

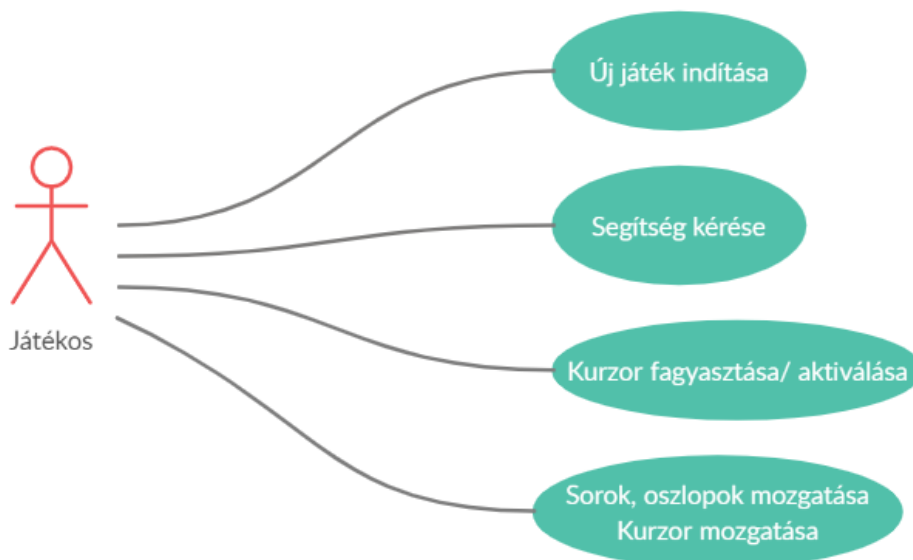
## 7. Rubik tábla

Készítsünk programot, amellyel egy Rubik táblát lehet kirakni.

A Rubik tábla lényegében a Rubik-kocka két dimenziós változata. A játékban egy  $4 \times 4$  mezőből álló táblán 4 különböző színű mező lehet, mindegyik színből pontosan  $n$  darab, kezdetben véletlenszerűen elhelyezve. A játék célja az egyes sorok, illetve oszlopok mozgatásával (ciklikus tologatásával, azaz ami a tábla egyik végén lecsúszik, az ellentétes végén megjelenik) egyszínűvé alakítani vagy a sorokat, vagy az oszlopokat (azaz vízszintesen, vagy függőlegesen csíkokat kialakítani).

A program ismerje fel, ha vége a játéknak. Ekkor jelenítse meg, hány lépéssel győzött a játékos, majd automatikusan kezdjen új játékot.

A játékos 4 különböző tevékenységet végezhet.

