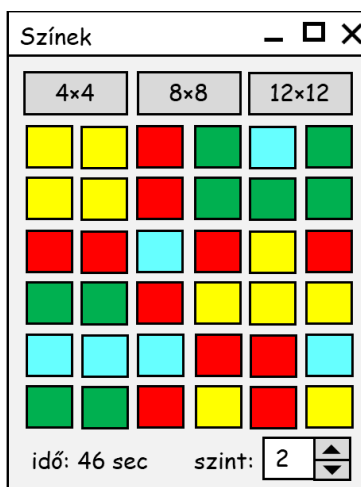


5. hét

Cél: egy zárthelyi feladat megoldása modell-nézet architektúrával.

Színváltó rács

Egy $n \times n$ -es játéktábla ($n = 4, 8$, vagy 12) mezőit 4 különböző színnel (piros, sárga, zöld, kék) véletlenszerűen színezzük ki. Rögzítjük a színek között a fenti sorrendet. Amikor egy mezőre kattintunk, akkor annak és az összes szomszédjának (tehát egy 3×3 -as területen összesen 9 mezőnek – kivéve, ha a tábla szélén jelölünk ki mezőt) a színe a soron következő színre változik: pirosról sárgára, sárgáról zöldre, zöldről kékre, kékről pirosra. A cél az, hogy a teljes játéktáblát egyszínűvé alakítsuk, minél rövidebb idő alatt. Ez eltelt időt a felületen megjelenítjük. A játékban van egy nehezítés: meghatározott lépésenként (ez a játék nehézségi szintjétől függ) történik egy színléptetés egy 3×3 -as (véletlenszerűen kiválasztott) területen.¹



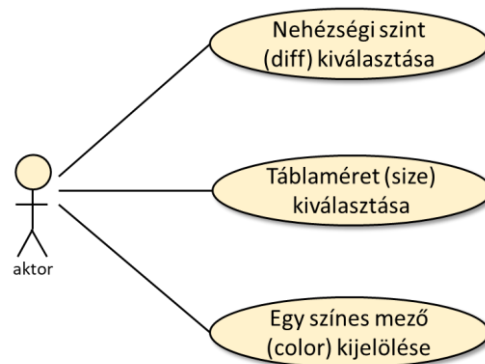
¹ A zárthelyiben egy ilyen a feladat négy szintre van tagolva. **Elégségesért** a programnak meg kellett jeleníteni a játéktáblát, amelynek mérete megváltozik a megfelelő nyomógombra kattintással, és ilyenkor új játék indul. Egy mező kijelölésével az adott mezőnek, illetve a szomszédjainak a színe eggyel tovább lép. A játék kezdetekor a táblaállást úgy kell beállítani, hogy az egy teljesen piros színű táblából n darab véletlenszerűen kiválasztott lépés alkalmazásával álljon elő. **Közepesért** a programnak számolni kell a játék időt másodpercben, amelyet minden kattintásra még 10 másodperccel meg kellett növelni. **Jó** az értéke annak a programnak, amelyik azt is felismeri, hogy vége a játéknak, ezt egy üzenet-ablakban jelzi, majd automatikusan új játékot kezd. **Jeles** eléréséhez a játék nehézségét is lehet állítani. A könnyű fokozat megegyezik az eddigiekkel. Közepes fokozaton minden 10. lépés után, míg nehéz fokozaton minden 4. lépés után egy véletlen kiválasztott mezőnek és szomszédjainak a színe eggyel tovább lép.

Projekt létrehozása

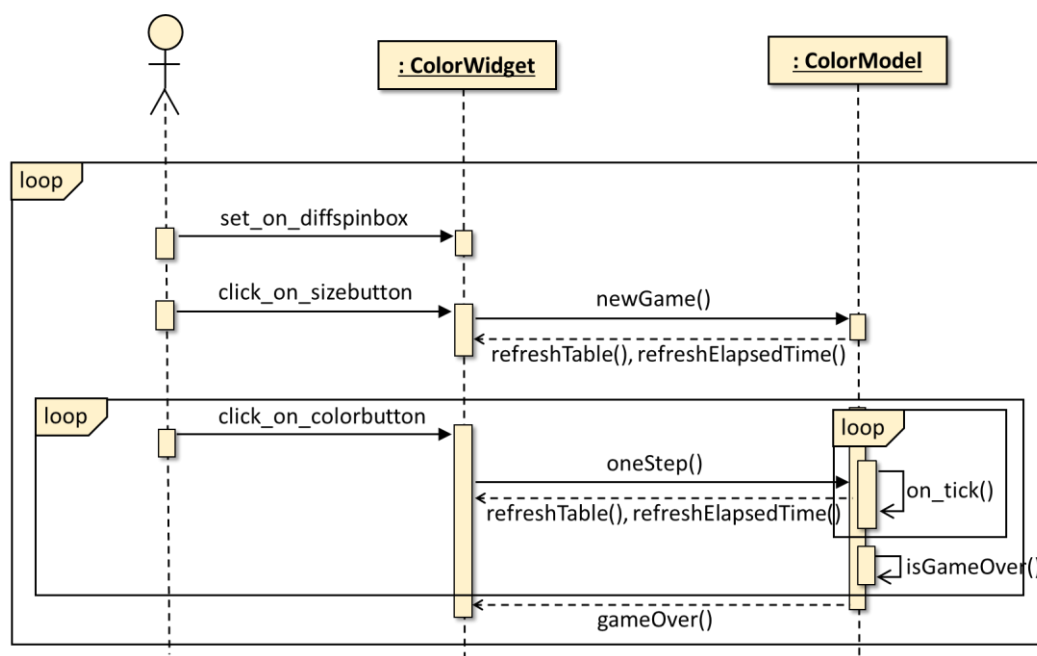
Hozzunk létre egy *Qt Widget Application* projektet. A főablakunk őssztálya legyen a *QWidget*, amelynek grafikus felületét a programkódból építjük fel, ezért a *QtCreator*-ban ne pipáljuk be a *Generate form* lehetőséget.

