

## Készítette

Balogh Richárd

E-mail: [balogh.ricsi.felvi@gmail.com](mailto:balogh.ricsi.felvi@gmail.com)

Csoportszám: 17

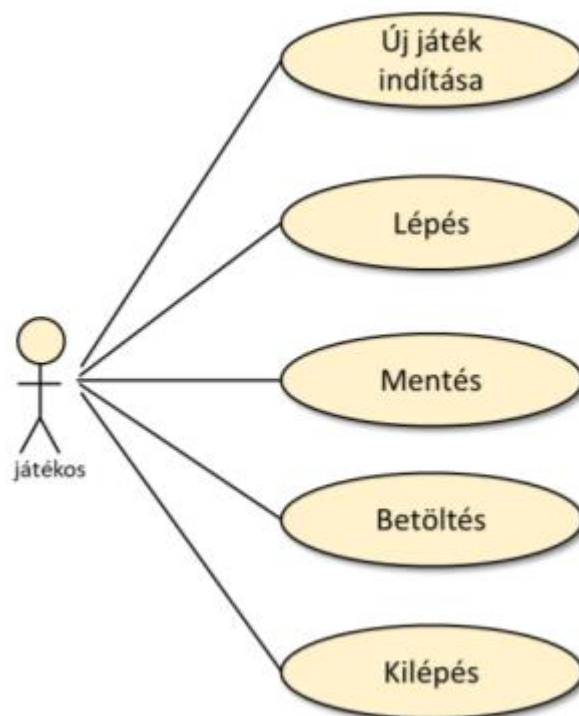
## Feladat

Készítsünk programot, amellyel a következő két személyes játékot lehet játszani.

Adott egy  $n \times m$  mezőből álló tábla ( $n$  az oszlopok,  $m$  a sorok száma), amelyre a játékosok  $X$ , illetve  $O$  jeleket potyogtatnak (azaz egy adott oszlopban a karakter mindig „leesik” a legalsó üres sorba, függetlenül attól, melyik sorban helyeztük le). A játékosok felváltva lépnek, és egy oszlopban csak akkor helyezhetnek el új jelet, ha az még nem telt meg. A játékot az nyeri, aki előbb elhelyez vízszintesen, vagy átlósan négy szomszédos jelet. A játék döntetlennel ér véget, ha betelik a tábla.

A program biztosítson lehetőséget új játék kezdésére a táblaméret megadásával ( $8 \times 5$ ,  $10 \times 6$ ,  $12 \times 7$ ), valamint az aktuális játék mentésére és egy korábban elmentett játék betöltésére. Ismerje fel, ha vége a játéknak, és jelenítse meg, melyik játékos győzött, illetve azt is, ha döntetlen lett a vége, majd automatikusan kezdjen új játékot.

### A Feladat elemzése



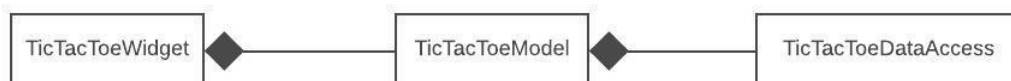
A felhasználói tevékenységek során az alábbi esetek következhetnek be.

