

Eseményvezérelt alkalmazások fejlesztése I

7. előadás

Összetett grafikus felületű alkalmazások

Giachetta Roberto

<http://people.inf.elte.hu/groberto>

Összetett grafikus felületű alkalmazások

Ablakok

- A grafikus felületű alkalmazásokban a vezérlőket ablakokra helyezzük
- ablaknak minősül bármely vezérlő, amely egy `QWidget`, vagy bármely leszármazottjának példánya, és nincs szülője
- adottak speciális ablaktípusok is, pl.:
 - üzenőablak* (`QMessageBox`), elsősorban üzenetek közlésére, vagy kérdések feltételére
 - dialogusablak* (`QDialog`), amelynek eredménye van, elfogadható (`accept`), vagy elutasítható (`reject`)
 - főablak* (`QMainWindow`), amely számos kiegészítést biztosít összetett ablakok megvalósítására

Összetett grafikus felületű alkalmazások

Főablakok

- A *főablak* (`QMainWindow`) egy olyan speciális ablaktípus, amely megkönnyíti összetett, speciális vezérlőket tartalmazó ablakok létrehozását, úgymint
 - menüsor* (*Menu Bar*): menüpontok gyűjteménye az ablak tetején
 - státuszsor* (*Status Bar*): állapotkijelző sor az ablak alján
 - eszköztár* (*Toolbar*): ikonyűjteményeket tartalmazó funkciógombok, amely az ablak bármely szélére elhelyezhetőek
- Az ablakon belül további vezérlőket helyezhetünk el, amelyeket dokkolhatunk az ablak széléhez, vagy középre

Összetett grafikus felületű alkalmazások

Főablakok



Összetett grafikus felületű alkalmazások

Akciók

- A különböző vezérlők sokszor ugyanazon funkciókat biztosítják más formában (ikon, szöveg, ...)
- A funkciókat egységesen *akcióként* (`QAction`) kezelhetjük, amely
 - rendelkezik felirattal (`text`), ikonnal (`icon`), gyorsbillentyűvel (`shortcut`), segédüzenettel (`statusTip`)
 - lehetőséget ad kijelölésre (`checked`), valamint billentyűs gyorsnavigálásra (az `&` karakterrel)
 - kiváltható billentyűzettel vagy egérrel, a kiváltást esemény (`triggered`)
 - felhelyezhető tetszőleges menüre, illetve eszköztárra

Összetett grafikus felületű alkalmazások

Akciók

- pl.:

```
QAction newAct = new QAction(QIcon("new.png"),
                               tr("Új"), this);
// ikon és név megadása, a j billentyűre
// gyorsnavigál a menüben
newAct->setShortcuts(QKeySequence::New);
// a keretrendszer által kirendelt "új"
// billentyűkombináció
newAct->setStatusTip(tr("Új fájl létrehozása"));
connect(newAct, SIGNAL(triggered()),
        this, SLOT(newFile()));
// eseménykezelő társítás
fileMenu->addAction(newAct); // felhelyezés
fileToolBar->addAction(newAct);
```

| Összetett grafikus felületű alkalmazások | |
|---|-----|
| Menü | |
| <ul style="list-style-type: none"> A menüt (<code>QMenu</code>) a főablak <code>menuBar</code> tulajdonságán keresztül kezelhetjük, a menühöz felvehetünk almenüket, akciókat és elválasztókat (<code>separator</code>) <ul style="list-style-type: none"> a menük tetszőlegesen egymásba ágyazhatóak pl.: <pre> QMenu fileMenu = this->menuBar() ->addMenu(tr("&Fájl")); // új almenü létrehozása fileMenu->addAction(newAct); // menüpont felvétele fileMenu->addSeparator(); // elválasztó fileMenu->addMenu(tr("&Legutóbbi fájlok")); // beágyazott almenü </pre> | |
| ELTE IK, Eseményvezérelt alkalmazások fejlesztése I | 7:7 |

| Összetett grafikus felületű alkalmazások | |
|---|-----|
| Eszköztár | |
| <ul style="list-style-type: none"> Eszköztárakból (<code>QToolBar</code>) tetszőlegesen sokat vehetünk fel, amelyek alapértelmezetten az ablak tetején jelennek meg <ul style="list-style-type: none"> ikonok sorozatát adják, esetleges elválasztókkal szeparálva az eszköztárak alapértelmezés szerint utólag áthelyezhetőek bármely szélére az ablaknak, illetve lehet lebegő (<code>floating</code>) állapotban is pl.: <pre> QToolBar fileBar = this->addToolBar(tr("Fájl")); // új eszköztár felvétele fileBar->addAction(newAct); // új akció felvétele fileBar->addSeparator(); // elválasztó </pre> | |
| ELTE IK, Eseményvezérelt alkalmazások fejlesztése I | 7:8 |

