**Felhasználói dokumentáció**

**Játék rövid leírása**

A mi feladatunk a darts krikett nevű játék volt. A játék szabályát és menetét ismertetem.

Ezt a játékot úgy játsszuk 15-től 20-ig és a 25-öst(bullt) 3-szor ki kell dobnunk úgy, hogy nekünk legyen a legkevesebb büntetőpontunk. Három nyíllal dobunk és egy nyíllal lehet sima, dupla(kétszeres), tripla(háromszoros) értéket dobni. Büntetőpontot úgy lehet adni, hogy egy számot én kidobok 3-nál többször addig, amíg az ellenfelem nem dobta ki 3-szor azt a számot. (Például én kidobom a 20-ast 5-ször és a 15-öst 2-ször, akkor a 3-mat kidobtam és kész vagyok a 20-as mezővel és az ellenfelemnek adok 2\*20 büntetőpontot tehát 40-et, a 15 pedig csak 2-szer van kidobva, tehát még nincs kész. Ezután az ellenfelem dob 9\*15-öt, akkor neki kész a 15-ös mező és ad nekem 6\*15 büntetőpontot, ami 90 pont. Ha kidobná az ellenfelem a 20-ast 3-nál többször, akkor nem tudna már büntetőpontot adni nekem, mert kész az a mezőm. Az nyer akinek a legkevesebb büntetőpontja van és kész kidobta az összes számot 3-szor.)

**Mi programunk**

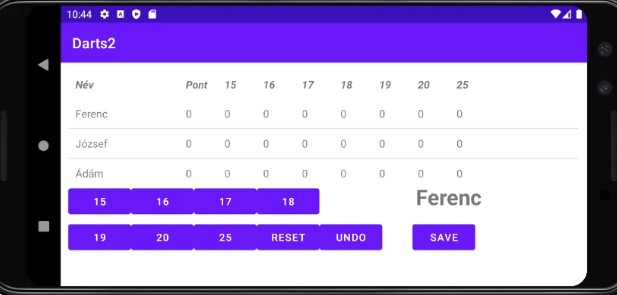
Nézzük akkor, hogy a mi programunk, hogy működik.   
Első lépésben megtudjuk adni a játékosaink nevét.



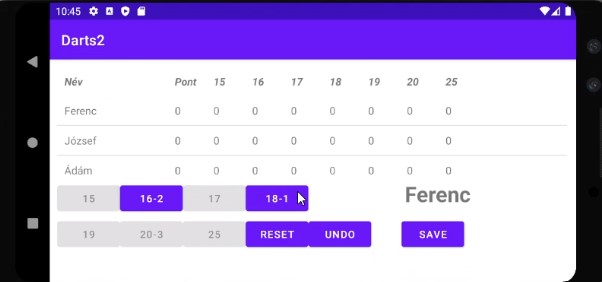
A neveket beírtuk utána mentjük a neveket és elkezdjük a játékot a „mentés és játék gombbal.



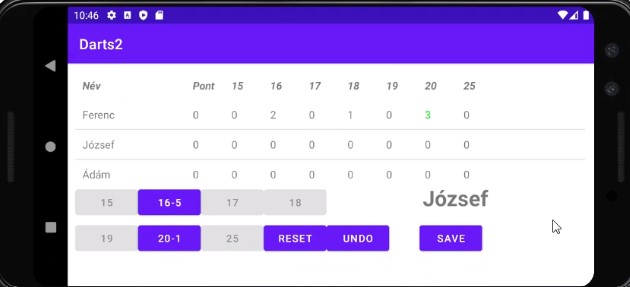
Megjelenik a játékunk, ahol a dobásokat rögzíteni tudjuk. A reset gombot, akkor használjuk, ha még nem mentettük az eredményt, de már egy mezőre kattintottunk véletlenül, amit nem is dobtunk. Az undot pedig akkor használjuk, ha már mentettük az eredményt, de utólag rájöttünk, hogy rossz eredményt mentettünk.



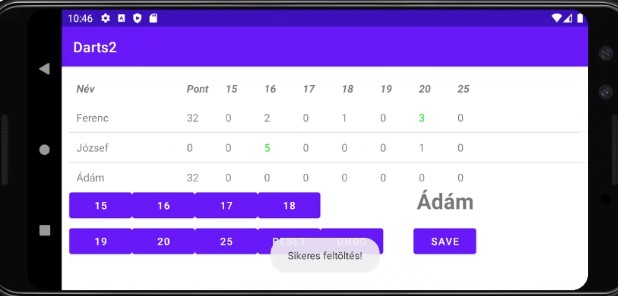
A dobásainkat úgy tudjuk rögzíteni, hogy a dobott értékre annyiszor nyomunk ahányszor kidobtuk azt az értéket és a save gombbal mentjük. Ahogy látjuk egy nyíllal maximum 3-szoros értéket lehet dobni és 3 nyílunk van. 3 különböző értéket dobtunk, az első a 3db 20-as ami le van zárva azért, mert mellette másik két érték is dobva van a képen látott mennyiségben. A többi mező, amit nem dobtunk, azért van lezárva, mert 3 nyíllal már dobtunk 3 értéket. A 16 és 18 azért nincs lezárva, mert abból dobhattunk akár triplát is és csak akkor zárna le ha elérjük a 3 darabszámot. A save gombbal mentjük az értékeket.



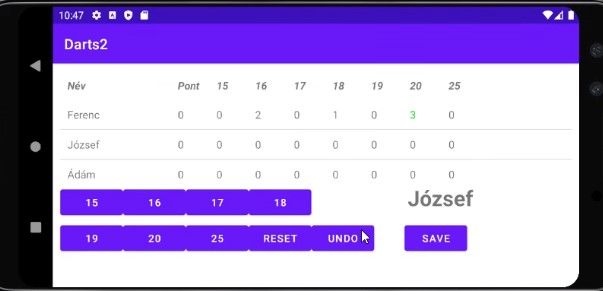
Az első játékosnál látjuk a bevitt értékeket és most a második következik. 5db 16-ost és 1db 20-ast dob. Ez úgy lehetséges hogy 1tripla 16, 1dupla 16 és 1sima 20-as. elfogyott a 3db nyílunk, ezért a többi mező lezár és csak akkor zárna le a 16- és 20-as mező, ha mindegyik tripla lenne.



