Programozási Technológia 3. Beadandó (1. feladat: Maci Laci), dokumentáció

- Készítette: Szabó Péter Bence (QDMPVQ)

A kiosztott feladat teljes szövege

A meséből jól ismert Maci Laci bőrébe bújva a Yellowstone Nemzeti Park megmászhatatlan hegyei és fái között szeretnénk begyűjteni az összes rendelkezésre álló piknik kosarat. Az átjárhatatlan akadályok mellett Yogi élelem szerzését vadőrök nehezítik, akik vízszintesen vagy függőlegesen járőröznek a parkban. Amennyiben Yogi egy egység távolságon belül a vadőr látószögébe kerül, úgy elveszít egy élet pontot. (Az egység meghatározása rád van bízva, de legalább a Yogi sprite-od szélessége legyen.) Ha a 3 élet pontja még nem fogyott el, úgy a park bejáratához kerül, ahonnan indult.

A kalandozás során, számon tartjuk, hogy hány piknik kosarat sikerült összegyűjtenie Lacinak. Amennyiben egy pályán sikerül összegyűjteni az összes kosarat, úgy töltsünk be, vagy generáljunk egy új játékteret. Abban az esetben, ha elveszítjük a 3 élet pontunkat, úgy jelenjen meg egy felugró ablak, melyben a nevüket megadva el tudják menteni az aktuális eredményüket az adatbázisba. Legyen egy menüpont, ahol a 10 legjobb eredménnyel rendelkező játékost lehet megtekinteni, az elért pontszámukkal, továbbá lehessen bármikor új játékot indítani egy másik menüből.

A program működésének rövid leírása

Alapok

A programot elindítva azonnal új játék indul. A `Játék` menüben található három gomb, illetve jelentésük:

- `Új játék`: Új játék kezdése.
- `Dicsőségtábla`: A legjobb 10 játékos nevét, illetve pontszámát jeleníti meg táblázatban.
- `Kilépés`: Kilép a játékból.

A játék célja

Összegyűjteni minél több kosarat anélkül, hogy elveszítenénk mind a 3 életpontunkat, ami akkor történik, ha elkap egy járőr.

Irányítás

A játékos a pályán felfelé, balra, lefelé és jobbra mozgatható a billentyűzet `WASD` gombjaival.

Játékmenet

A játéktér 10 ms-onként frissül, a játéktábla felett található a játékkal kapcsolatos összes információ: az életek száma, az eddigi összpontszám, a jelenlegi pályán összegyűjtött kosarak száma, illetve a játék elejétől eltelt idő. Új játék indulásakor a `levels.txt` forrásfájlban található, előre definiált 10 pálya közül egy véletlenszerű pálya töltődik be, és induláskor 3 életpontja van a játékosnak. A játékos (Maci karakterrel reprezentálva) minden pályán adott pozícióból indulva a `WASD` gombokkal tud mozogni. A pályán átjárhatatlan akadályokként hegyek és fák szerepelnek, ezen kívül az összegyűjtendő kosarak, és a vízszintesen vagy függőlegesen mozgó járőrök, akik másodpercenként egy egységnyit lépnek az adott irányba. Ha túl közel megyünk egy őrhöz, akkor visszakerül a maci az induló pozícióba, és elvesztünk egy életpontot. Ha egy pályán az összes kosarat összegyűjtöttük, azonnal új pálya töltődik be. A játék akkor ér véget, ha mind a 3 életpontunkat elvesztjük. Ekkor egy felugró ablakban megadhatjuk a nevünket, ha szeretnénk elmenteni az eredményünket, és ha az eddig beküldött pontszámok közül legalább egynél jobb (vagy egyenlő), akkor a `Dicsőségtábla` menüpontban láthatóvá válik a táblázatban.

Fontosabb technikai részletek

A játékszintek betöltése

A játék szintjei egy szövegfájlból (*res/levels.txt*) töltődnek be. A szövegfájl 10 pályát tartalmaz, mindegyikük 10x10-es dimenziójú (ámbár nem kötelező ez a méret, más dimenziójú pályákkal is működne a játék). Hétféle karakter szerepelhet egy pálya reprezentációjában, ezeknek jelentései az alábbiak:

- `E`: Üres mező (Empty)
- `M`: Hegy (Mountain)
- `T`: Fa (Tree)
- `B`: Kosár (Basket)
- `H`: Vízszintesen mozgó járőr (Guard)
- `V`: Függőlegesen mozgó járőr (Guard)
- `Y`: A játékos kezdőpozíciója

Ezek alapján kapják egy-egy pályának a mezői a tulajdonságaikat, megfelelő sprite-jaikat.