| Összetett grafikus felületű alkalmazások | |
|--|-----|
| Státuszor és tartalom | |
| <ul style="list-style-type: none"> A státuszor (<code>QStatusBar</code>) alapvetően státuszüzenetek kiírására szolgál, ugyanakkor bármilyen vezérlő ráhelyezhető <ul style="list-style-type: none"> üzenetet kiírni a <code>showMessage(<üzenet>)</code> utasítással tudunk, törölni a <code>clearMessage()</code> utasítással pl.: <code>this->statusBar()->showMessage(tr("Kész"));</code> Az ablak területére célszerű egy külön vezérlőben elhelyezni a tartalmat, ez a központi vezérlő (<code>centralWidget</code>) Amennyiben több tartalmat helyeznénk az ablakra, lehetőségünk van azokat dokkolni a <code>QDockWidget</code> osztály segítségével, amelyet az <code>addDockWidget(<vezérlő>)</code> művelettel helyezhetünk az ablakra | |
| ELTE IK, Eseményvezérelt alkalmazások fejlesztése I | 7:9 |

| Összetett grafikus felületű alkalmazások | |
|--|------|
| Alkalmazásszintű tulajdonságok | |
| <ul style="list-style-type: none"> A Qt alkalmazásokat minden esetben egy alkalmazás (<code>QApplication</code>) objektum vezérli, amely számos értéket tárol, úgymint: <ul style="list-style-type: none"> alkalmazás információk (<code>applicationName</code>, <code>organizationName</code>, <code>applicationVersion</code>) környezeti információk (<code>applicationDirPath</code>, <code>arguments</code>, <code>keyboardModifiers</code>, <code>clipboard</code>) grafikus környezeti adatok (<code>allWindows</code>, <code>windowIcon</code>, <code>palette</code>, <code>styleSheet</code>, <code>font</code>) Az alkalmazás értékeihez bárholnan, statikus műveletekkel hozzáférhetünk | |
| ELTE IK, Eseményvezérelt alkalmazások fejlesztése I | 7:10 |

| Összetett grafikus felületű alkalmazások | |
|---|------|
| Alkalmazásszintű tulajdonságok | |
| <ul style="list-style-type: none"> Pl.: <pre> QApplication::setOrganizationName("MySoft"); QApplication::setApplicationName("MyApp"); // beállítunk némi információt ... QString executableName = QApplication::arguments()[0]; // lekérjük a programmevet if (QApplication::arguments().size() > 1) { // ha még van ezen felül argumentum QString arg1 = QApplication::arguments()[1]; } </pre> | |
| ELTE IK, Eseményvezérelt alkalmazások fejlesztése I | 7:11 |

| Összetett grafikus felületű alkalmazások | |
|--|------|
| Beállítások kezelése | |
| <ul style="list-style-type: none"> Nagyobb alkalmazások rendszerint rendelkeznek külön alkalmazásszintű <i>beállításokkal</i>, amelyek célszerű elmentenünk, és újabb futtatáskor betöltenünk A beállítások eltárolhatóak egyedileg, de használhatjuk a beépített <code>QSettings</code> osztályt, amely egyszerűsíti a beállítások kezelését <ul style="list-style-type: none"> a beállítások eltárolásának módja platformonként változik (Linux esetén konfigurációs fájlok, Windows esetén regisztrációs adatbázis), ezt az osztály elfedi, így a programozónak a tárolás módjával nem kell törődnie a beállítások egy adott alkalmazásra és felhasználóra vonatkoznak | |
| ELTE IK, Eseményvezérelt alkalmazások fejlesztése I | 7:12 |

