Eseményvezérelt alkalmazások fejlesztése 1

1. beadandó – Lovagi Torna

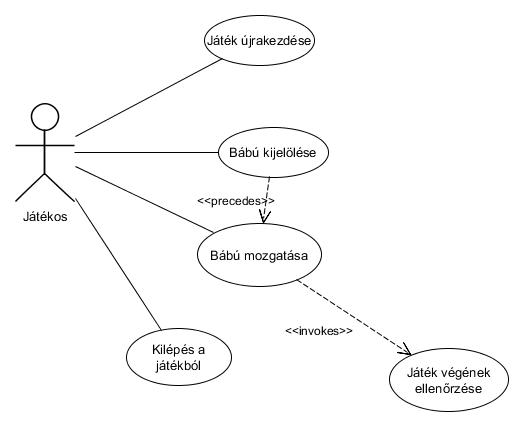
# Feladat

Készítsünk programot, amellyel a következő két személyes játékot lehet játszani. Adott egy n×n mezőből álló tábla, amelynek a négy sarkába 2-2 fehér, illetve fekete ló figurát helyezünk el (az azonos színűek ellentétes sarokban kezdenek). A játékosok felváltva lépnek, a figurák L alakban tudnak mozogni a játéktáblán. Kezdetben a teljes játéktábla szürke színű, de minden egyes lépés után az adott mező felveszi a rá lépő figura színét (bármilyen színű volt előtte). A játék célja, hogy valamely játékosnak függőlegesen, vízszintesen, vagy átlósan egymás mellett 4 ugyanolyan színű mezője legyen. Ha ezt valamelyik játékos elérte, vége a játéknak. A program biztosítson lehetőséget új játék kezdésére a táblaméret megadásával (4×4,6×6,8×8), és ismerje fel, ha vége a játéknak. Ekkor jelenítse meg, melyik játékos győzött, majd automatikusan kezdjen új játékot.!

# Elemzés

* A játék egy grafikus felületen helyezkedik el, amelyen beállítástól függően 16, 36, vagy 64 nyomógomb helyezkedik el, melyek a sakktábla mezőit reprezentálják. A nyomógombokhoz közös eseménykezelő tartozik, mely a gomb pozíciója alapján frissíteni fogja a játék állását
* Az ablak tetején kap helyet egy Restart gomb a játék újrakezdéséhez, és egy legördülő lista, amiből kiválaszthatjuk az új játéktábla méretét
* A játék felülete tetszőleges méretezhető, a minimális méretet a Layout-ok biztosítják
* A játék pillanatnyi állapotát és állapotváltozásai a Game namespace osztályai kezelik