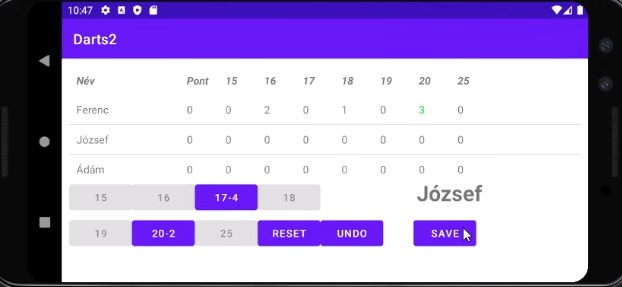
Látjuk, hogy sikeres volt a feltöltés és a másik két játékos 32 büntetőpontot kapott, mert 5db 16-ost dobott József. 3db volt amíg elérte, hogy kidobott mező legyen és 2 maradt, ami pedig büntetőpont lett.



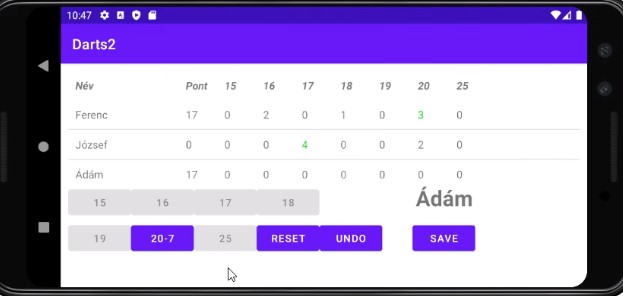
Ezt az undo gombbal vissza tudjuk léptetni, ha véletlen rossz eredményt viszünk be.



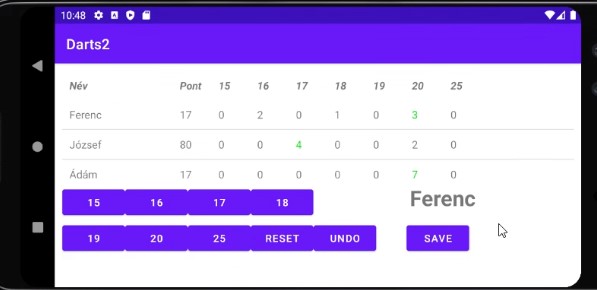
Most már a jó értéket visszük be. Látjuk, hogy a 17-ből 4db-t dobott József, ezért 17 büntetőpontot kap majd a másik két játékos.



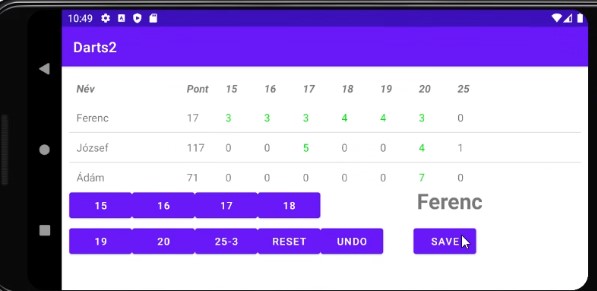
Ádám dob 2db tripla 20-ast és 1db simát. Mivel Józsefnek még nincs 3-szor kidobva a 20-as mező, ezért ő 4\*20 büntetőpontot kap. Mivel Ferencnek már kész a 20-as mező, ezért ő nem kap büntetőpontot.



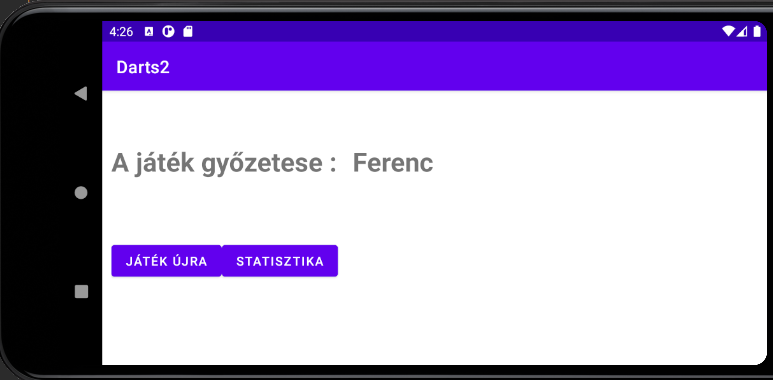
József meg is kapta a 80 büntetőpontot.



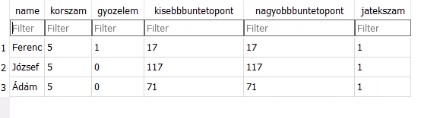
Zajlik tovább a játék és Ferencnek már csak 3db 25ös hiányzik. Ezt meg is dobja és ahogy látszik neki van a legkevesebb büntetőpontja.

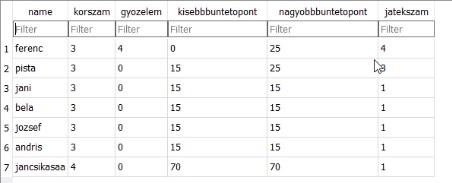


A nyertes Ferenc lett. Ezután megtudjuk nézni a statisztikát vagy új játékot játszani.



Itt a mérkőzésünk statisztikáját látjuk. Látható a statisztikában, hogy hány kör volt a játéka, mennyi győzelme van a játékosnak, legkevesebb és legtöbb büntetőponttal lejátszott mérkőzés és az eddigi játékainak száma.



Itt pedig az összes eddigi játék statisztikája. 