1	Alkalmazás indítása	GIVEN:	Az alkalmazás telepítve van.
		WHEN:	Alkalmazás indítása.
		THEN:	Megjelenik az üres játéktábla.
2	Kilépés	GIVEN	A játéktábla aktív
		WHEN	A játékfelület ablakának lezáró ikonjára kattintunk.
		THEN	Az alkalmazás bezáródik.
3a	Lépés	GIVEN	A játéktábla aktív.
		WHEN	Üres mezőre kattintunk, de ezzel nem alakul ki négy egyforma jel egy sorban, oszlopban, vagy átlóban, és marad még üres mező.
		THEN	Attól függően, hogy ki a soron következő játékos egy 'X' vagy 'O' jel kerül a kiválasztott mező oszlopának legutolsó szabad sorába.
3b	Lépés	GIVEN	A játéktábla aktív
		WHEN	Üres mezőre kattintunk, és bár ezzel nem alakul ki négy egyforma jel egy sorban, oszlopban, vagy átlóban, de nem marad üres mező sem.
		THEN	Attól függően, hogy ki a soron következő játékos egy 'X' vagy 'O' jel kerül a kiválasztott mezőre, majd egy külön jelzést kapunk arról, hogy a játék döntetlennel végződött. Ennek elolvasása után új játék kezdődik
3c	Lépés	GIVEN	A játéktábla aktív.
		WHEN	Üres mezőre kattintunk, és ezzel négy egyforma jel jelenik meg egy sorban, oszlopban, vagy átlóban
		THEN	Attól függően, hogy ki a soron következő játékos egy 'X' vagy 'O' jel kerül a kiválasztott mező legutolsó szabad oszlopába, majd egy külön jelzést kapunk arról, hogy a játékos nyert. Ennek elolvasása után új játék kezdődik.
3d	Lépés	GIVEN	A játéktábla aktív.
		WHEN	Foglalt mezőre kattintunk
		THEN	Nem változik a játék állapota.
4	Új Játék	GIVEN	A játéktábla aktív.
		WHEN	Jelezzük új játék indítási szándékunkat, a kiválasztott táblaméretben.
		THEN	Megjelenik az üres játéktábla, a megfelelő paraméterekkel.
5a	Mentés	GIVEN	A játéktábla aktív
		WHEN	Jelezzük a mentési szándékunkat.
		THEN	Megnyílik egy dialógus a mentéshez, az első tárhely a kiválasztott.
5b	Mentés	GIVEN	A mentés dialógus aktív, egy tárhely kijelölve
		WHEN	Kiválasztunk egy tárhelyet.
		THEN	A mentéshez megnyitott dialógus aktív, a kiválasztott tárhely van kijelölve.
5c	Mentés	GIVEN	A mentés dialógus aktív, egy tárhely kijelölve
		WHEN	Kijelölt tárhely üres, és mentést kezdeményezünk
		THEN	A mentés megtörténik, a dialógus bezárul
5d	Mentés	GIVEN	A mentés dialógus aktív, egy tárhely kijelölve.

		WHEN	Kijelölt tárhely nem üres, és mentést kezdeményezünk.
		THEN	A mentést felülírjuk.
5e	Mentés	GIVEN	A mentés dialógus aktív, egy tárhely kijelölve
		WHEN	Elállunk mentési szándékunktól.
		THEN	A dialógus mentés nélkül bezárul.
6a	Betöltés	GIVEN	A betöltés dialógus aktív, egy tárhely kijelölve
		WHEN	Kiválasztunk egy tárhelyet.
		THEN	A betöltéshez megnyitott dialógus aktív, a kiválasztott tárhely van kijelölve.
6b	Betöltés	GIVEN	A betöltés dialógus aktív, egy tárhely kijelölve
		WHEN	Kiválasztunk egy tárhelyet
		THEN	A betöltéshez megnyitott dialógus aktív, a kiválasztott tárhely van kijelölve
6c	Betöltés	GIVEN	A betöltés dialógus aktív, egy tárhely kijelölve.
		WHEN	Kijelölt tárhely nem üres, és betöltést kezdeményezünk.
		THEN	A betöltés megtörténik, a dialógus bezárul.
6d	Betöltés	GIVEN	A betöltés dialógus aktív, egy tárhely kijelölve
		WHEN	Kijelölt tárhely üres, és betöltést kezdeményezünk
		THEN	Hiba üzenetet kapunk.
6e	Betöltés	GIVEN	A betöltés dialógus aktív, egy tárhely kijelölve
		THEN	Elállunk betöltési szándékunktól.
		WHEN	A dialógus betöltés nélkül bezárul.

## Architektúra

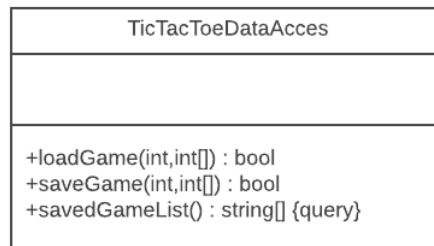
Az alkalmazást négy rétegű (nézet-modell-perzisztencia) architektúra valósítjuk meg. A nézet a játéktáblát megjelenítő ablak mellett két dialógus ablakot is tartalmaz a mentés, illetve a betöltés párbeszédhez.



