

# Készítette

Baranyai Dominik

E-mail: [bdominik0914@gmail.com](mailto:bdominik0914@gmail.com)

Csoportszám: 17

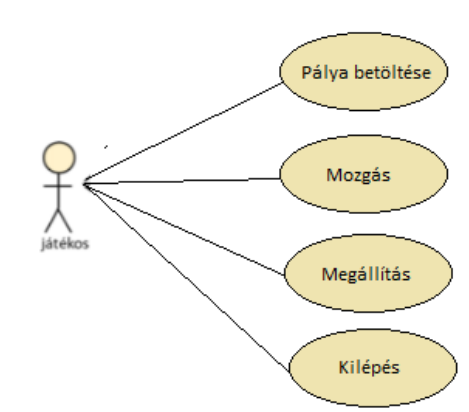
## Feladat

2.beadandó / 11.feladat

### Labirintus

Készítsünk programot, amellyel a következő játékot játszhatjuk. Adott egy  $n \times n$  elemből álló játékpálya, amely labirintusként épül fel, azaz fal, illetve padló mezők találhatók benne, illetve egy kijárat a jobb felső sarokban. A játék célja az, hogy egy játékost minél előbb kivezessünk a labirintusból a bal alsó sarokból indulva. A labirintusban nincs világítás, csak egy fáklyát visz a játékos, amely a 2 szomszédos mezőt világítja meg (azaz egy  $5 \times 5$ -ös négyzetet), de a falakon nem tud átvilágítani. A játékos figurája kezdetben a bal alsó sarokban helyezkedik el, és vízszintesen, illetve függőlegesen mozoghat (egyesével) a pályán. A pályák méretét, illetve felépítését (falak, padlók) tároljuk fájlban. A programot legalább 3 különböző méretű pályával lehessen használni. A program biztosítson lehetőséget új játék kezdésére a pálya kiválasztásával, valamint játék szüneteltetésére (ekkor nem telik az idő, és nem léphet a játékos), továbbá ismerje fel, ha vége a játéknak. A program játék közben folyamatosan jelezze ki a játékidőt.

### A feladat elemzése



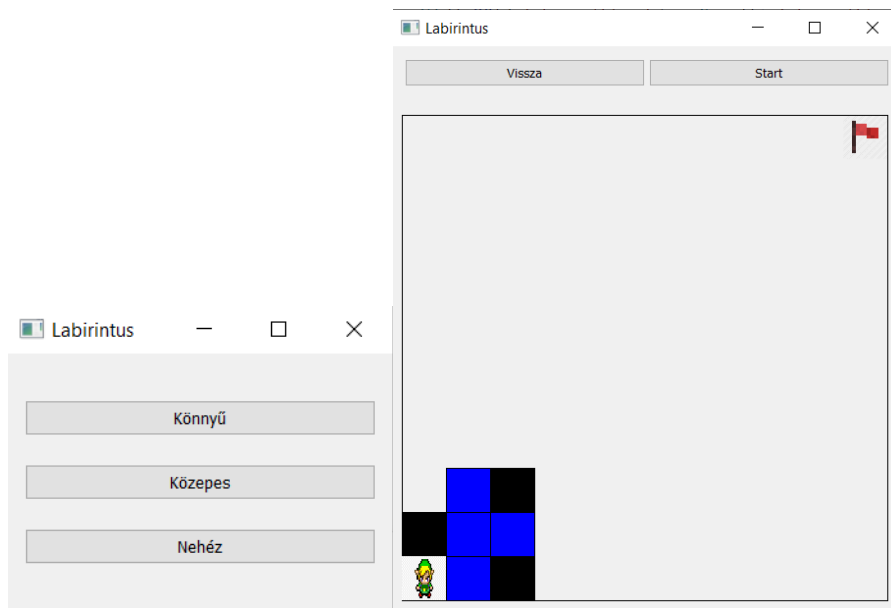
A felhasználói tevékenységek során az alábbi esetek következhetnek be.