Ugyanez a játékon belül:

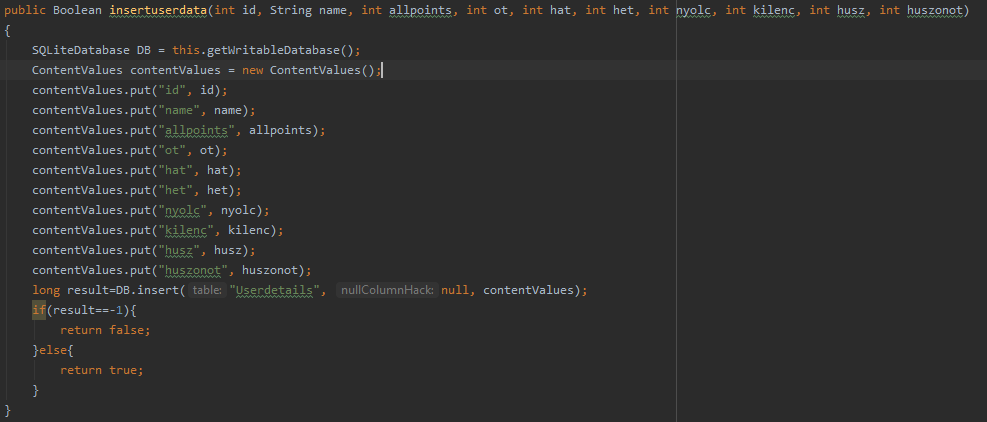


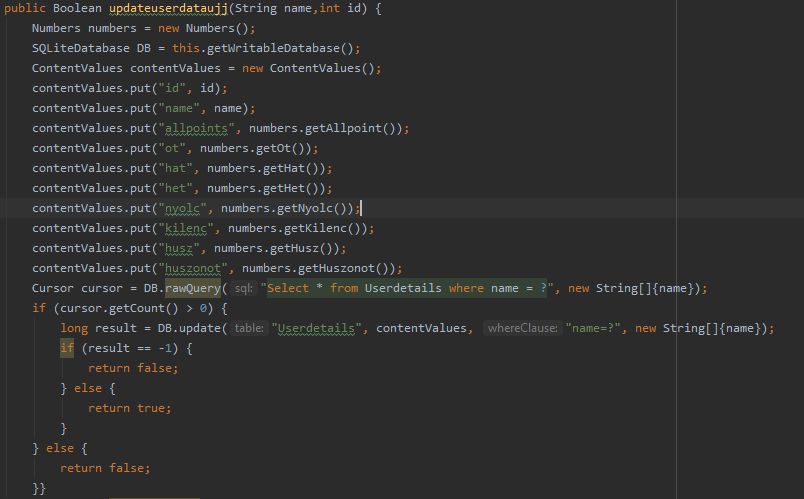
Fejlesztői dokumentáció

**DBhelper**

A dbhelper osztály a táblázatért felelős, míg a dbhelpersecond a staatisztikáért

insteruserdata: játékosok rögzítése a táblába. Rögzíti a neveket és a dobható pontokat

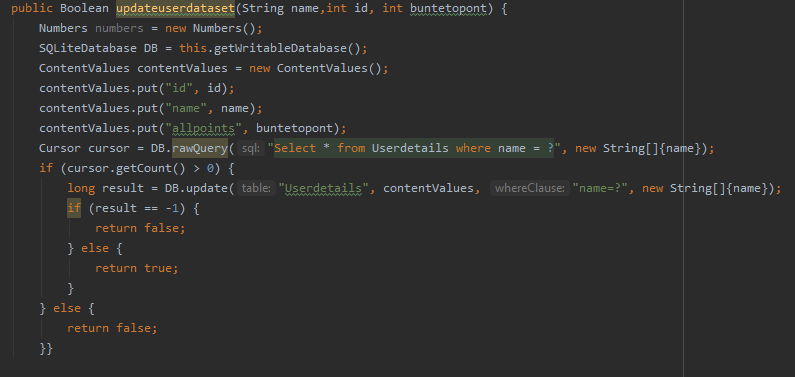


updateuserdataujj: frissíti a táblát a dobott pontokkal és lekéri a számok összegét, ami tartalmazza a dobott számokat és a számok összegét

updateoldlist: az eggyel korábbi állapotot tartalmazó táblát frissíti. Lekéri a játékosok egyes értékeit és azt tölti bele az adatbázisba



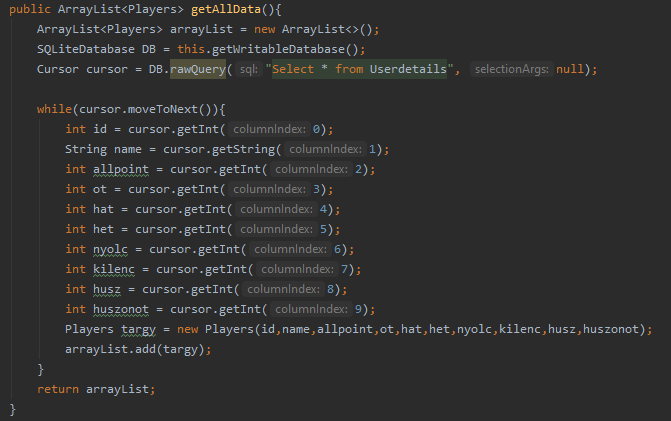
updateuserdateset: a játékoshoz tartozó büntetőpontokat frissíti. Lekéri a játékosokhoz tartozó pontokat.



getdata: beolvassa a három játékos adatait a táblából az adatbázisba.

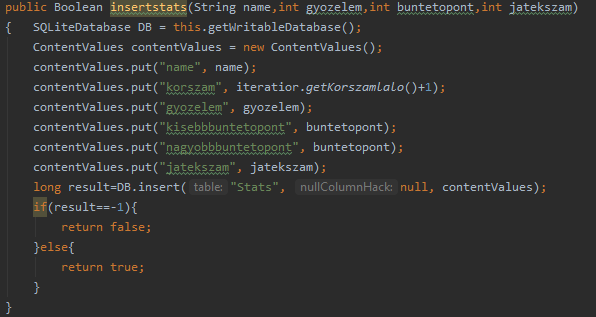


getAllData: az előbb betöltött adatokat, majd mezőnként szétdarabolja és egy players tipusú osztálynak visszaadja az értékét, majd beletölti egy listába