| Összetett grafikus felületű alkalmazások | |
|---|------|
| Beállítások kezelése | |
| <ul style="list-style-type: none"> a beállításokba kulcs/érték párokat vehetünk fel a <code>setValue (<kulcs>, <érték>)</code> utasítással, ahol a kulcs szöveges, az érték tetszőleges <code>QVariant</code> lehet, lekérdezni a <code>value (<kulcs>)</code> utasítással tudunk a <code>contains (<kulcs>)</code> függvény ellenőrzi a kulcs létezését <p>A <code>QVariant</code> egy általános típus, amely a primitív típusokat tudja „becsomagolni”, így az ottani tartalom rendelkezik több konverziós művelettel, pl.:</p> <pre>QVariant vi(123); // létrehozás egészből int i = vi.toInt(); // visszaalakítás egészre QVariant vc = QColor(15, 20, 200); // színből QColor c = vc.value<QColor>(); // vissza színbe</pre> | |
| ELTE IK, Eseményvezérelt alkalmazások fejlesztése I | 7:13 |

| Összetett grafikus felületű alkalmazások | |
|--|------|
| Beállítások kezelése | |
| <ul style="list-style-type: none"> Pl.: <pre>QString value; ... QSettings settings("MySoft", "MyApp"); // beállítások létrehozása (ha korábban // beállítottuk az alkalmazás információkat, // használhatunk alapértelmezett konstruktort) settings.setValue("myValue", value); // érték beállítása ... settings.value("myValue").toString(); // visszakerjünk az értéket és szöveggé // alakítjuk</pre> | |
| ELTE IK, Eseményvezérelt alkalmazások fejlesztése I | 7:14 |

| Összetett grafikus felületű alkalmazások | |
|--|------|
| Erőforrások | |
| <ul style="list-style-type: none"> A főablakon használt akciókat célszerű ellátni ikonokkal, amelyeket az alkalmazáshoz kell, hogy csatoljunk Az alkalmazáshoz használt ikonokat és egyéb nem kód tartalmat lehetőségünk van erőforrásként (<i>resource</i>) csatolni az alkalmazáshoz <ul style="list-style-type: none"> az erőforrások tartalma belefordul a futtatandó állományba, így nem kell külön másolni őket az erőforrásokat a projekthez tartozó <code>.qrc</code> fájlban nevezhetjük meg az erőforrásként megadott fájlokat a <code>:<elérési útvonala></code> hivatkozással hívhatjuk be, pl.: <pre>QIcon(":/images/new.png"); // erőforrás elérése</pre> | |
| ELTE IK, Eseményvezérelt alkalmazások fejlesztése I | 7:15 |

| Összetett grafikus felületű alkalmazások | |
|---|------|
| Példa | |
| <p><i>Feladat:</i> Készítsünk egy <i>Memory</i> kártyajátékot, amelyben két játékos küzd egymás ellen. A játékoson kártyapárok találhatóak, és a játékosok feladata ezek megtalálása.</p> <ul style="list-style-type: none"> a játék különböző kártyacsomagokkal játszható, amelyek könyvtárakból tölthetők be, minden ilyen könyvtárban található egy <code>name.txt</code>, ami a csomag nevét tartalmazza, és tetszőleges számú kép (ezek a kártyák), valamint egy hátlap (<code>back</code> fájlnevel) lehetőségünk van egy beállító ablakban megadni a kiválasztott kártyacsomagot, valamint a játéktábla méretét (csak páros méretű, de legalább 4 kártyából álló lehet), valamint a játékosok neveit | |
| ELTE IK, Eseményvezérelt alkalmazások fejlesztése I | 7:16 |

| Összetett grafikus felületű alkalmazások | |
|---|------|
| Példa | |
| <ul style="list-style-type: none"> kezdetben minden kártya le van fordítva, a játékosok felváltva lépnek, minden lépésben felfordíthatnak két kártyát amennyiben a kártyák egyeznek, úgy felfordítva maradnak és a játékos ismét léphet, különben 1 másodperc múlva visszafordulnak, és a másik játékos következik a játékot az nyeri, aki több kártyapárt talált meg megnyert játékok számát göngyöltve jelenítjük meg, amíg új játékosokat nem állítunk be a felületen folyamatosan megjelenítjük a játékosok adatait (sikeres, sikertelen lépések száma, megnyert játszmák száma) | |
| ELTE IK, Eseményvezérelt alkalmazások fejlesztése I | 7:17 |

| Összetett grafikus felületű alkalmazások | |
|--|------|
| Példa | |
| <p><i>Felhasználói esetek:</i></p> <pre> graph TD User((Felhasználó)) UjJatek((Új játék)) Kilapas((Kilépés)) Beallitasok((Beállítások)) Eloskar (Első kártya felfordítása) Masodikkar (Második kártya felfordítása) Nevemegad (Nevék megadása) Tablameret (Táblaméret megadása) Kartacsomag (Kártyacsomag megadása) User --> UjJatek User --> Kilapas User --> Beallitasok UjJatek --> Eloskar UjJatek --> Masodikkar Beallitasok --> Nevemegad Beallitasok --> Tablameret Beallitasok --> Kartacsomag Eloskar -.-> <pre><code></code></pre> Masodikkar Eloskar -.-> <pre><code></code></pre> Beallitasok Masodikkar -.-> <pre><code></code></pre> Beallitasok </pre> | |
| ELTE IK, Eseményvezérelt alkalmazások fejlesztése I | 7:18 |

