

# Android - Feladat

Csináljunk TicTacToe alkalmazást!!!4!négy

A folytatás

# III. feladatrész

- ▶ Eredmény tárolása, nyertes kiírása
- ▶ Osztályváltozó: eredmény 3x3 int tömb, aktuális bökött X és Y (a kép pozíciója), ez is nyilván int lesz
- ▶ Érdemes készíteni egy függvényt, amely az aktuális játékos számát adja vissza a boolean alapján, amit utána majd a tömbben tárolunk
- ▶ onCreate kiegészítése
  - ▶ Ahol az onClickListenererek be vannak állítva, ott beteggelhetjük a képeinket a sorszámaikkal, hogy utána könnyen tudjuk csak az onClickben kapott View paraméteréből, hogy az pontosan hanyadik kép is volt, amire böktünk.
  - ▶ Erre van egy egyszerű módszer minden View-nak van egy tag változója, amibe bármit bele tehetünk a setter-ével és kivehetjük a getter-ével:
  - ▶ `view.setTag(i), int i = (Integer) view.getTag()`

# III. Feladatrész - folytatás

- ▶ **onClick:**
  - ▶ Itt meg kell határoznunk az aktuális X-et és Y-t, hogy az eredmény tömbbe beleír hassuk a játékosunk számát, aki épp soron van
  - ▶ Az egyikhez csak ki kell olvasni a már jó előre elteggelt integerünket
  - ▶ A másikhoz meg kell néznünk, hogy a paraméterül kapott megbökött View, melyik sorba tartozik
  - ▶ `view.getParent() == row`
  - ▶ Ezután csak beírjuk az eredmény tömbbe, az aktuális játékos számát a kiszámolt pozícióba
  - ▶ Majd meghívunk egy nyertes ellenőrző függvényt (amit létre is kell hoznunk)

# III. Feladatrész - A nyertes ellenőrzése

- ▶ Hogyan tudjuk leellenőrizni, hogy vége-e a játéknak?
- ▶ (Nyilván elég ennyit tudni, hiszen, ha vége, akkor az nyert, aki épp most tett, teljes indukcióval bizonyítható, házi feladat 😊)
- ▶ 4 dolog van, amit ellenőriznünk kell
  - ▶ Az aktuális sor
  - ▶ Az aktuális oszlop
  - ▶ Az egyik átló
  - ▶ A másik átló

### III. Feladatrész - 4 dolog

- ▶ Az aktuális sorhoz lehet csinálni egy for ciklust ami fut 0-2-ig és ha a sor minden elemében az a játékos szerepel, aki most épp soron van, akkor ő nyert.
- ▶ Az aktuális oszlophoz lehet csinálni egy for ciklust ami fut 0-2-ig és ha az oszlop minden elemében az a játékos szerepel, aki most épp soron van, akkor ő nyert.
- ▶ Az egyik átlóhoz lehet csinálni egy for ciklust ami fut 0-2-ig és ha az átló minden elemében az a játékos szerepel, aki most épp soron van, akkor ő nyert.
- ▶ A másik átlóhoz lehet csinálni egy for ciklust ami fut 0-2-ig és ha az átló minden elemében az a játékos szerepel, aki most épp soron van, akkor ő nyert.

# III. Feladatrész - 1 dolog

- ▶ Hasonlóak, nem?
- ▶ Most akkor csináljak 4 for ciklust?
- ▶ Csinálhatsz, de az egész megoldható egyben is.
- ▶ 😊

# III. Feladatrész - A nyertes kiírása

- ▶ Valamilyen formában írjuk ki a nyertest
- ▶ Erre tökéletes, ha megjelenítünk egy TextView-t, amibe beleírjuk, hogy nyert a Player 1. vagy 2.