Architektúra

Egy modell-nézet architektúrát valósítunk meg. Ebben a felhasználó jelzéseit (*set_on_diffspinbox*, *click_on_sizebutton*, *click_on_colorbutton*) a nézet dolgozza fel.

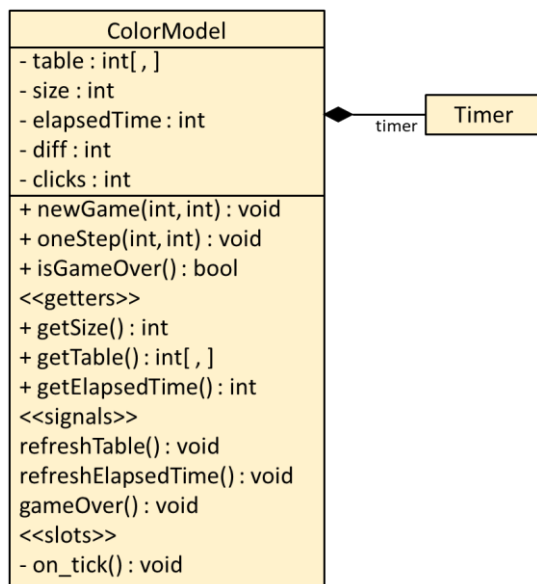


A nézet, miután példányosítja és aggregálja a modellt, a felhasználó jelzéseinek hatására közvetlenül hívhatja annak metódusait, valamint feldolgozza a modell által küldött szignálokat.



Modell

A modell a játéktáblát egy $n \times n$ -es mátrixszal ábrázolja (*table*), amelynek elemei a színek kódjai (1 - piros, 2 - sárga, 3 - zöld, 4 - kék). A *size* ezen négyzetes mátrix oldalának mérete, a *diff* a játék nehézségi szintjét jelzi. Az *elapsedTime* az aktuális játékban eddig eltelt másodperceket, a *clicks* az eddigi kattintások számát mutatja. A modell tartalmaz egy időzítőt (*timer*) is.



A konstruktor inicializálja a méretet (*size*), a véletlenszám generátort, példányosítja az időzítőt (*timer*), amelyhez hozzárendeli (*connect*) az *on_tick()* eseménykezelőt, amelyik másodpercenként megnöveli eggyel az aktuális játékban eltelt időt (*elapsedTime*), és ezt jelzi a nézetnek is.

<i>küldő</i>	<i>signal</i>	<i>fogadó</i>	<i>slot</i>
<i>timer</i>	<i>timeout()</i>	<i>modell</i>	<i>on_tick()</i>

A ***newGame()*** beállítja a méretet (*size*), a nehézségi szintet (*diff*), létrehoz egy $n \times n$ -es mátrixot (*table*), lenullázza az eddigi kattintások számát (*clicks*), az eltelt időt (*elapsedTime*) és elindítja az időzítőt (*timer*), amely másodpercenként növeli az eltelt időt (*on_tick()*). Jelzi a nézetnek, hogy frissíteni kell a táblát (*refreshTable()*) és a játékidőt (*refreshElapsedTime()*).