Összetett grafikus felületű alkalmazások

Példa

Tervezés (architektúra):

- a játékot kétrétegű architektúrában valósítjuk meg
- a modell tartalmazza:
 - magát a játékot, amit egy kezelőosztály felügyel (**GameManager**), valamint hozzá segédosztályként a játékost (**Player**)
 - a kártyacsomagokat (**CardPack**)
- a nézet tartalmazza:
 - a játék főablakát (**MainWindow**), amely tartalmaz egy menüt és egy státuszsort
 - a beállítások segédablakát (**ConfigurationDialog**)

ELTE IK, Eseményvezérelt alkalmazások fejlesztése I

7:19

Összetett grafikus felületű alkalmazások

Példa

- a játékelületet megjelenítő vezérlőt (**GameWidget**), amely tartalmazza a játékmóddel kapcsolatos tevékenységeket
- ehhez segédosztályként a felhasználói információkat kiíró vezérlőt (**PlayerStatusWidget**, ezt előléptetett vezérlővel állítjuk be a felülettervezőben), valamint a képet megjeleníteni tudó egyedi gombot (**ImageButton**)
- a nézet a modell publikus műveleteit hívja, és eseményeket is kaphat tőle
- egy csomag kártyát erőforrásként csatolunk az alkalmazáshoz (**packs.qrc**), hogy mindig legyen legalább egy csomag kártya

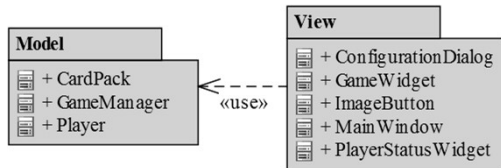
ELTE IK, Eseményvezérelt alkalmazások fejlesztése I

7:20

Összetett grafikus felületű alkalmazások

Példa

Tervezés (architektúra):



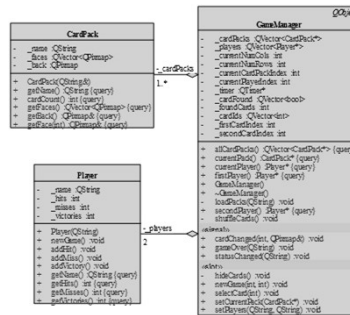
ELTE IK, Eseményvezérelt alkalmazások fejlesztése I

7:21

Összetett grafikus felületű alkalmazások

Példa

Tervezés (modell):



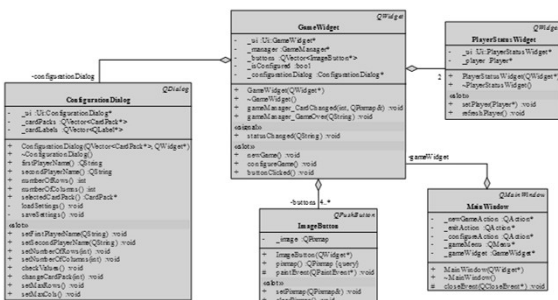
ELTE IK, Eseményvezérelt alkalmazások fejlesztése I

7:22

Összetett grafikus felületű alkalmazások

Példa

Tervezés (nézet):



ELTE IK, Eseményvezérelt alkalmazások fejlesztése I

7:23

Összetett grafikus felületű alkalmazások

Példa

Tervezés (dinamikus):

- új játék indításához először a főablakban (**MainWindow**) kell kiváltanunk (**triggered**) a megfelelő akciót (**newGameAction**)
- ennek hatására a főablak új játékot indít (**newGame**) a játék nézetében (**GameWidget**)
- a nézet beállítja a játék paramétereit (**configureGame**)
- a nézet létrehozza az új játékot (**newGame**) a modellben (**GameManager**)
- a modell megkeveri a kártyákat (**shuffleCards**), majd eseménnyel jelzi az állapot változását (**changeStatus**)

