

Android alapú szoftverfejlesztés

5. Labor



Játékkalkalmazás fejlesztése Android platformon egyedi felületi elemmel

Tartalom

1	Felkészülés a laborra.....	2
2	Saját nézet létrehozása	2
3	Laborfeladatok	5

1 Felkészülés a laborra

A labor célja, hogy bemutassa egy egyszerű játék alkalmazás fejlesztését és egyedi vezérlők használatát.

2 Saját nézet létrehozása

A laborfeladat megvalósításához szükséges egy saját nézet létrehozása. Az Android erre nagyon jó lehetőséget biztosít. A feladatunk csupán annyi, hogy egy saját osztályt hozzunk létre, amely a *View* osztályból származik le és az *onDraw(...)* függvényt felüldefiniálva megadhatjuk, hogy hogyan rajzolja ki magát a nézet. Ezen kívül az *onMeasure(...)* függvény felüldefiniálásával megadhatjuk, hogy a nézet milyen méretben jelenjen meg (például kikényszeríthetjük a négyzetes megjelenést), valamint az *onTouchEvent(...)* felüldefiniálásával az érintés eseményt kezelhetjük a saját nézetünkön. Amennyiben az eseményt szeretnénk átadni egy másik osztálynak, vagy például a nézetünket tartalmazó Activity-nek, készíthetünk egy saját interface-t, amelyet az Activity implementál és ezen keresztül kommunikálhatunk vele.

A következőkben nézzünk egy egyszerű példát, ahol a saját nézetünk kirajzol egy háttérképet erőforrásból, valamint egy átlós vonalat jelenít meg, illetve kikényszeríti a négyzetes megjelenítést, valamint az érintés eseményt átküldi a tartalmazó Activity-nek, aki a LogCat-ba kiírja a kapott koordinátákat.

A saját nézetet megvalósító osztály:

```
public class GameView extends View {

    public interface ITouchListener {
        abstract void onTouchSelected();
    }

    private Drawable mDrawableBg;
    private int touchX;
    private int touchY;
    private ITouchListener touchListener = null;

    public GameView(Context context, AttributeSet attrs) {
        super(context, attrs);
        requestFocus();

        mDrawableBg =
            getResources().getDrawable(R.drawable.ic_launcher);
        setBackgroundDrawable(mDrawableBg);
    }

    public void setTouchListener(ITouchListener aTouchListener) {
        touchListener = aTouchListener;
    }
}
```

```
private void touchSelected() {
    if (touchListener != null)
        touchListener.onTouchSelected();
}

public int getTouchX() {
    return touchX;
}

public int getTouchY() {
    return touchY;
}

@Override
protected void onDraw(Canvas canvas) {
    super.onDraw(canvas);

    Paint mLinePaint = new Paint();
    mLinePaint.setColor(0xFFFFFFFF);
    mLinePaint.setStrokeWidth(5);
    mLinePaint.setStyle(Style.STROKE);

    canvas.drawLine(0, 0, getWidth(), getHeight(), mLinePaint);
}

@Override
protected void onMeasure(int widthMeasureSpec,
    int heightMeasureSpec) {
    // Négyzetes felület kikényszerítése
    int w = MeasureSpec.getSize(widthMeasureSpec);
    int h = MeasureSpec.getSize(heightMeasureSpec);
    int d = w == 0 ? h : h == 0 ? w : w < h ? w : h;
    setMeasuredDimension(d, d);
}

@Override
public boolean onTouchEvent(MotionEvent event) {
    int action = event.getAction();

    if (action == MotionEvent.ACTION_DOWN) {
        return true;
    } else if (action == MotionEvent.ACTION_UP) {
        touchX = (int) event.getX();
        touchY = (int) event.getY();
        touchSelected();
        return true;
    } else {
        // Nem kezeltük le az érintés eseményt
        return false;
    }
}
}
```

Az alkalmazás felhasználói felületét megadó XML (figyeljük meg, hogy hivatkozunk csomag és osztálynévvel XML-ben a saját felületi elemre):

