

Készítette:

Bodó József

Email: i7p4uq@inf.elte.hu

Csoportszám: 7

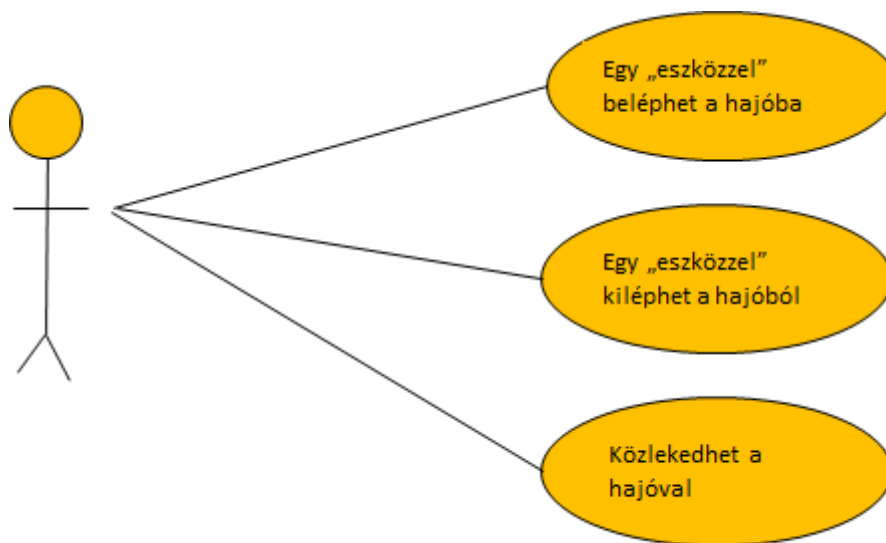
Feladat:

Készítsünk programot, amely bemutatja a farkas-kecske-káposzta problémát. Adott egy folyó, amelynek az egyik partján egy farkas, egy kecske, egy káposzta található. Átkelésükhöz adott továbbá egy csónak, amely a csónakoson kívül legfeljebb egy dolgot (farkast, kecskét, vagy káposztát) tud egyszerre szállítani. Az átkelések alkalmával nem szabad, hogy valamelyik parton ott maradjon a farkas a kecskével, vagy a kecske a káposztával.

Jelenítsük meg a játék egy-egy állapotát: farkast, kecskét vagy káposztát valamelyik parton vagy a csónakban, illetve a csónak helyét: egyik vagy másik part, illetve a folyó közepe. Lehessen a két partról a csónakba helyezni, valamint visszahelyezni farkast, kecskét, vagy káposztát, illetve kezdeményezni átkelést, amely csak akkor történik meg, ha a játékállás a feltételeket az átkelés során és után is kielégíti. A program ismerje fel, ha vége a játéknak. Ekkor jelenítse meg, hány lépéssel (átkeléssel) oldották meg a problémát, majd kezdjen automatikusan új játékot.

A feladat elemzése:

A játékban egy játékos vesz részt, így az alkalmazásban is csak egy felhasználó lesz. A felhasználó háromféle tevékenységet végezhet:

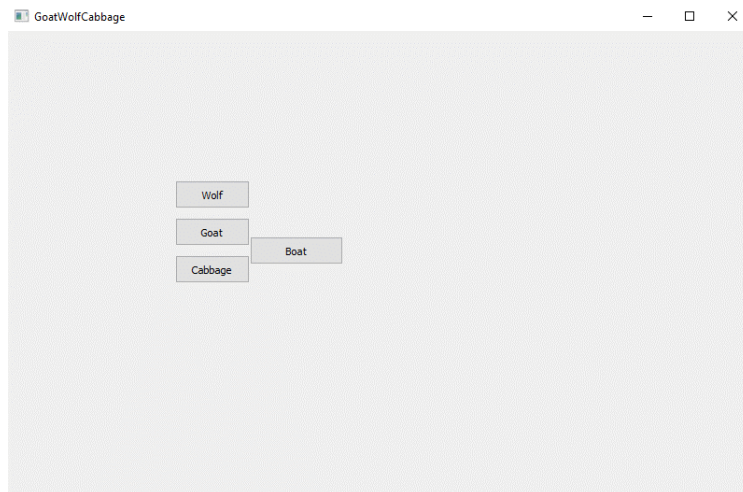


A felhasználói tevékenységek során az alábbi esetek következhetnek be.