## Perzisztencia

A korábban elmentett, legfeljebb 5 játékállást külön-külön állományok tároljuk. Az adatelérési réteg egyetlen osztálya ezen állományokhoz történő hozzáférést biztosítja a modell számára úgy, hogy a modellnek nem kell tudni arról, hogy milyen módon történik a tárolás.

## Osztálydiagram



## Metódusok

Az adatelérési réteg négy metódust kínál

A `loadGame()` egy megadott sorszámú állományból (ezt adjuk meg az első paraméterével) tud betölteni egy (44/64/88) egész számmal leírt játékállást (ez a második paraméter).

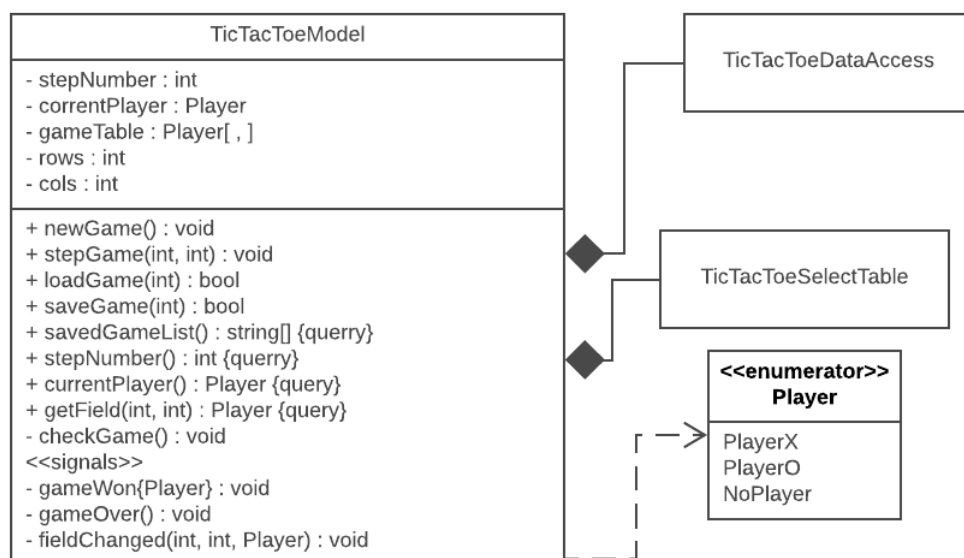
A `saveGame()` egy megadott sorszámú állományba (ezt adjuk meg az első paraméterrel) tud elmenteni egy (44/64/88) egész számmal leírt játékállást (ez a második paraméter).

A `saveGameList()` az elmentett állományok utolsó mentési dátumait adja vissza egy öt elemű sorozatban. A sorozat i-dik eleme üres, ha még nem létezik i-dik állomány.

## Modell

A modell segítségével tároljuk el az aktuális állást. A modell publikus metódusait a nézet hívja, a modell szignálok segítségével üzen a nézetnek.

## Osztálydiagram



## Adattagok

A `stepNumber` az eddigi lépések számát, a `currentPlayer` a soronkövetkező játékost, a `gameTable` pedig a játéktáblát mutatja, a `rows` illetve a `cols`, pedig a sorok és oszlopok számát jelölik.

## Metódusok

A privát adattagok értéke (az aktuális játékállás) a `stepNumber()`, `currentPlayer()`, és `getField()` getterek segítségével kérdezhetőek le.