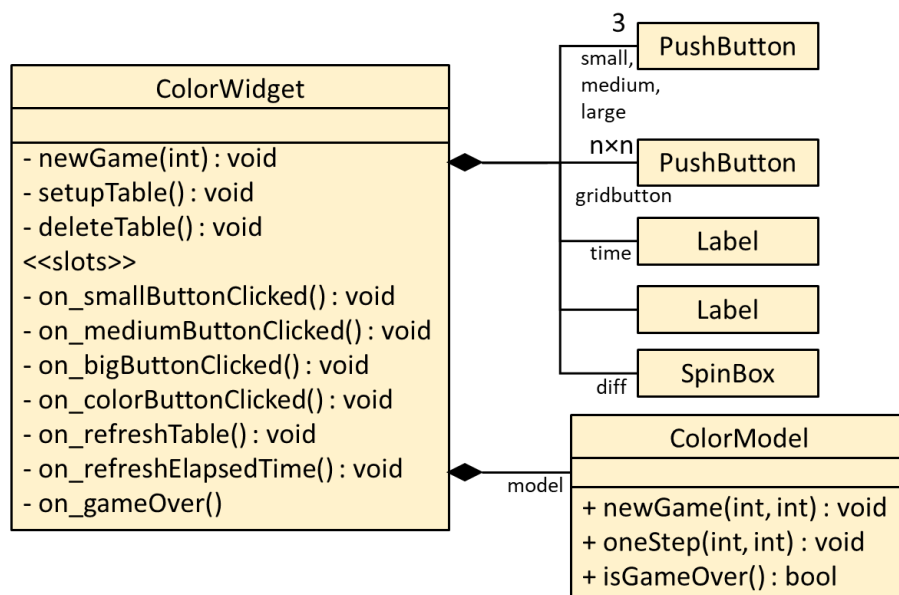
A ***oneStep()*** a tábla megadott pozíciójának 3×3 -as környezetében léptet a mezők színét, eggyel növeli az eddigi kattintások számát (*clicks*), hozzáad 10-et az eltelt időhöz (*elapsedTime*). A játék nehézségi szintjétől (*diff*) függően végrehajt egy extra lépést a tábla véletlen választott mezőjén. Jelzi a nézetnek, hogy frissíteni kell a táblát (*refreshTable()*) és a játékidőt (*refreshElapsedTime()*). Minden esetben ellenőrzi (*isGameOver()*), hogy vége van-e a játéknak, és ha igen, azt jelzi a nézetnek (*gameOver()*).

Az ***isGameOver()*** vizsgálja, hogy a tábla egyszínű-e.

Nézet

A nézet (*widget*) statikus elemei a méretet változtató nyomógombok (közös néven: *sizebutton*), a nehézségi szint megadására szolgáló színbeállító (*diffspinBox*), és az eltelt időt mutató címke.

Dinamikusan változhat a mérete a felületen mátrixszerűen elhelyezett színes nyomógomboknak (közös néven: *colorbutton*).



A konstruktor létrehozza és felteszi a nézetre a statikus vezérlőket, példányosítja a modellt, hozzáköti (*connect*) annak szignáljaihoz a megfelelő eseménykezelőket (*on_refreshTable()*, *on_refreshElapsedTime()*, *on_gameOver()*). A játék méretét beállító (*smallbutton*, *mediumbutton*, *bigbutton*) nyomógombok kattintás eseményéhez hozzáköti azok eseménykezelőit (*on_smallButtonClicked()*, *on_mediumButtonClicked()*, *on_bigButtonClicked()*). Végül elindít egy 8×8-as játékot.

küldő	signal	fogadó	slot
<i>smallbutton</i>	<i>clicked()</i>	<i>nézet</i>	<i>on_smallButtonClicked()</i>
<i>mediumbutton</i>	<i>clicked()</i>	<i>nézet</i>	<i>on_mediumButtonClicked()</i>
<i>bigbutton</i>	<i>clicked()</i>	<i>nézet</i>	<i>on_bigButtonClicked()</i>
<i>modell</i>	<i>refreshTable()</i>	<i>nézet</i>	<i>on_refreshTable()</i>
<i>modell</i>	<i>refreshElapsedTime()</i>	<i>nézet</i>	<i>on_refreshElapsedTime()</i>
<i>modell</i>	<i>gameOver()</i>	<i>nézet</i>	<i>on_gameOver()</i>
<i>colorbutton</i>	<i>clicked()</i>	<i>nézet</i>	<i>on_colorButtonClicked()</i>

A ***setupTable()*** hozza létre a játék színes nyomógombjait, kiszámítja és rögzíti a felületet méretét úgy, hogy a gombok elférjenek, és elhelyezi a gombokat rácselrendezésben. Mindegyik gomb kattintás eseményéhez ugyanazt az eseménykezelőt (***on_colorButtonClicked()***) köti hozzá.

A ***deleteTable()*** eltávolítja a rácselrendezőről a színes gombokat és meg is szünteti azokat.

A ***newGame()***, ha változik a játék mérete, lebontja (***deleteTable()***), majd újraépíti (***setupTable()***) a színes nyomógombokat, és meghívja a modell ***newGame()*** eljárását.

Az ***on_smallButtonClicked()***, ***on_mediumButtonClicked()***, és ***on_bigButtonClicked()*** új játékot indít (***newGame()***).

Az ***on_colorButtonClicked()*** hívja meg a modell ***oneStep()*** metódusát, miután azonosította, hogy hányadik sor hányadik oszlopának nyomógombján kattintott a felhasználó.

Az ***on_refreshTable()*** a modellnek megfelelően átszínezi a felület színes nyomógombjait.

Az ***on_refreshElapsedTime()*** frissíti a felület alján megjelenő eltelt időt.

Az ***on_gameOver()*** hatására egy üzenet-ablakban kapunk tájékoztatást arról, hogy sikerült megoldani a játékot, majd ezt követően új játék indul.