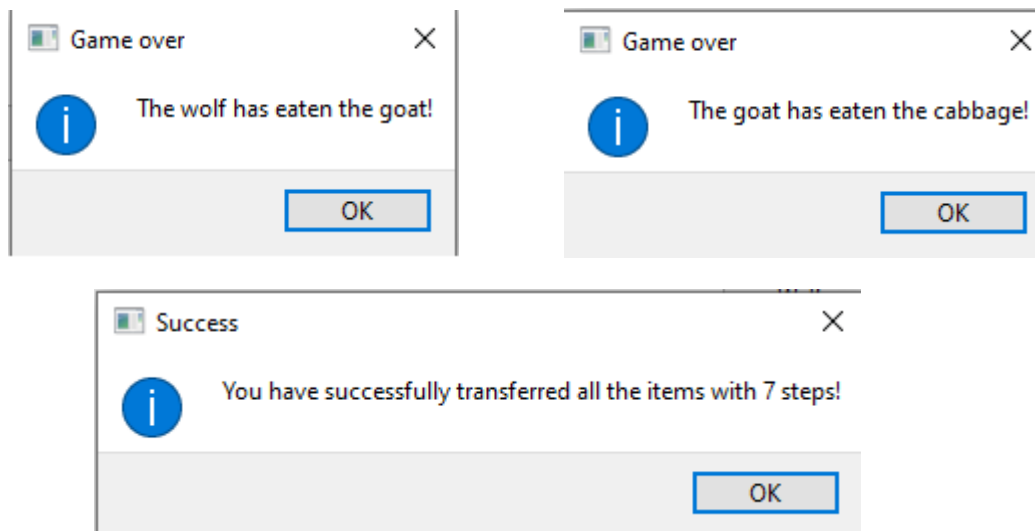
1	Alkalmazás indítása	GIVEN:	Az alkalmazás telepítve van.
		WHEN:	Alkalmazás indítása.
		THEN:	Megjelenik a játéktábla.
2	Kilépés	GIVEN:	A játéktábla aktív.
		WHEN:	A játéktábla ablakának lezáró ikonjára kattintunk.
		THEN:	Az alkalmazás bezárul
3	Eszköz berakás	GIVEN:	A játéktábla aktív, és a hajó azon az oldalon van.
		WHEN:	Az egyik eszközre kattintunk.
		THEN:	Az eszköz a hajóba kerül.
4	Eszköz kitevés	GIVEN:	A játéktábla aktív, és a hajó azon az oldalon van.
		WHEN:	Arra az eszközre kattintunk, ami éppen a hajóban van.
		THEN:	Az eszköz a partra kerül, éppen arra, amelyik oldalon a hajó áll.
5a	Közlekedés a hajóval	GIVEN:	A játéktábla aktív és a hajó üres.
		WHEN:	A hajóra kattintunk, a hajó átutazik az éppen másik partra úgy, hogy a bal parton még mindhárom eszköz ott van.
		THEN:	A játék véget ér, kapunk egy jelzést arról, hogy miért lett vége, majd újraindul.
5b	Közlekedés a hajóval	GIVEN:	A játéktábla aktív és a hajó üres.
		WHEN:	A hajóra kattintunk, a hajó átutazik az éppen másik partra úgy, hogy a két parton nem sérül a játék feltétele.
		THEN:	Eggyel megnövekszik a lépésszámláló.
5c	Közlekedés a hajóval:	GIVEN:	A játéktábla aktív és a hajóban van utas.
		WHEN:	A hajóra kattintunk, a hajó átutazik az éppen másik partra és vele együtt az utas is, de a másik parton sérül a játék feltétele.
		THEN:	A játék véget ér, kapunk egy jelzést arról, hogy miért lett vége, majd újraindul.
5d	Közlekedés a hajóval:	GIVEN:	A játéktábla aktív és a hajóban van utas.
		THEN:	A hajóra kattintunk, a hajó átutazik az éppen másik partra és vele együtt az utas is, és nem sérül a játék feltétele.
		THEN:	A játékos dönthet, hogy kirakja-e a partra az utast, avagy sem. Megnöveljük eggyel a lépésszámlálót.

Felhasználói felület terve:

A játék ablakán 4 gomb jelenik meg. Három gomb szimbolizálja az eszközöket, azaz a Farkast, a Kecskét és a Káposztát, ezek egymás alatt helyezkednek az egyik oldalt, míg a negyedik gomb szimbolizálja a hajót, amely a két oldal között helyezkedik el. Ezenkívül üzenet-ablakok jelennek meg, amivel jelezzük a játék végét: vagy azt, hogy a felhasználónak nem sikerült átvinnie a három eszközt, vagy azt, hogy sikerült neki.