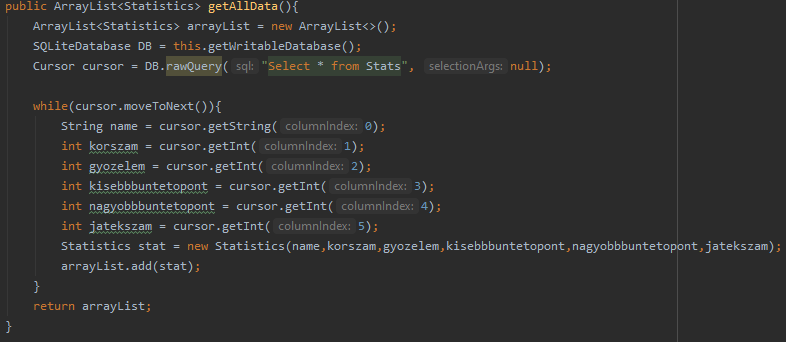


**DBhelperSecond**

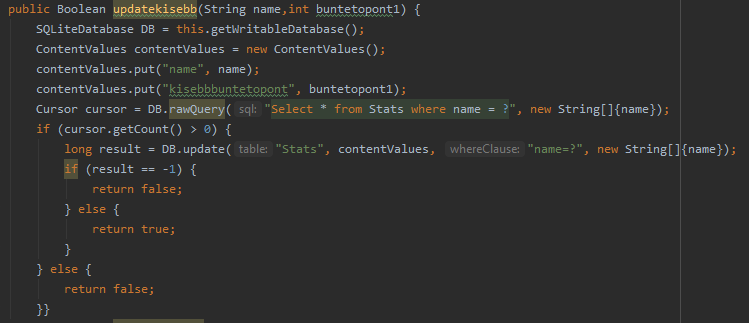
insertstats**:** bementi a játékos adatait a statisztikába, a contentvalue első értéke a mezőnév, értéke az amit megkap paraméterként az insertstats metódus

****

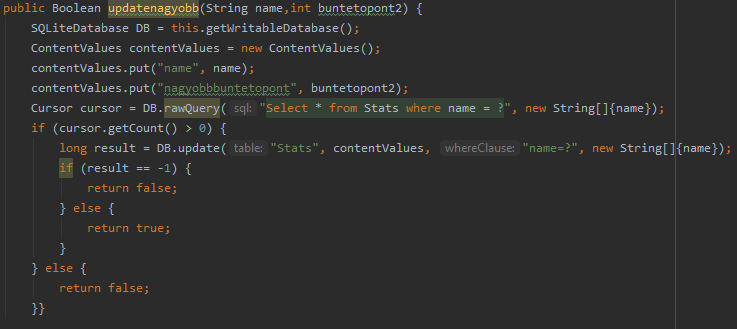
getAllData: elmenti egy listába a játékoshoz tartozó változókat, feldarabolja a sorokat részekre addig olvas ameddig van sor és ez visszaad egy statisztika listát

****

updatekisebb: frissíti a kisebb büntetőpont értékét az adatbázisban

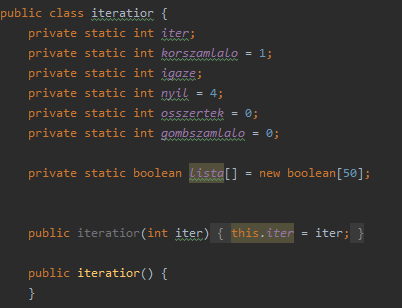


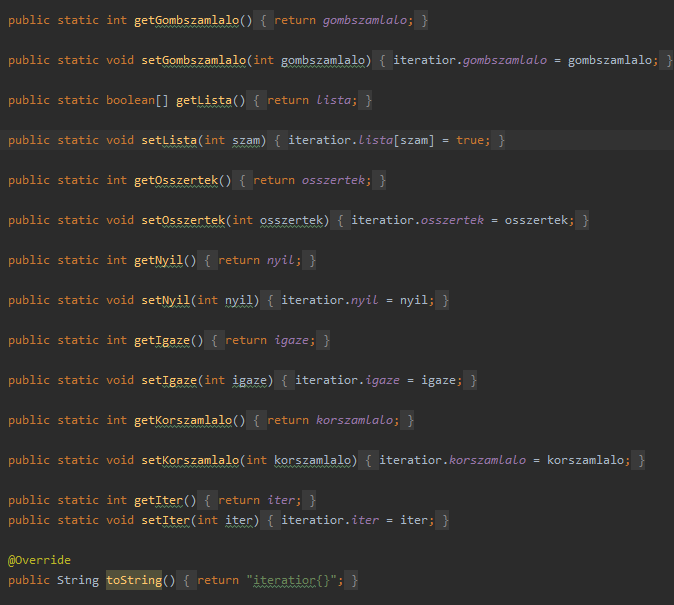
updatenagyobb: a nagyobb büntetőpontok felülírja az adatbázisban



**iteratior**

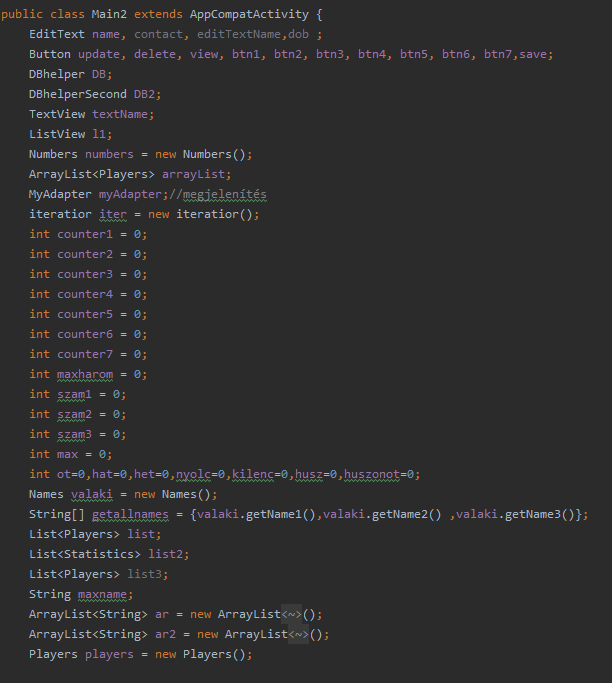
Olyan változókat tárolunk itt, amelyeket a program során használunk. Ilyen változók például a léptető, a körszámláló, a nyilak száma és a pontok összege.