# Használati esetek



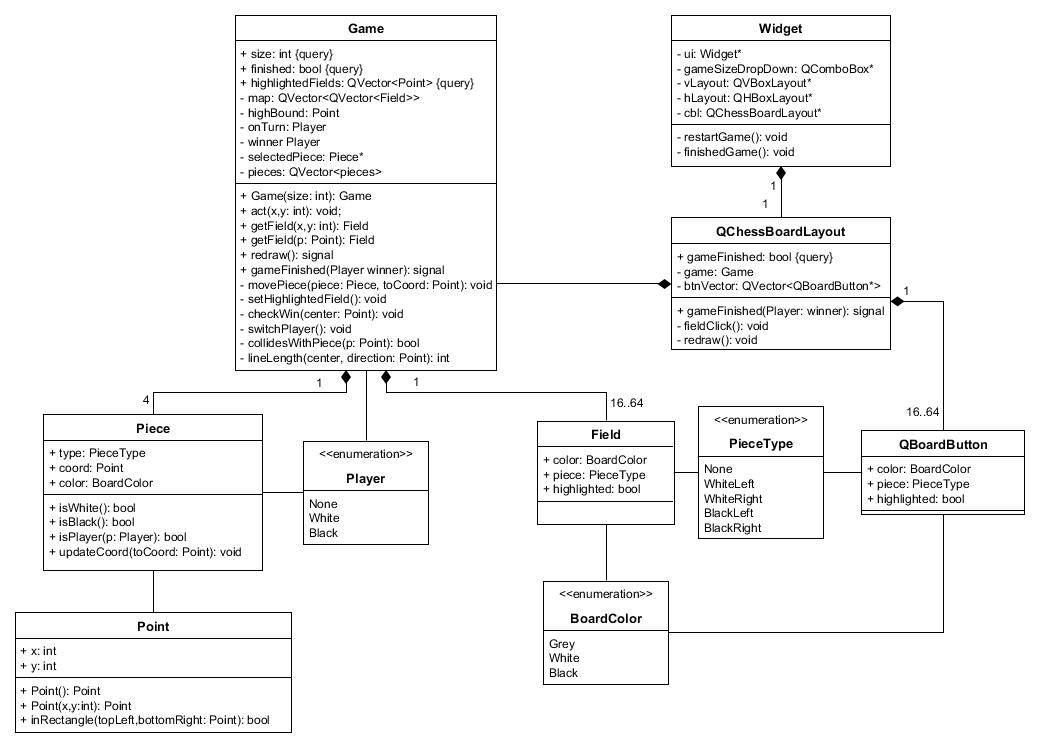
|  |  |  |  |
| --- | --- | --- | --- |
|  | Felhasználói eset | Leírás | |
| 1 | program indítása | GIVEN | a program telepítve van |
| WHEN | elindítom a programot |
| THEN | üres 8×8as tábla jelenik meg |
| 2 | játék vége | GIVEN | a játéknak vége |
| WHEN | újrakezdése kattintok |
| THEN | új játék kezdődik |
| 3 | játék közbeni újrakezdés | GIVEN | a játéknak nincs vége |
| WHEN | újrakezdése kattintok |
| THEN | a program megkérdezi, hogy biztosan újraindítom-e |
| 4 | lépés | GIVEN | én jövök |
| WHEN | kiválasztok egy bábút |
| THEN | a játék megjeleníti az azzal a bábúval elérhető mezőket |
| 5 | lépés | GIVEN | ki van választva egy bábú |
| WHEN | egy helyes mezőre kattintok |
| THEN | a játék az új mezőre mozgatja a bábút |
| 6 | játék vége | GIVEN | én jövök |
| WHEN | olyan helyre lépek, hogy kijön a 4 egyszínű mező vízszintesen, függőlegesen vagy átlósan |
| THEN | megjelenik egy üzenet, mely közli, hogy megnyertem a játékot |

# Tervezés

A játéklogikát a Game névtér osztályai valósítják meg, azon belül pedig a Game nevű osztály felel a játék állapotának számontartásáért és kezeléséért. A pálya egy kétdimenziós QVector lesz, Field elemekkel.

A játék állapotát a QChessBoardLayout jeleníti meg grafikusan, QBoardButton-ök segítségével, melyeknek kezelését magától végzi. A QChessBoardLayout-hoz mindig tartozik pontosan 1 db Game példány, aminek az állapotát megjeleníti, illetve közvetíti felé a felhasználó tevékenységét a QBoardButton kattintásain keresztül.

# Osztályszerkezet



# Eseménykezelés

* QChessBoardLayout
  + fieldClick()
    - kiváltó: a táblán lévő bármely mezőre való kattintás
    - feladata: megállapítani a kattintott mező helyét, és ezzel léptetni a játék állapotát
  + redraw()
    - kiváltó: Game::redraw()
    - feladata: újrarajzolni a táblát a játék állapota alapján
* Widget
  + restartGame
    - kiváltó: a Restart gombra való kattintás
    - feladata: töröli a mostani játékot, és egy újat hoz létre a legördülő listában épp kiválasztott méretekkel. Ha a játéknak még nem volt vége, megjelenít egy párbeszédablakot, hogy biztosan újra akarjuk-e kezdeni a játékot. Ha a játéknak vége volt, akkor az új játék elkészítése automatikusan megtörténik
  + finishedGame
    - kiváltó: QChessBoardLayout::gameFinished
    - feladata: megjeleníteni a győztes játékos nevét egy párbeszédablakon