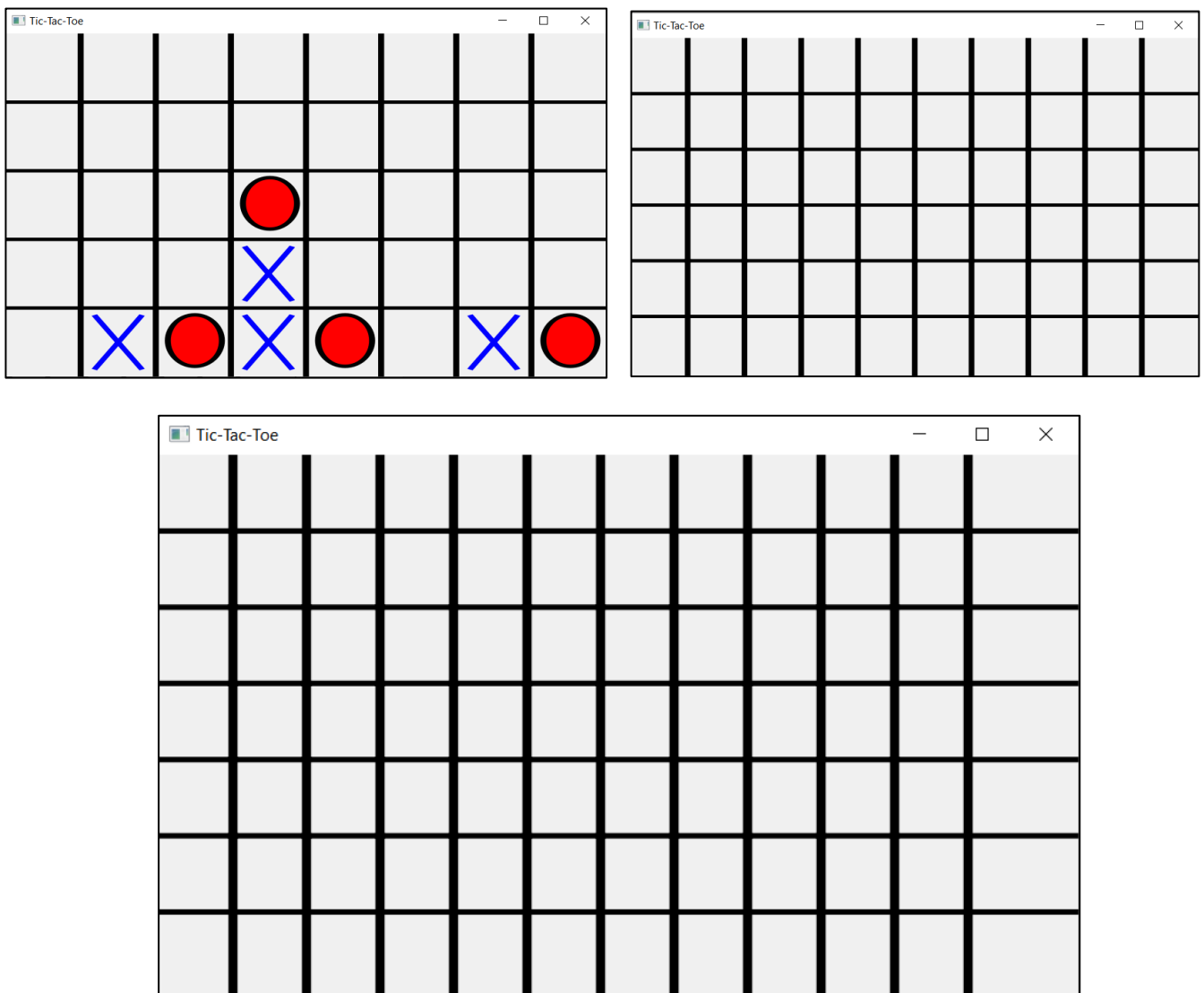
A `newGame()` egy új játék generálását, a `stepGame()` a játék egy lépését végzi. A `stepGame()` hívja a `checkGame()` metódust, amely a játék esetleges befejeződését vizsgálja. Az aktuális állás változásáról szignálok segítségével (`gameWon()`, `gameOver()`, `fieldChanged()`) értesíti a modellt a nézetet.

A modell biztosít kapcsolatot a nézet és az adattárolás között. A `savedGameList()` metódus az adatelérési réteg ugyanilyen nevű metódusát hívja meg, a `loadGame()` és `saveGame()` az elmenteni kívánt, illetve betöltött adatokat alakítja megfelelő formájúra.