**Main2**

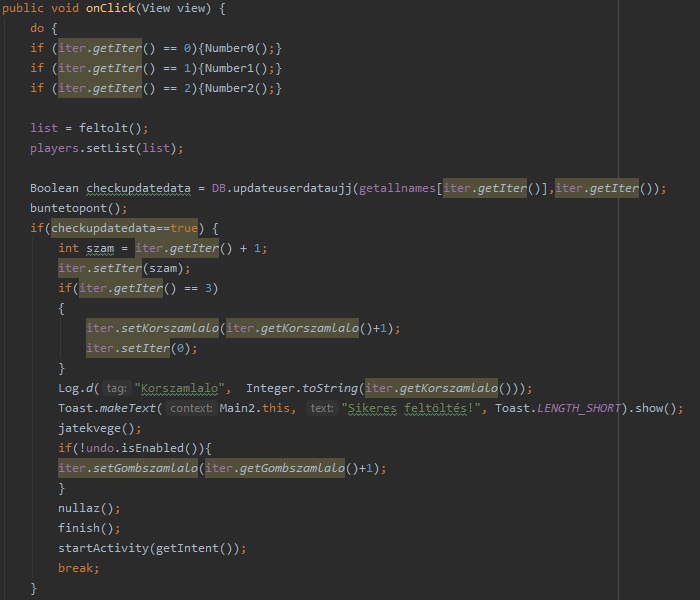
Ez a program fő kódsora. Az elején változók és a listák létrehozása található.



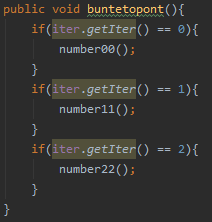
onCreate: a játék indítása fut le. Beállítja a változókat, megjeleníti a táblázatot az adatokkal és megjeleníti a gombokat.



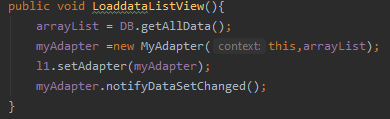
onClick: A save gomb lenyomásakor a pontokat elmenti az adatbázisba és a büntetőpontokat kiosztja. A harmadik játékos után frissíti a körszámlátót. A save gomb lenyomásával frissíti az adatbázist és megjeleníti azt.



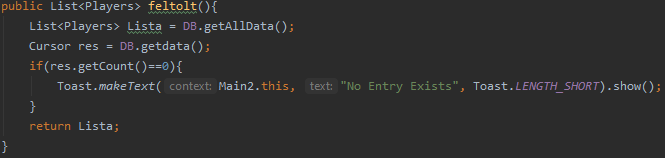
buntetopont: azért felelős hogy a büntetőpontot a megfelelő játékosnak osztja ki



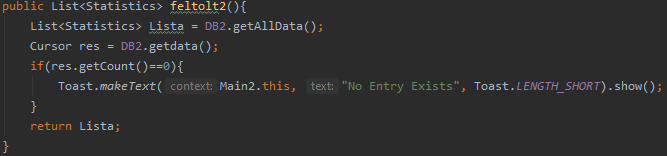
LoaddataListView: egy adaptert hoz létre aminek a tartalma az adatbázis.



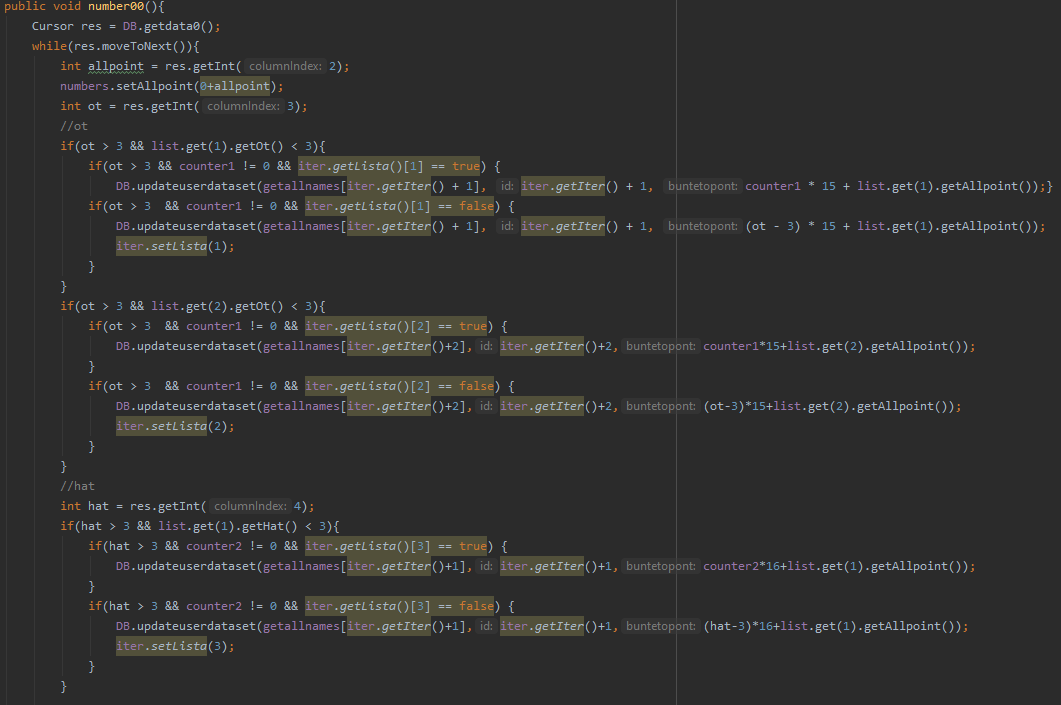
feltolt: a játékosokat adatait feltölti egy listába



feltolt2: a statisztikát beletölti a listába

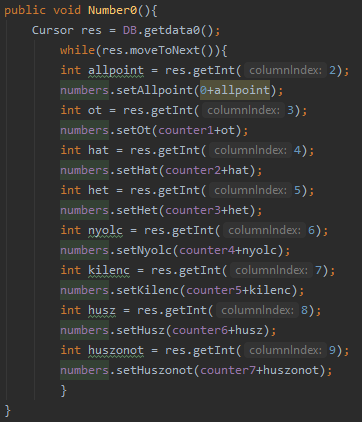


number00: Ez az első játékossal foglalkozik, a while ciklus során figyel hogy megfelelő mennyiségű büntetőpontot kapjon a játékos, majd minden ponthoz meg van adva a büntetőpont számítás módszere, majd fel is tölti az adatokat az adatbázisba



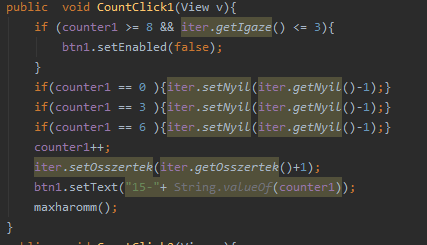
A number11 és a number22 az előző függvény nagyon hasonló, csak a második és a harmadik játékoshoz van igazítva.