1a	Pálya betöltése	GIVEN:	Az alkalmazás telepítve van.
		WHEN:	Valamelyik nyomógombra kattintás.
		THEN:	Betölti a megfelelő pályát.

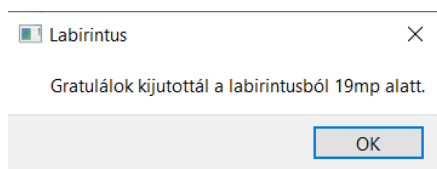
<b>1b</b>	Pálya betöltése	GIVEN:	Be van töltve az egyik pálya.
		WHEN:	Vissza gombra kattintás.
		THEN:	Visszatölti a menüt.
<b>2a</b>	Mozgás	GIVEN:	A játék aktív.
		WHEN:	Megnyomva az egyik gombot az irányításhoz, amivel abba az irányba tud mozogni, de nem lesz vége a játéknak.
		THEN:	Abba az irányba mozog a karakter, vele együtt a látható mezők is változnak.
<b>2b</b>	Mozgás	GIVEN:	A játék aktív.
		WHEN:	Megnyomva az egyik gombot az irányításhoz, amivel abba az irányba tud mozogni, és ezzel vége a játéknak.
		THEN:	Abba az irányba mozog a karakter, majd egy jelzést kap a játékos hogy mennyi idő alatt teljesítette a pályát.
<b>2c</b>	Mozgás	GIVEN:	A játék aktív.
		WHEN:	Megnyomva az egyik gombot az irányításhoz, amivel abba az irányba nem tud mozogni.
		THEN:	Nem változik semmi.
<b>3a</b>	Megállítás	GIVEN:	A játék aktív.
		WHEN:	Stop gombra kattintás.
		THEN:	Megáll a játék.
<b>3b</b>	Megállítás	GIVEN:	A játék áll.
		WHEN:	Start gombra kattintás.
		THEN:	A játék elindul.

## Felhasználói felület terve

A játék elsőnek megjelenő felületén 3 nyomógomb található, amik lerajzolják a játék 3 különböző pályáját. A pályára való lépéskor még két nyomógomb jelenik meg, egy a visszalépésre, egy pedig a játék indítására/megállítására. Ezen kívül üzenet-ablakok is megjelennek.



A játék végét üzenet-ablak jelzik:

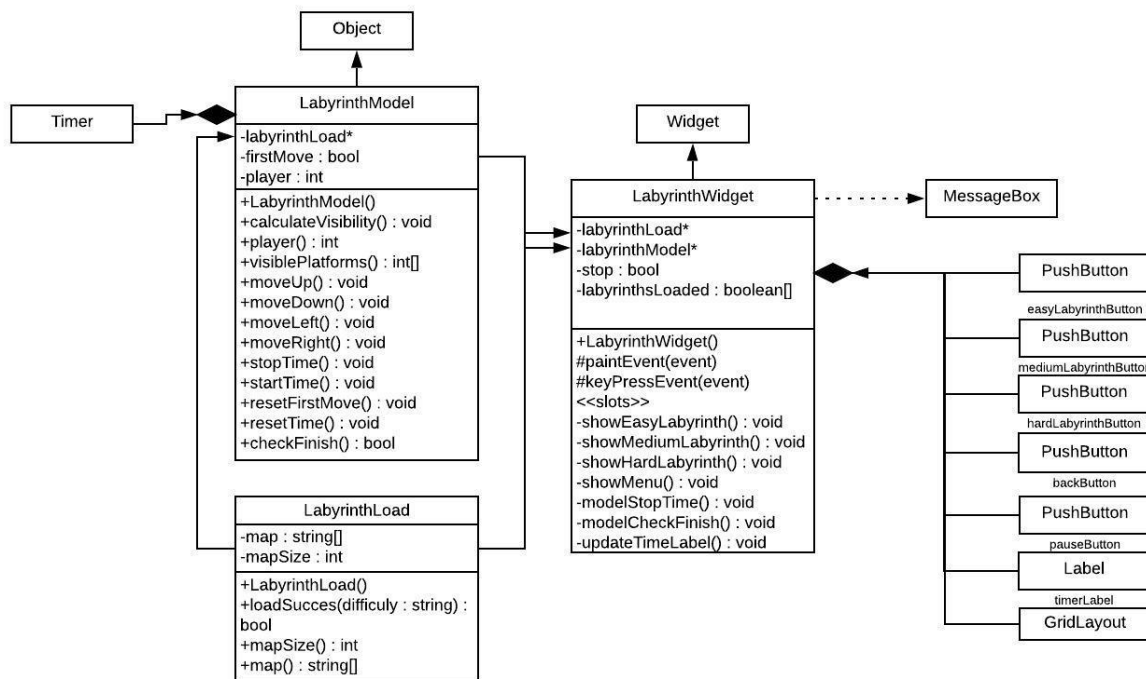


## Architektúra

Az alkalmazást két rétegű (nézet-modell) architektúrában valósítjuk meg.

## Osztálydiagram

Az alkalmazást három osztállyal írjuk le. LabyrinthModel a játék modelljeként szolgál, LabyrinthLoad a pályákat tölti be és tárolja az adott fájlból, LabyrinthWidget a játék felületét kezeli.



## Vezérlők és adattagok

### LabyrinthWidget

Az osztály a QWidget-ből származik majd, a nyomógombok és a címke mellett elrendezőket is tartalmaz(gridLayout). Itt adattagokat nem tárolunk.

### LabyrinthLoad

Itt tároljuk a betöltött pálya adatait: méretét (mapSize) és a mezőket (map).

### LabyrinthModel

A játékos helyét (player) és az abból következő mezőket, amiket lát a játékos (visiblePlatforms) tároljuk itt. Valamint egy igaz/hamis változót (firstMove) a számítások meggyegyszerűsítése érdekében.

## Metódusok

### LabyrinthWidget

4 metódus található benne, ami csak a pályák és a menü közötti váltásra szolgál (show... metódusok). Van még két metódus, ami csak a modellt használja, amik megállítják/elindítják a játékot (modelStopTime) és ellenőrzik hogy vége van a játéknak (modelCheckFinish), és még egy ami csak az eltelt idő megjelenítésére szolgál (updateTimeLabel).

### LabyrinthLoad

Egy fontos metódusa van csak ami a pálya betöltésére szolgál.

### LabyrinthModel

Legfontosabb metódusa szolgál a látható mezők kiszámolására (calculateVisibility), ezen kívül van még 4 metódusa, ami a játékos mozgására szolgál (move... metódusok), 2 ami a játék megállítására/elindítására, 2 ami a segédváltozók alaphelyzetbe állítására szolgál (reset... metódusok) és egy ami csak ellenőrzi a játék végét.

### Eseménykezelés

sender	signal	receiver	slot
easyLabyrinthButton	Clicked()	LabyrinthWidget	showEasyLabyrinth()
mediumLabyrinthButton	Clicked()	LabyrinthWidget	showMediumLabyrinth()
hardLabyrinthButton	Clicked()	LabyrinthWidget	showHardLabyrinth()
pauseButton	Clicked()	LabyrinthWidget	modelStopTime()
backButton	Clicked()	LabyrinthWidget	showMenu()
timer	timeout()	LabyrinthWidget	updateTimeLabel()

### Végfelhasználói tesztesetek:

	Teszteset	Elvárt hatás
<b>1a</b>	Pálya betöltése	Megjelenik a pálya.
<b>1b</b>	Menü betöltése	Megjelenik a menü.

<b>2a</b>	Mozgás, jó irányban	Mozog a karakter.
<b>2b</b>	Mozgás, jó irányban és vége a játéknak	Mozog a karakter és vége a játéknak.
<b>2c</b>	Mozgás, rossz irányban	Nem mozog a karakter.
<b>3a</b>	Megállítás	Megáll a játék.

<b>3b</b>	Elindítás	Elindul a játék.
-----------	-----------	------------------