# Memóriajáték

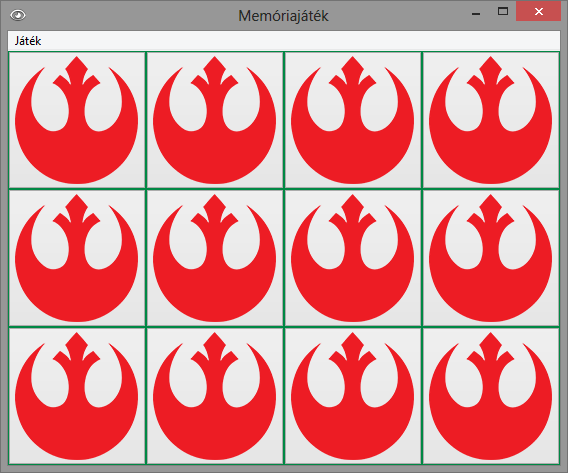
# Felépítés

A memóriajáték 2 ablakból áll:

* Bejelentkező ablak
* Játéktér

A program indulásakor a játéktér ablakát elrejtjük, csak a bejelentkező ablakot mutatjuk.

Amennyiben sikerül bejelentkezni, megjelenik a játéktér.



A játéktérek egy 4X3-as rácsok csempék helyezkednek el. Ezeket a lapokat a Tile osztály valósítja meg.

A Tile a Button-tól származtatott osztály, így annak minden tulajdonságát és metódusát örökli.

A Tile fontos tulajdonságai:

* COVER\_IMAGE\_PATH konstans, a lefordított lapok borítóképe
* cover\_image a lefordított lapok kép objektuma
* image a felfordított lapok kép objektuma
* image\_path a felfordított lapok képének elérése
* visible True, ha felfordított, False, ha a borító látható
* paired True, ha már megtalálták a párját

Az image\_path fog segíteni minket abban, hogy két lapról megállapítsuk, párok-e. Amennyiben a két lap egy párt mutat, akkor az image\_path értéke azonos kell legyen.

Amennyiben egy lap paired, akkor a lefordítás metódusnak nem szabad lefordítani a lapokat.

A Tile match metódusa bemenő paraméterként egy másik Tile objektumot vár. A két Tile összevetése után True vagy False értékkel tér vissza.

A feladat a kódban TODO-val jelölt részek teljesítése.

# Feladat 1 (LoginScreen.login\_action)

A felhasználó csak akkor jelentkezhet be, ha a megadott felhasználónév csak kis- és nagybetűket tartalmaz és számokat. A unicode karakterek nem elfogadottak (öüóúőáűéí stb.).

Csak a compile metódusban lévő mintát írd át!

# Feladat 2 (GameScreen.\_\_init\_\_)

A menüsorban látható, hogy a Ctrl+n billentyűzet kombinációval kérhető egy új játék.

Valósítsd meg a hozzárendelést.

Más sorhoz ne nyúlj!

# Feladat 3 (GameScreen.init\_game):

Ez az új játéktér inicializáló függvénye.

Be kell állítanod, hogy eddig 0 darab pár találtunk meg. Csak a meglévő tulajdonságokat használd!

Inicializáld a 4X3-as rácsot. Az images 12 darab képet tartalmaz, pontosan 6 lappárt. Ezeket használd fel, amikor a lapokat (Tile) létrehozod. A képek már random sorrendben vannak, ezzel nem kell foglalkoznod.

Ügyelj rá, hogy a lapra kattintáskor fusson le a pair metódus!

# Feladat 4 (GameScreen.pair)

A pair függvény csak eseményként futhat, ekkor a target változóba egy Tile objektum kerül.

Ellenőrizd, hogy ez egy felfordított lap-e. Ha igen, a függvény ne csináljon semmit (return).

Amennyiben már volt felfordítva lap, vessük össze az aktuális és az előző lapot.

Ha a match() metódus szerint azonosak, akkor a pairs tulajdonságot növeld egyel. A lapokat ekkor még ne fordítsd le!

Amennyiben még nem volt felfordítva lap, mentsük el a prev\_card-ba az aktuális lapot.

Mivel nincs mivel összevetni, csak ennyit kell tennünk.

Amennyiben a két előzmény lap nem None (prev\_card és next\_card), akkor azokat Most fordítsd le és nullázd ki a referenciáikat.

Ments el a prev\_card-ba az aktuális lapot (Tile).

Amennyiben megtalálod a 6 párt, a játéknak vége. Tájékoztasd a játékost a hírről egy messagebox-szal.

# Feladat 5 (close)

Amennyiben a felhasználó a menüben a Kilépés opciót választja, kérdezd meg, valóban ki akar-e lépni.

# Utószó

Érdemes átnézni a játék teljes felépítését, osztályait. A vizsgában hasonló felépítéssel kaptok majd feladatot, azaz a GUI el lesz készítve nagy részben. Néhány részt nektek kell majd hozzáfűzni és a funkcionalitás hiányos lesz. A vizsgában valószínűleg nem fogunk ilyen játékot írni, hiszen már lesznek adatbázis adataink, melyekkel sokkal érdekesebb feladatokat is végrehajthatunk.

Többen kértetek gyakorló, egyszerűbb feladatokat is, ezeket is mellékelem, **azonban ezek a házi feladatnak nem képezik részét**. Pontokat nem lehet értük kapni, egyszerűen csak gyakorlási célt szolgálnak.