Number0: a while ciklusban az első játékos adatait olvassa ki az adatbázisból, és beállítja a változóit

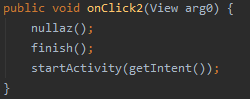


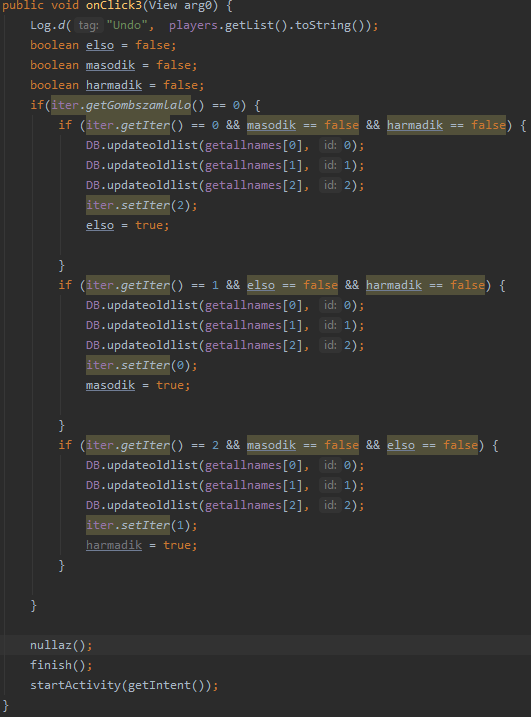
Number1 és a Number2 a második és a harmadik játékos adatit olvassa ki.

CountClick1-7: Ez a dobható számok gombjait tartalmazza. Számolja, hogy hány dobás maradt még tiltja is a gombokat a játékszabályzat szerint.

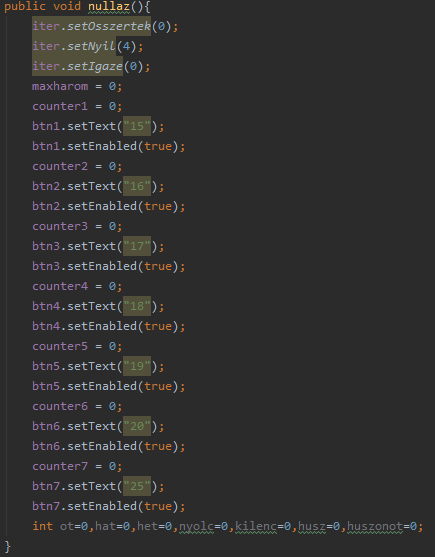


onClick2: Ez a reset gomb. Ha a játékos rossz számokat ír be akkor a reset gombbal le tudja tudja nullázni



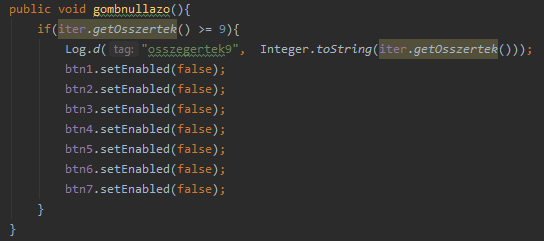
onClick3: az undo gomb működése. Ha a kör után ha meg akarjuk változtatni az adatokat akkor az undo gombbal visszamegy az előző lépésre és újra be lehet vinni az adatokat.

nullaz: Az onclick3 ezt hívja meg, ez állít vissza mindent az eredeti állásra

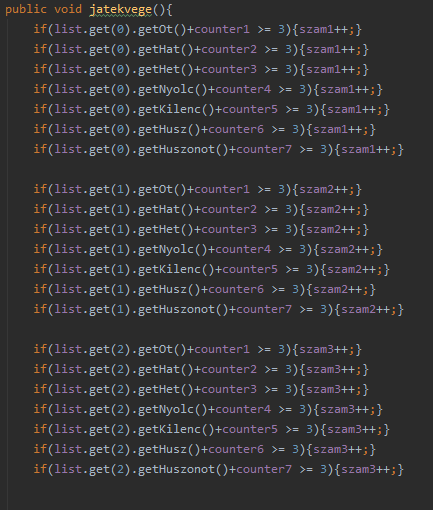


maxharom: tiltogatás szabályzat. Ha idáig eljutottál inkább gyere Valorantozni. Izgalmasabb mint ez a szar

