecute(Void result) {  
 super.onPostExecute(result);  
 mIsBluetoothConnected = false;  
 if (mIsUserInitiatedDisconnect) {  
 finish();  
 }  
 }  
  
 }  
  
 private void msg(String s) {  
 Toast.*makeText*(getApplicationContext(), s, Toast.*LENGTH\_SHORT*).show();  
 }  
  
 @Override  
 protected void onPause() {  
 if (mBTSocket != null && mIsBluetoothConnected) {  
 new DisConnectBT().execute();  
 }  
 Log.*d*(*TAG*, "Paused");  
 super.onPause();  
 }  
  
 @Override  
 protected void onResume() {  
 if (mBTSocket == null || !mIsBluetoothConnected) {  
 new ConnectBT().execute();  
 }  
 Log.*d*(*TAG*, "Resumed");  
 super.onResume();  
 }  
  
 @Override  
 protected void onStop() {  
 Log.*d*(*TAG*, "Stopped");  
 super.onStop();  
 }  
  
 private class ConnectBT extends AsyncTask<Void, Void, Void> {  
 private boolean mConnectSuccessful = true;  
  
 @Override  
 protected void onPreExecute() {  
  
 progressDialog = ProgressDialog.*show*(Controlling.this, "Hold on", "Connecting");// http://stackoverflow.com/a/11130220/1287554  
  
 }  
  
 @Override  
 protected Void doInBackground(Void... devices) {  
  
 try {  
 if (mBTSocket == null || !mIsBluetoothConnected) {  
 mBTSocket = mDevice.createInsecureRfcommSocketToServiceRecord(mDeviceUUID);  
 BluetoothAdapter.*getDefaultAdapter*().cancelDiscovery();  
 mBTSocket.connect();  
 }  
 } catch (IOException e) {  
// Unable to connect to device`  
 // e.printStackTrace();  
 mConnectSuccessful = false;  
  
  
  
 }  
 return null;  
 }  
  
 @Override  
 protected void onPostExecute(Void result) {  
 super.onPostExecute(result);  
  
 if (!mConnectSuccessful) {  
 Toast.*makeText*(getApplicationContext(), "Could not connect to device.Please turn on your Hardware", Toast.*LENGTH\_LONG*).show();  
 finish();  
 } else {  
 msg("Connected to device");  
 mIsBluetoothConnected = true;  
 mReadThread = new ReadInput(); // Kick off input reader  
 }  
  
 progressDialog.dismiss();  
 }  
  
 }  
  
 protected void onDestroy() {  
 // *TODO Auto-generated method stub* super.onDestroy();  
 }  
}

CONTROLLING. XML\

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout 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=".Controlling">  
 <TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"/>  
 <Button  
 android:id="@+id/Obsticle\_Avoiding"  
 android:layout\_width="150dp"  
 android:layout\_height="100dp"  
 android:text="Obsticle Avoiding"  
 android:layout\_marginTop="100dp"  
 android:layout\_marginLeft="100dp"/>  
  
 <Button  
 android:id="@+id/Bluetooth"  
 android:layout\_width="150dp"  
 android:layout\_height="100dp"  
 android:text="Hand following"  
 android:layout\_marginTop="250dp"  
 android:layout\_marginLeft="100dp"/>  
  
 <Button  
 android:id="@+id/IR\_Remote"  
 android:layout\_width="150dp"  
 android:layout\_height="100dp"  
 android:text="Line Following"  
 android:layout\_marginTop="400dp"  
 android:layout\_marginLeft="100dp"/>  
  
 <Button  
 android:id="@+id/Line\_Following"  
 android:layout\_width="150dp"  
 android:layout\_height="100dp"  
 android:layout\_marginLeft="100dp"  
 android:layout\_marginTop="550dp"  
 android:text="IR remote Control" />  
  
</RelativeLayout>

MANIFEST

<?xml version="1.0" encoding="utf-8"?>  
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
 package="com.example.autonomouscar">  
  
 <uses-permission android:name="android.permission.BLUETOOTH" />  
  
 <uses-permission android:name="android.permission.BLUETOOTH\_ADMIN" />  
  
 <application  
 android:allowBackup="true"  
 android:icon="@mipmap/ic\_launcher"  
 android:label="@string/app\_name"  
 android:roundIcon="@mipmap/ic\_launcher\_round"  
 android:supportsRtl="true"  
 android:theme="@style/Theme.AutonomousCar">  
 <activity  
 android:name=".Controlling"  
 android:exported="false" />  
 <activity  
 android:name=".MainActivity"  
 android:exported="true">  
 <intent-filter>  
 <action android:name="android.intent.action.MAIN" />  
  
 <category android:name="android.intent.category.LAUNCHER" />  
 </intent-filter>  
 </activity>  
 </application>  
  
</manifest>