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >
    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/hello" />
    <hu.bute.daai.amorg.examples.GameView
        android:id="@+id/game_view"
        android:layout_width="300dp"
        android:layout_height="300dp"
        />
</LinearLayout>
```

Végül az Activity-t megvalósító osztály:

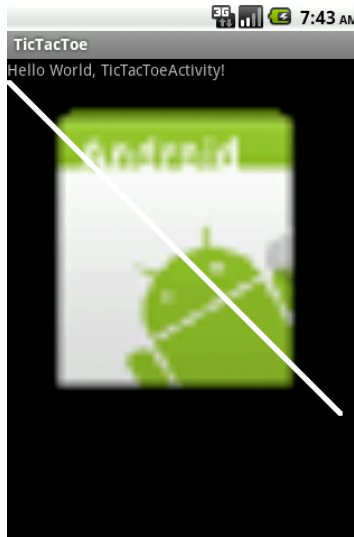
```
public class TicTacToeActivity extends Activity implements
ITouchListener {

    private GameView gv;

    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        gv = (GameView) findViewById(R.id.game_view);
        gv.setTouchListener(this);
        gv.setFocusable(true);
        gv.setFocusableInTouchMode(true);
    }

    @Override
    public void onTouchSelected() {
        Log.d(getClass().getName(), gv.getTouchX() + "," +
gv.getTouchY());
    }
}
```

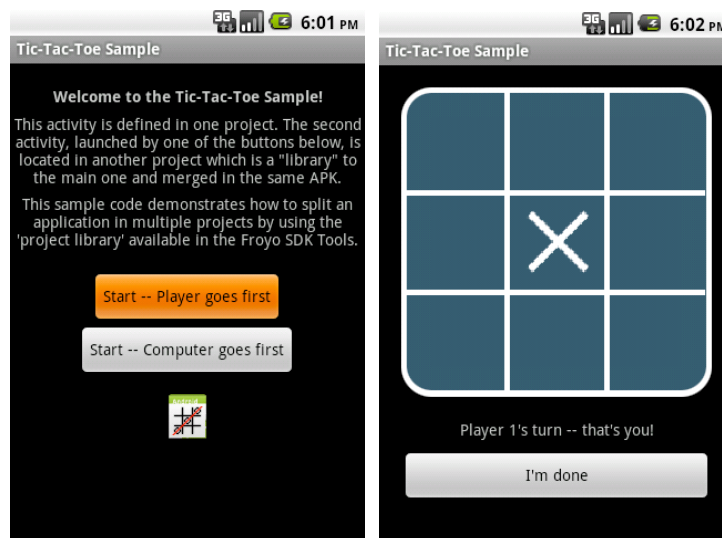


2.1. ábra - Saját nézet példa

Az újrarajzolás kikényszerítése a saját *View* osztályon belül az *invalidate()* függvény meghívásával érhető el.

3 Laborfeladatok

A labor során a feladat egy egyszerű Tic-Tac-Toe (3x3-as amőba) játék elkészítése. Az alkalmazás indulásakor a program egy menüből tudja elindítani a játékot, ahol lehetőség van gép ellen, illetve egymás ellen játszani. A labor megoldásához a gép elleni játékot nem kötelező megvalósítani, illetve a gép elleni logika lehet egy egyszerű véletlen generátor is.



3.1. ábra - Tic Tac Toe Android példa felület

Az alkalmazás megtervezése és megvalósítása teljes mértékben kötetlen, az elvárt minimális funkcionalitás:

- Saját View osztályból leszármazott nézet létrehozása
- Lépések megfelelő kezelése érintő képernyővel
- Győzelem/döntetlen detektálása
- Menüből való játék indítás és a végén menühöz való visszatérés
- A játék állapotának mentése: ha a felhasználó elnavigál az alkalmazásból, illetve ha elforgatja az eszközt, ne vesszen el a játék aktuális állása

Az állapotmentés megvalósítható az *.onSaveInstanceState(...)* metódus felüldefiniálásával, illetve a Fragment előadáson hallott módszerrel is (*.setRetainInstance(true)* hívás a Fragment példányon).

A fenti specifikációnak megfelelő játék már jeles osztályzatot jelent.

Az AUT portálra feltöltendő az alkalmazás projekt könyvtárán felül egy 1-2 oldalas felhasználói kézikönyv, ami leírja az elkészített szoftver funkcióit, és képeket is tartalmaz minden releváns képernyőről. **A feltöltés határideje: vasárnap 23:59.**

Jó munkát!