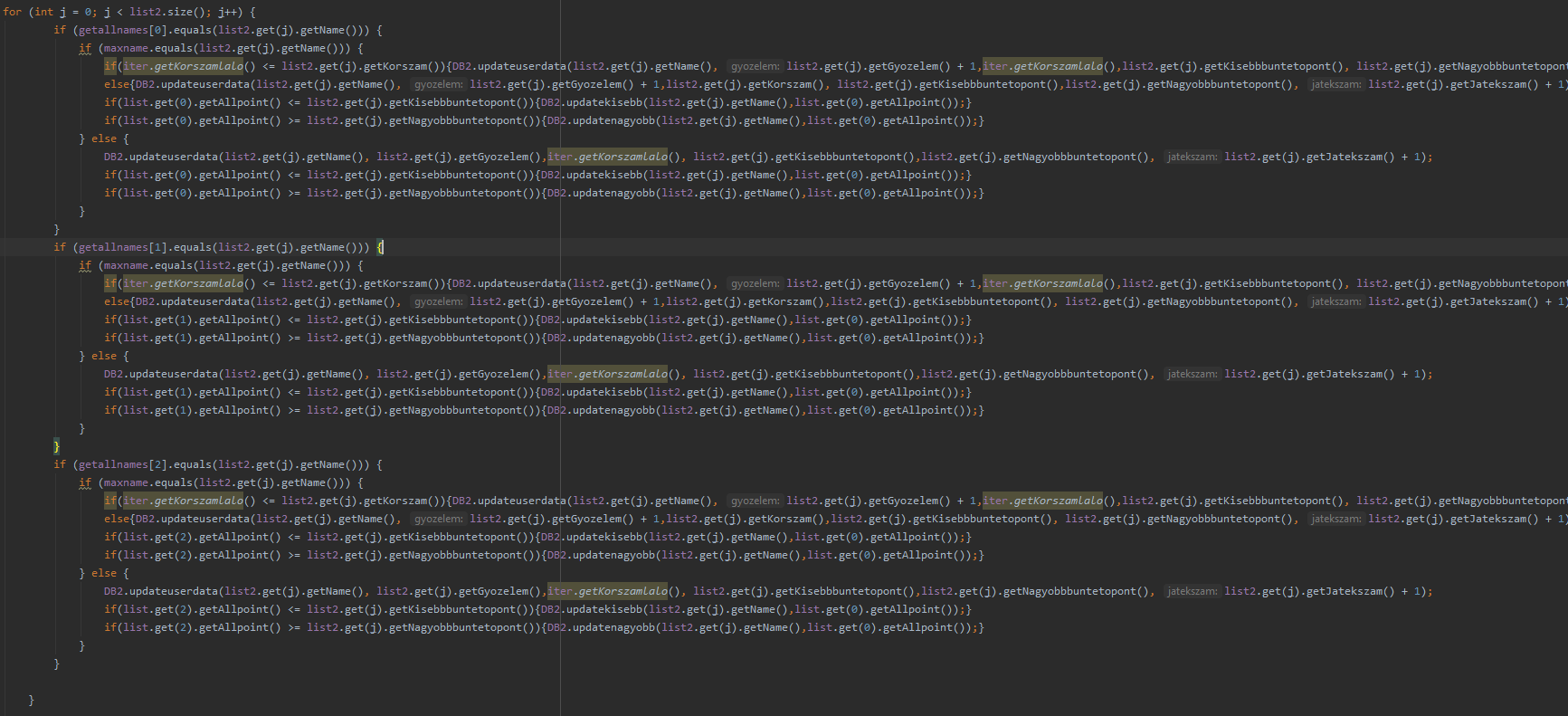
gombnullazo: azt figyeli hogy a játékost maximum 9-szer nyomhatja meg ugyanazt a gombot.



jatekvege: visszaadja a játékost a dobott pontjával együtt



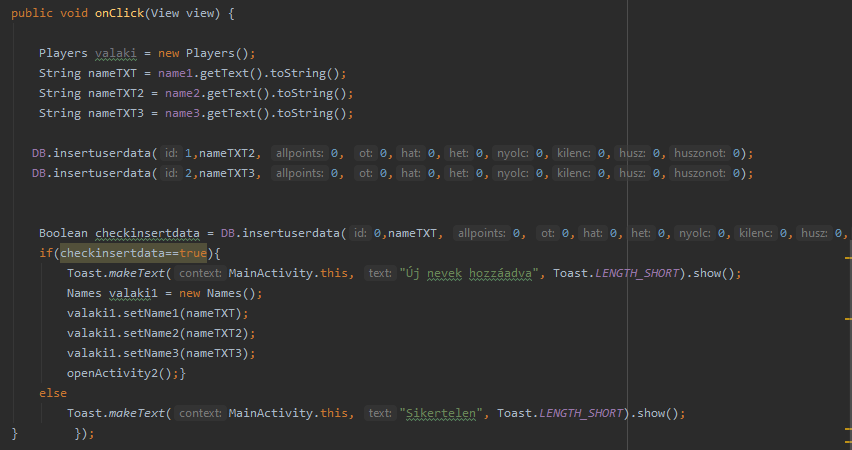
Ez a for ciklus először megnézi hogy a játékos létezik-e a statisztikában, ha nem akkor hozzáadja, majd összehasonlítja a két listát(az adatbázist és a statisztikát) és ha van olyan játékos aki még nincs a statisztikában akkor hozzáadja. A győztesnek a gyozelem tagjét átírja 1-re



**MainActivity**

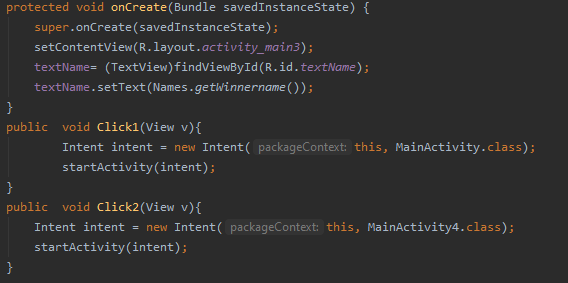
Ez a játékos bevivő oldal kódja.

onClick: a három játékost bekéri és a hozzá tartozó adatokat lenullázza, majd ellenőrzi hogy jól töltötte-e fel, ekkor kiírja hogy „Új nevek hozzáadva”, sikertelen esetben a „Sikertelen” üzenetet. Sikeres esetben megnyitja az Activity2-t



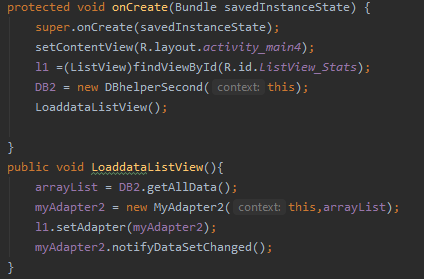
**MainActivity3**

Ez a 3 oldal kódja. Kiírja, hogy melyik játékos nyert, innen lehet új játékot indítani vagy a statisztikát megnézni.