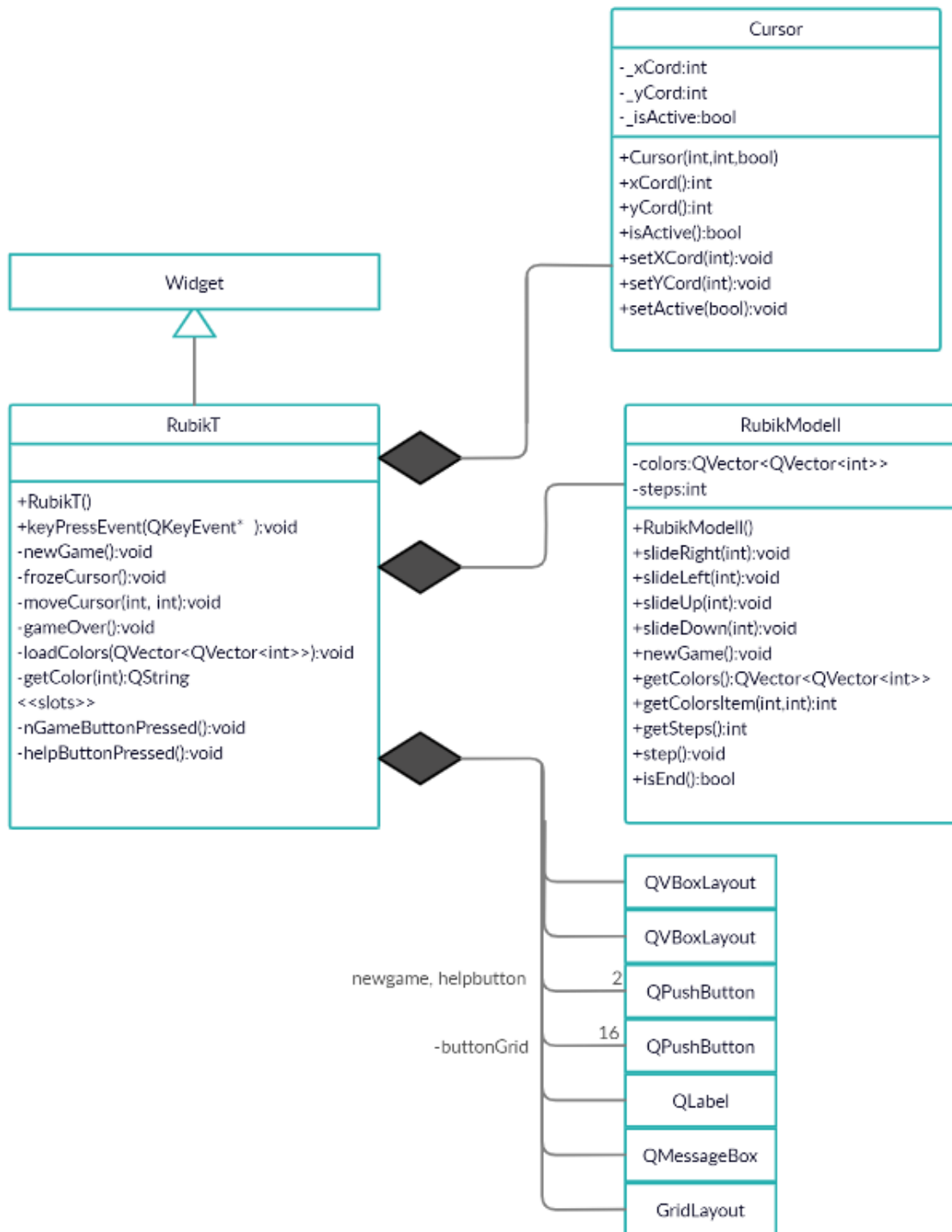


A felhasználói tevékenységek során az alábbi esetek következhetnek be.

1.	Kilépés	GIVEN:	Az alkalmazás meg van nyitva.
		WHEN:	Lezáró ikonra kattintunk.
		THEN:	Az alkalmazás bezáródik.
2.	Játék indítása	GIVEN:	Az alkalmazás meg van nyitva.
		WHEN:	Az „Új játék” gombra kattintunk
		THEN:	A játék elindul / újraindul.
3.	Segítség kérése	GIVEN:	Az alkalmazás meg van nyitva.
		WHEN:	A „Segítség” ikonra kattintunk.
		THEN:	Egy ablak felugrik a magyarázattal.
4.	Kurzor mozgatása	GIVEN:	A játék elindult. A kurzor szabad.
		WHEN:	A nyilakat megnyomjuk.
		THEN:	A kurzor az adott irányba mozdul.
5.	Kurzor rögzítése / elengedése	GIVEN:	A játék elindult.
		WHEN:	Megnyomjuk a SPACE gombot.
		THEN:	A kurzor rögzül / felszabadul.
6. a,	Sor / oszlop csúsztatása	GIVEN:	A játék elindult. A kurzor rögzítve van.
		WHEN:	Megnyomjuk a nyilakat.
		THEN:	A sor / oszlop a kurzor pozíciójába az adott irányba mozdul.
6. b,	Sor / oszlop csúsztatása	GIVEN:	A játék elindult. A kurzor rögzítve van.
		WHEN:	Megnyomjuk a nyilakat.
		THEN:	Ha a tábla ki lett rakva az adott mozgatóval, kiírja a lépések számát és újraindul a játék.

## Osztálydiagram

A fő osztály (RubikT) a Widget-ből származik és a megjelenésért felel. A játék adataiért és azok kezeléséért a RubikModell osztály felel. A kurzor nyilvántartásáért a Cursor osztály felel.



### **Játék indítása**

Amikor a játék elindul a RubikModell véletlen számokat (0 és 4 intervallumban) rendel a mátrixhoz, amiben a tábla adatait tároljuk. Ez alapján a RubikT feltölti színekkel a táblázatot (loadColors()), ami egy QPushButton-okat tartalmazó mátrix.

### **Eseménykezelés**

A felületen két gomb helyezkedik el. Az új játék gomb megnyomásával meghívjuk a RubikT newGame() metódusát. A segítség gomb megnyomásával egy QMessageBox jelenik meg a magyarázószöveggel. A billentyűzet jeleinek fogadásáért a QKeyEvent osztály felel és reagál a nyilakra, valamint a SPACE gombra. A kurzor állapotának megfelelően (aktív / rögzített) a nyilak lenyomásával a kurzor pozícióját módosítjuk vagy pedig az adatmátrix elemeit rendezzük.

A RubikModell minden mozgás után ellenőrzi, hogy a játék véget ért-e (RubikModell::isEnd()). Ha igen, a RubikT-n meghívódik a gameOver() metódus.