## IV. Feladatrész - StartActivity

- ▶ Új Activity létrehozása
- ▶ Oldalsó fanézetben keressük meg a java mappában a package-ünket
- ▶ Jobb klikk, New / Activity / Blanc Activity
  - ▶ Nevezzük el, ahogy tetszik érdemes valami start-al kapcsolatos nevet adni neki
  - ▶ Ez ugyanaz a felület lesz, ami már a projekt készítésekor is fogadott minket
  - ▶ Szerencsénkre ez legenerál nekünk egy layoutot is hozzá
  - ▶ Illetve beleírja a Manifest fájlba az Activity-t.
  - ▶ Nyissunk meg a Manifest fájlt (oldalsó fa nézetben AndroidManifest.xml)
  - ▶ Másoljuk át az eredeti Activity-nknél található <intent-filter> tag részt az új, most létrehozott activity-hez (ezzel azt érjük el, hogy ha elindul az alkalmazás, akkor ez az Activity fog elindulni, nem a játék)



## IV. Feladatrész - Nézet elemek

- ▶ Legyen a StartActivity nézetünkön két EditText, amely a két játékosunk nevét fogja tartalmazni
- ▶ Valamint egy Játék kezdése gomb

## IV. Feladatrész - Az indulás

- ▶ Az onCreate-ben mentjük el az EditTexteket, valamint tegyünk a gombra onclicklistenert
- ▶ Ha rányomnak a gombra kivesszük a játékosok neveit az EditTextből
  - ▶ `String playerName = editText.getText().toString();`
- ▶ Majd szándékunkban áll új Activity-t indítani, extraként megadva a játékosaink neveit 😊
  - ▶ `Intent i = new Intent(this, GameActivity.class);`
  - ▶ `i.putExtra("player", playerName);`
  - ▶ `startActivity(i);`

## IV. Feladatrész - a Játék

- ▶ A játékhoz tartozó Activity-t egészítsük ki
- ▶ Az onCreate-jében mentjük el a korábban megadott játékosok neveit
  - ▶ `String playerName = getIntent().getStringExtra("player");`
  - ▶ (ennek a függvénynek a put-nál megadott kulcs a paramétere)
- ▶ Legyen a játékmező felett egy TextView, amiben kiírjuk melyik játékos van soron
  - ▶ Tegyük bele a Layout-ba a TextView-t
  - ▶ Az onCreate-ben mentjük el ezt is, valamint állítsuk be neki az első játékos nevét
  - ▶ Az onClick metódus legvégén, ahol a currentPlayer boolean-t is állítjuk, állítsuk át ezt is a másik játékos nevére

# V. Feladatrész - apró kiegészítések

- ▶ Ellenőrizzük le, hogy mindkét játékos neve ki van-e töltve!
  - ▶ Ha igen, indíthatjuk a játékot
  - ▶ Ha nem, akkor írjuk ki, hogy valami nincs kitöltve és ne indítsuk a játékot
- ▶ Ellenőrizzük le, hogy ahova éppen kattintottunk oda nem rakott már korábban másik játékos
  - ▶ Ha nem tett oda senki, akkor a szokott mederben haladunk tova
  - ▶ Ha viszont már foglalt, akkor ne mentjük az eredménybe és ne is cseréljük a képet, valamint az aktuális játékost se állítsuk át, viszont mondjunk tósztot.
  - ▶ `Toast.makeText(this, "Ide nem tehetsz!", Toast.LENGTH_SHORT).show();`

# PRO liga

- ▶ aki már eljutott eddig vagy otthon szeretne továbbmenni
- ▶ SharedPreferences-be menteni a játékosokhoz tartozó győzelmeket
  - ▶ <http://developer.android.com/training/basics/data-storage/shared-preferences.html>
- ▶ Még egy Activity statisztikáknak, ahol ki van listázva melyik játékos hányszor nyert
- ▶ Kiegészíteni a Start activity-t egy csúszkával, ahol beállítható, hogy hányszor hányas legyen a pálya
- ▶ És hogy mennyi kell a győzelemhez
- ▶ És a végére lesz egy egész jó amőbajáték 😊

Ne felejtsetek el nekem elküldeni  
levélben (mármint az én csoportom :D)

<http://erdekes.herokuapp.com>  
[gyulavari.adam+petrik@gmail.com](mailto:gyulavari.adam+petrik@gmail.com)