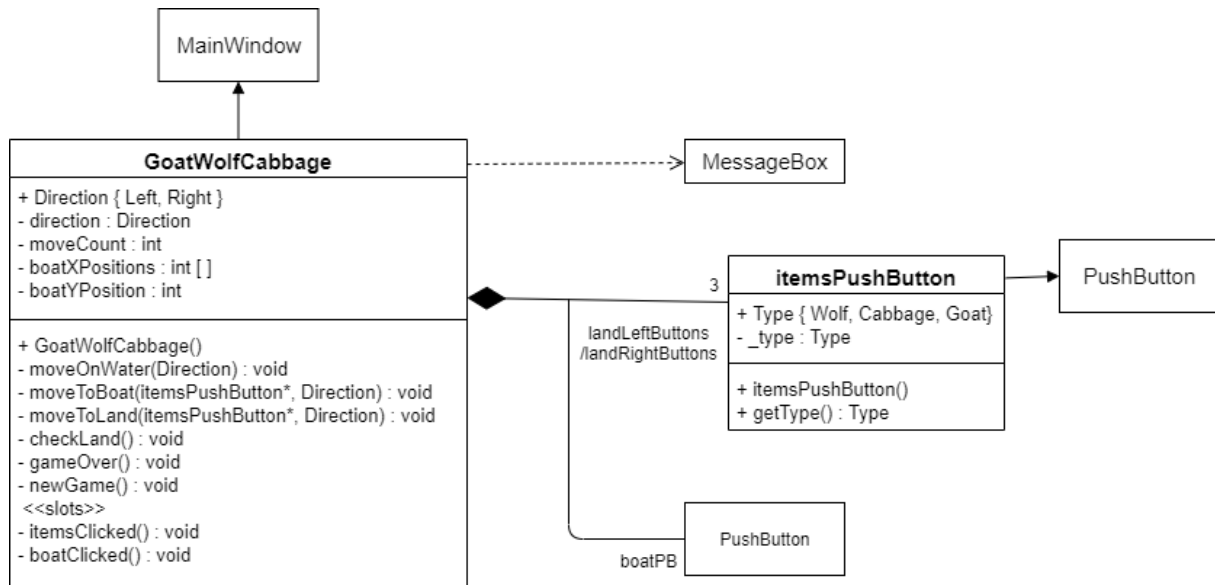


A játék végét üzenet-ablakok jelzik:



Osztálydiagram:

Az alkalmazást egyetlen osztállyal írjuk le, kiegészítve egy a QPushButton-ből leszármaztatott osztállyal. Ez biztosítja mind a nézetet, mind a játéklógikát.



Vezérlők és adattagok:

Az osztály a *QMainWindow*-ből származik majd. A nyomógombok helyét beégetve tároljuk, az alkalmazás méretét nem lehet változtatni. Az *itemsPushButton* osztály a *QPushButton*-ből származik és annyival egészíti ki a őosztályt, hogy tartalmaz egy *Type* adattagot. Az eszközök típusa lesz ilyen fajta nyomógomb és ezeket a játék futása elején a *landLeftButtons* *QVector* típusú halmazba tesszük, majd a játék futása során kerülhetnek át a *landRightButtons*-be.

A *direction* adattag segítségével jelöljük, hogy a hajó éppen melyik oldalt áll. Ez egy felsorolási típus, amely két értéket vehet fel: Bal vagy Jobb.

A *moveCount* adattag segítségével számoljuk, hogy a felhasználó hány lépésnél tart és ezt jelenítjük meg, amikor a játékot sikerült teljesíteni.

Metódusok:

A funkcióknak megfelelően bevezetünk egy *moveOnWater()*, *moveToLand()* és *moveToBoat()* metódusokat. Ezek végrehajtásukban az irány szerint kétfelé válnak. Külön metódust vezetünk be a játékalás ellenőrzésére: megsértettük-e a játék futásának feltételét, ezt a *checkLand()* ellenőrzi. A *gameOver()* függvény hívásakor azt ellenőrizzük, hogy sikerült-e végig játszani a játékot. A *newGame()* metódus feladata, hogy mindent visszaállítson az eredeti helyére, illetve a lépésszámlálót lecsökkentse 0-ra.

Eseménykezelés:

Az eszközök gombjaihoz közös eseménykezelőt rendelünk, amely egérekattintás hatására eldönti, hogy a hajóra vagy a szárazföldre kell-e tenni az eszközt. Mivel a hajón egyszerre csak egy eszköz lehet, így amikor a hajóra helyezzük az eszközt, a többi eszközt a parton letiltjuk. A hajó eseménykezelője egérekattintás után az aktuális irány alapján eldönti, merre fog közlekedni a hajó, majd meghívja az ellenőrző függvényt.

sender	signal	receiver	slot
landLeftButtons[i]	clicked()	nézet	itemsClicked()
boatPB	clicked()	nézet	boatClicked()