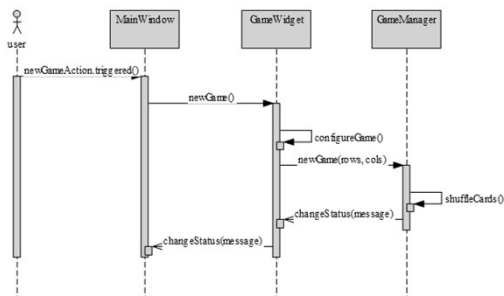
ELTE IK, Eseményvezérelt alkalmazások fejlesztése I

7:24

Összetett grafikus felületű alkalmazások

Példa

Tervezés (dinamikus):



ELTE IK, Eseményvezérelt alkalmazások fejlesztése I

7:25

Összetett grafikus felületű alkalmazások

Példa

Tervezés (dinamikus):

- amennyiben új játékot kezdünk (**newGame**), a felület aktív lesz, játék végén (**gameOver**) pedig inaktívvá válik
- a játék modellje kezdetben egy kártyát sem mutat, de új játék kezdésekor (**newGame**) az összes kártyát megmutatja, majd automatikusan elrejtő őket (**hideCards**)
- kiválasztás (**selectCard**) hatására előbb egyet, majd kettőt megmutathat (**cardChanged**)
- amennyiben a két kártya egyezik, és minden kártyát felfedtünk, vége a játéknak (**gameOver**)

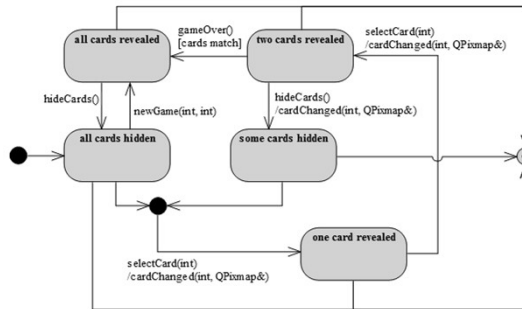
ELTE IK, Eseményvezérelt alkalmazások fejlesztése I

7:26

Összetett grafikus felületű alkalmazások

Példa

Tervezés (dinamikus):



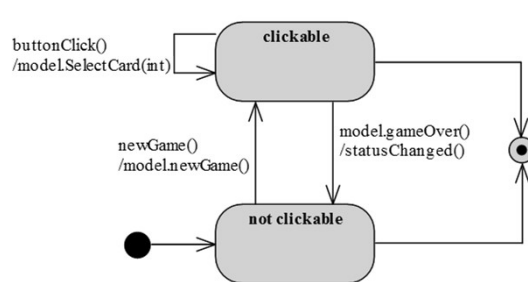
ELTE IK, Eseményvezérelt alkalmazások fejlesztése I

7:27

Összetett grafikus felületű alkalmazások

Példa

Tervezés (dinamikus):



ELTE IK, Eseményvezérelt alkalmazások fejlesztése I

7:28

Összetett grafikus felületű alkalmazások

Példa

Megvalósítás (mainwindow.cpp):

```

...
MainWindow::MainWindow() {
    ...
    connect(newGameAction, SIGNAL(triggered()),
            gameWidget, SLOT(newGame()));
    connect(configureAction, SIGNAL(triggered()),
            gameWidget, SLOT(configureGame()));
    ...
    connect(gameWidget, SIGNAL(statusChanged(
        QString)), this->statusBar(),
            SLOT(showMessage(QString)));
    // állapotváltás a játékban
}

```

ELTE IK, Eseményvezérelt alkalmazások fejlesztése I

7:29

Összetett grafikus felületű alkalmazások

Példa

Megvalósítás (gamewidget.cpp):

```

...
GameWidget::GameWidget(QWidget *parent) : ... {
    ...
    connect(manager,
            SIGNAL(statusChanged(QString)), this,
            SIGNAL(statusChanged(QString)));
    // a logikai réteg eseménye egy újabb
    // eseményt vált ki
    connect(manager,
            SIGNAL(statusChanged(QString)),
            ui->firstPlayerStatus,
            SLOT(refreshPlayer()));
    ...
}

```

ELTE IK, Eseményvezérelt alkalmazások fejlesztése I

7:30

Összetett grafikus felületű alkalmazások

Példa

Megvalósítás (gamemanager.cpp):

```
...
void GameManager::newGame(int numRows,
                          int numCols){
    ...
    statusChanged(trUtf8("Új játék elindítva, ") +
                  players[currentPlayerIndex]->getName() +
                  trUtf8(" következik.));
    // esemény kiváltása
}
}
```