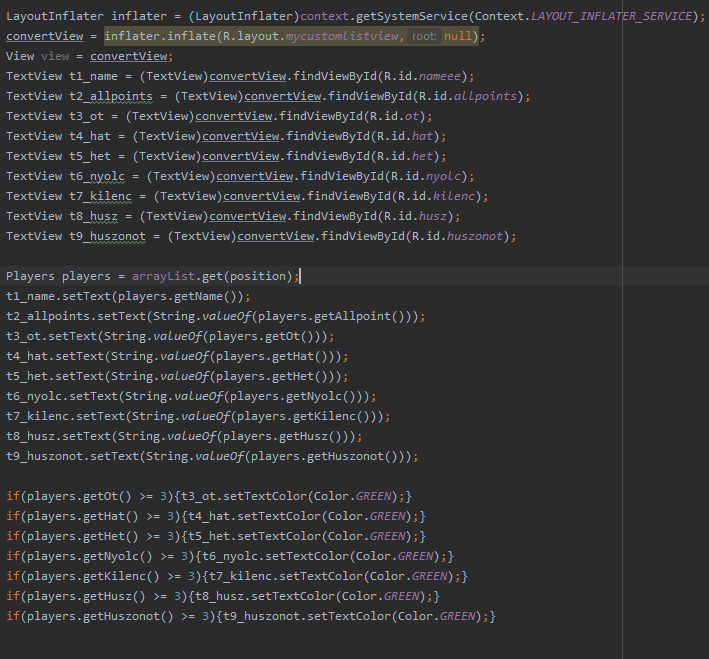


**MainActivity4**

A játék végén megjeleníti a statisztikákat egy táblázatban

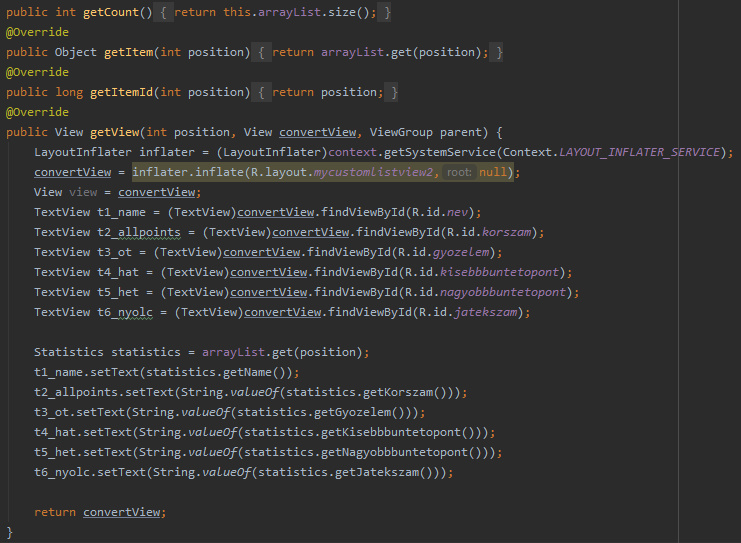


**MyAdapter**

Megjeleníti a 3 játékos adatait: a nevét, az összpontszámát és a dobott pontjait

**MyAdapter2**

A statisztikában szereplő adatokat jeleníti meg a statisztikánál.



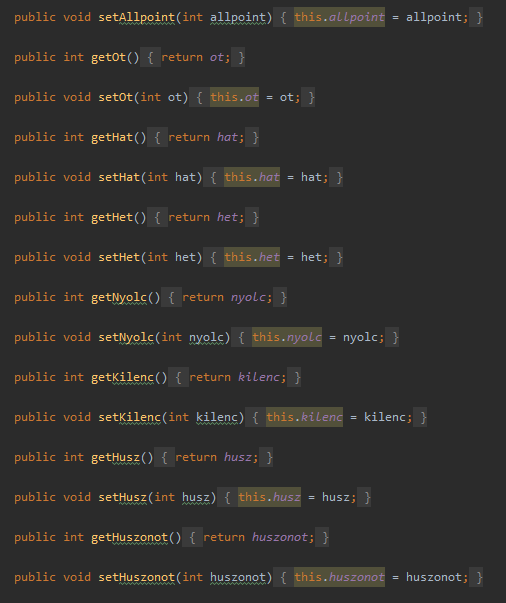
**Names**

Itt tároljuk el a játékosok nevét és a győztes nevét.



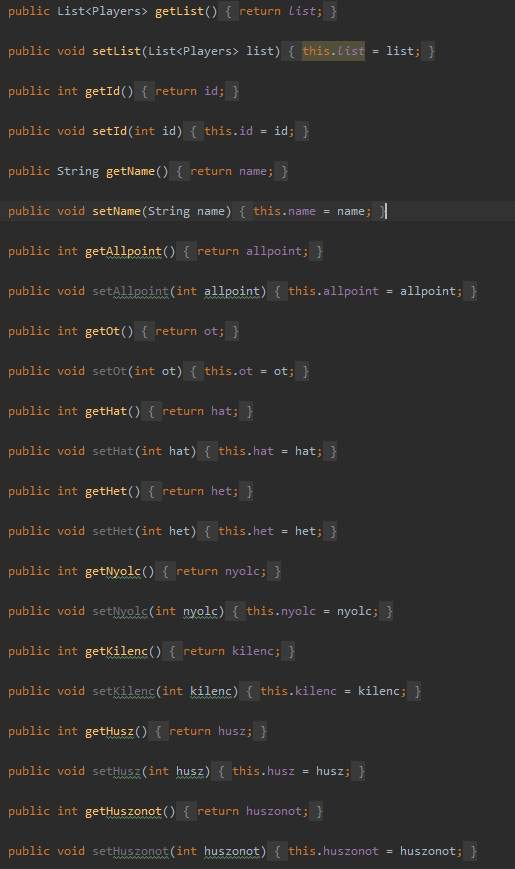
**Numbers**

A játék során dobott számok összegét és az egyes dobott pontokat tároljuk itt, hogy a játék során könnyen hozzáférhessünk és tudjuk frissíteni.



**Players**

Egy összesített lista az adatokról, játékosokról. Beállítjuk és lekérjünk a változókat a get és set metódusokkal.



**Statistics**

Lekérdezi és beállítja a neveket, győzelmek számát, a büntetőpontokat, játékok számát.



A megjelenítésért az activity\_main1-4-ig felelős. Itt állítjuk be a szöveg elrendezését, méretét, a megjelenítendő szöveget és gombokat.

A mycustomlistview1 és 2 a screen2 és a statisztika táblájának egy sorának a kinézetéért felelős.