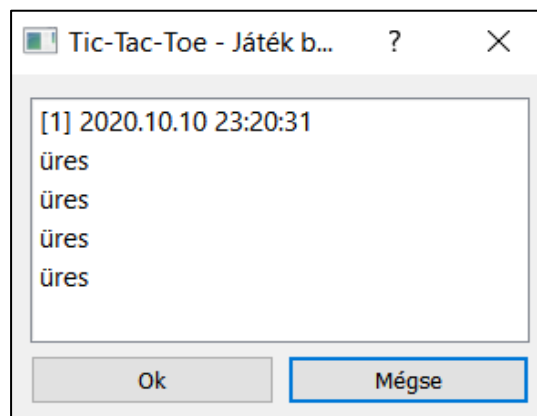
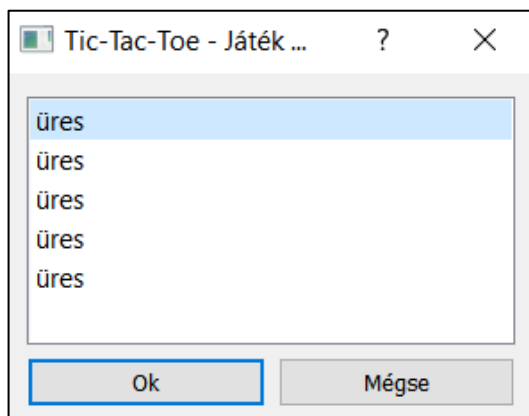
## Nézet

## Felhasználói felület terve

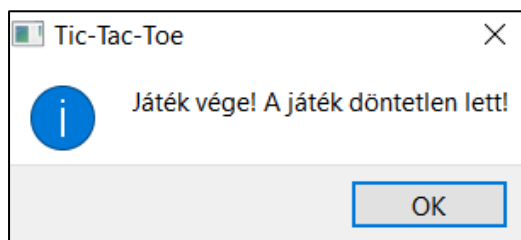
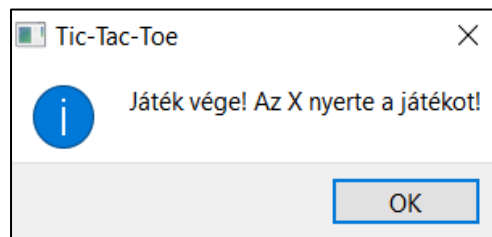
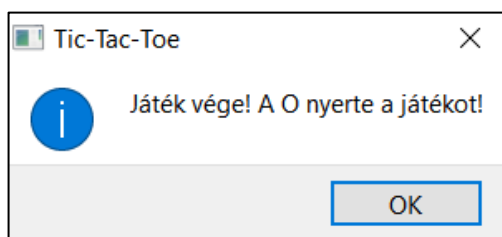
A játék (minimális méret mellett) tetszőlegesen átméretezhető felületén grafikusán kirajzolt X-et és O-t lehet elhelyezni a 4/5/6 függőleges vonal és 7/9/11 vízszintes vonallal kijelölt 8x5/10x6/12x7 játéktábla mezőin.



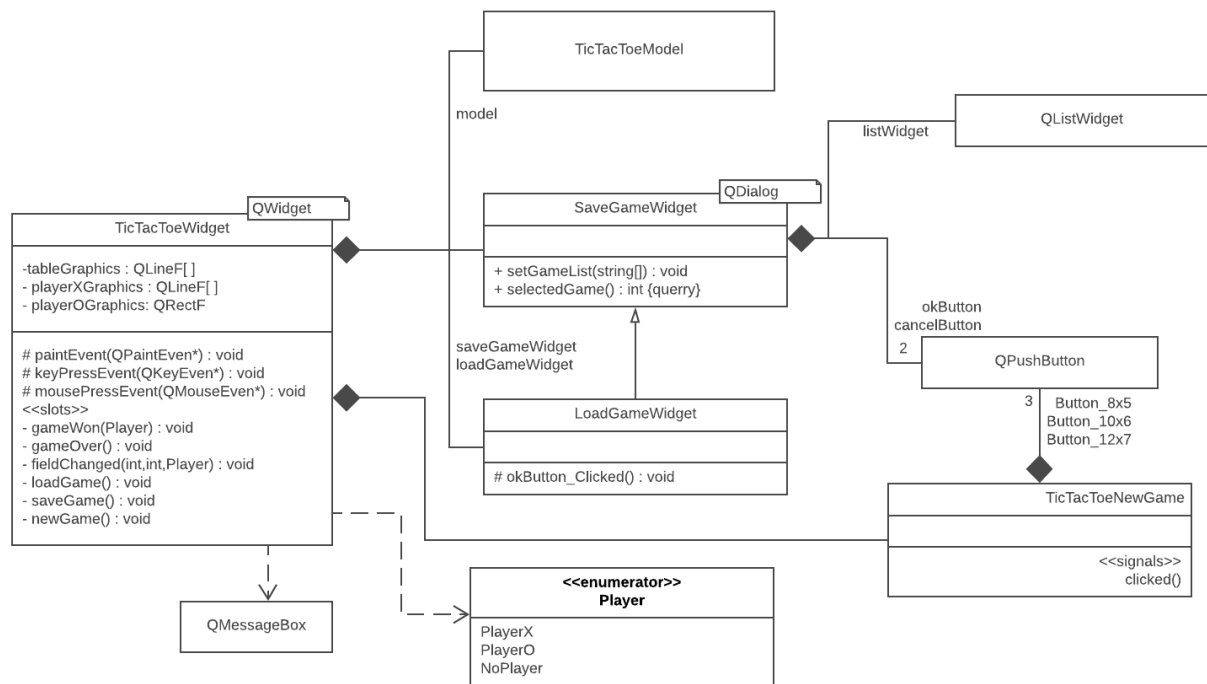
A játék végét üzenet-ablakok jelzik:



A mentés és betöltés dialógus-ablakon keresztül zajlik. A hibaüzeneteket üzenet-ablakok segítségével jelezzük.



## Osztálydiagram



### Vezérlők és adattagok

A főablak osztályának adattagjai a játék megjelenítésére szolgáló grafikai elemeket (tábla, játkosok jelei) tárolják, valamint egy-egy hivatkozás a mentés, betöltés és új játék kezdés dialógus ablakokra és a modellre.

### Metódusok

A főablak osztály felüldefiniálja a `paintEvent()`, `keyPressEvent()` és `mousePressEvent()` metódusokat. Az első felel a játékalás megjelenítéséért, a másik kettő kezeli a felhasználói akciókat.

### Eseménykezelés

Sender	signal	reciever	slot
model	gameOver	TicTacToeWidget	model_GameOver
model	gameWon	TicTacToeWidget	model_GameWon
model	fieldChanged()	TicTacToeWidget	model_FieldChanged
loadGameWidget	accepted()	TicTacToeWidget	loadGame()
saveGameWidget	accepted()	TicTacToeWidget	saveGame()
newGameWidget	clicked()	TicTacToeWidget	newGame()
newGameWidget	clicked()	TicTacToeWidget	newGame()
newGameWidget	clicked()	TicTacToeWidget	newGame()

## Végfelhasználói tesztesetek:

	Teszteset	Elvárt hatás
1	Az alkalmazás indítása	Megjelenik az üres játéktábla.
2	Kilépés folyó játékból	Az alkalmazás leáll
3a	Váltakozó lépések	Az üres mezőkre kattintva váltakozva hol 'X', hol 'O' jelenik meg a mező oszlopának utolsó szabad sorában.
3b	Nem üres mezőre kattintás	Nincs változás a táblán
3c	Létrejött négy azonos jel egy sorban egy oszlopban, egy átlóban	Mindhárom esetben a győztes kihirdetése és új játék kezdése
3d	Győztes nélkül betelt a játéktábla	Megjelenik a döntetlen felirat egy információs ablakban, majd új játék kezdődik.
4a	Új játék indítása egy folyamatban levő játéknál	Megjelenik az üres játéktábla
4b	Új játék indítása egy befejeződött játéknál	Megjelenik az üres játéktábla
5a	Mentés egy folyamatban levő játékról	A program elmenti az aktuális játékállást, ami után folytatni lehet a játékot
5b	Mentés egy befejeződött játékról	A program elmenti az aktuális játékállást, ami után új játékot lehet kezdeni.
5c	Mentés elvetése	Nem történik változás.
6a	Betöltés egy folyamatban levő játéknál	Az elmentett játék betöltődik, azzal lehet tovább játszani
6b	Betöltés egy befejeződött játéknál	Az elmentett játék betöltődik, azzal lehet tovább játszani.
6c	Betöltés, de kiválasztott játék nélkül	Figyelmeztető üzenetablak, de más változás nem történik.
6d	Betöltés elvetése	Nem történik változás.