A járőrök mozgásának és közelség-ellenőrzésének implementációja

A járőrök mozgatásáért a `MainWindow`-ban található `guardMoveTimer` felelős, amely másodpercenként lefuttatja a `game` (a játék példánya) `stepGuards` metódusát, illetve újrarajzolja a grafikát (`board.repaint()`). A `game.stepGuards()` metódus "továbbadja" a meghívást a `gameLevel` adattagnak, melynek a `moveGuards` metódusát hívja meg. Itt két külön metódushívás történik, az egyik a vízszintesen (`moveGuardsH`), a másik a függőlegesen (`moveGuardsV`) mozgó járőrök mozgatására létrehozott metódusra. Mindkettő ugyanúgy

működik, csak az irányok különbözőek: végigmegyünk az összes, listában eltárolt járőrön (`guardsH/guardsV`), megnézzük, hogy a jelenlegi pozíciójukhoz képest a következő, adott irányba történő lépésnél szabad mezőre lépnek-e, ha igen, akkor odaléptetjük, ha viszont akadály van előttük (hegy/fa/kosár), akkor megfordulnak az ellentétes irányba, és arra mennek, amíg nem ütköznek újabb akadályba. A járőrök `Guard` példányok, melyeknek van egy `Position` és egy `Direction` adattagja. A játékos közelségét az őrökhöz szintén egy időzítő, a `guardCheckTimer` ellenőrzi, ez 10 ms-onként megnézi a `game.checkYogiNearGuards()` visszatérési értékét. Ha igaz, akkor túl közel ment a játékos az őrhöz, elveszít egy életpontot. Ezután azt is ellenőrzi, hogy van-e még életpontja a játékosnak a `game.isGameEnded()` függvényen keresztül, és ha igazat ad vissza, vége a játéknak, leállítjuk az időzítőket, és megjelenítjük a név megadáshoz szükséges ablakot. A `game.checkYogiNearGuards()`-on belül a `gameLevel.check()` metódus visszatérési értékét figyeljük, ha igaz, akkor levonunk egy életpontot. A `gameLevel.check()` végzi el a nehezét a feladatnak, ugyanis itt végigiterálunk az összes járőrön, és egyesével végignézzük, hogy nincsen-e a járőrhöz képest eggyel felette, alatta, tőle balra, vagy tőle jobbra a játékos. Ha igen, akkor visszahelyezzük a játékost a kezdőpozícióra, és igazat adunk vissza, különben hamisat.

Tesztesetek

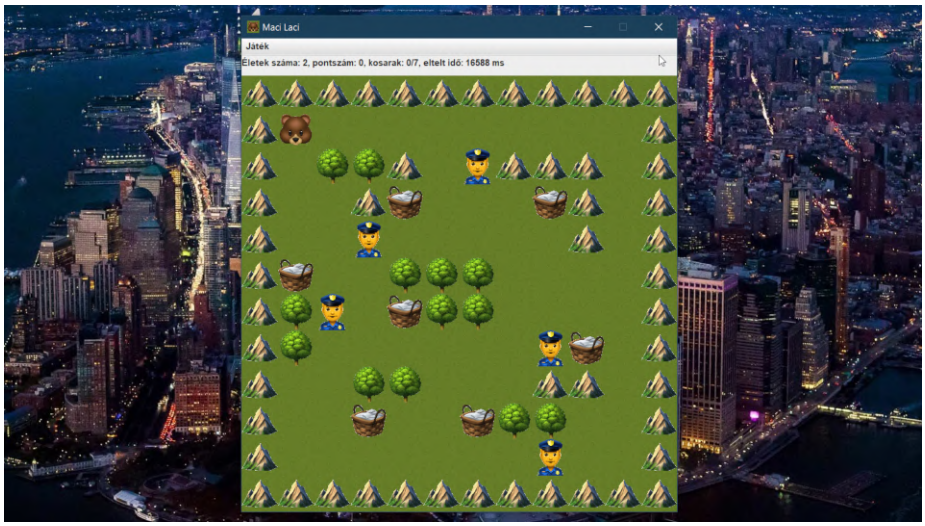
- Új játék indítása

- Lépés WASD gombokkal

- Kosár begyűjtése

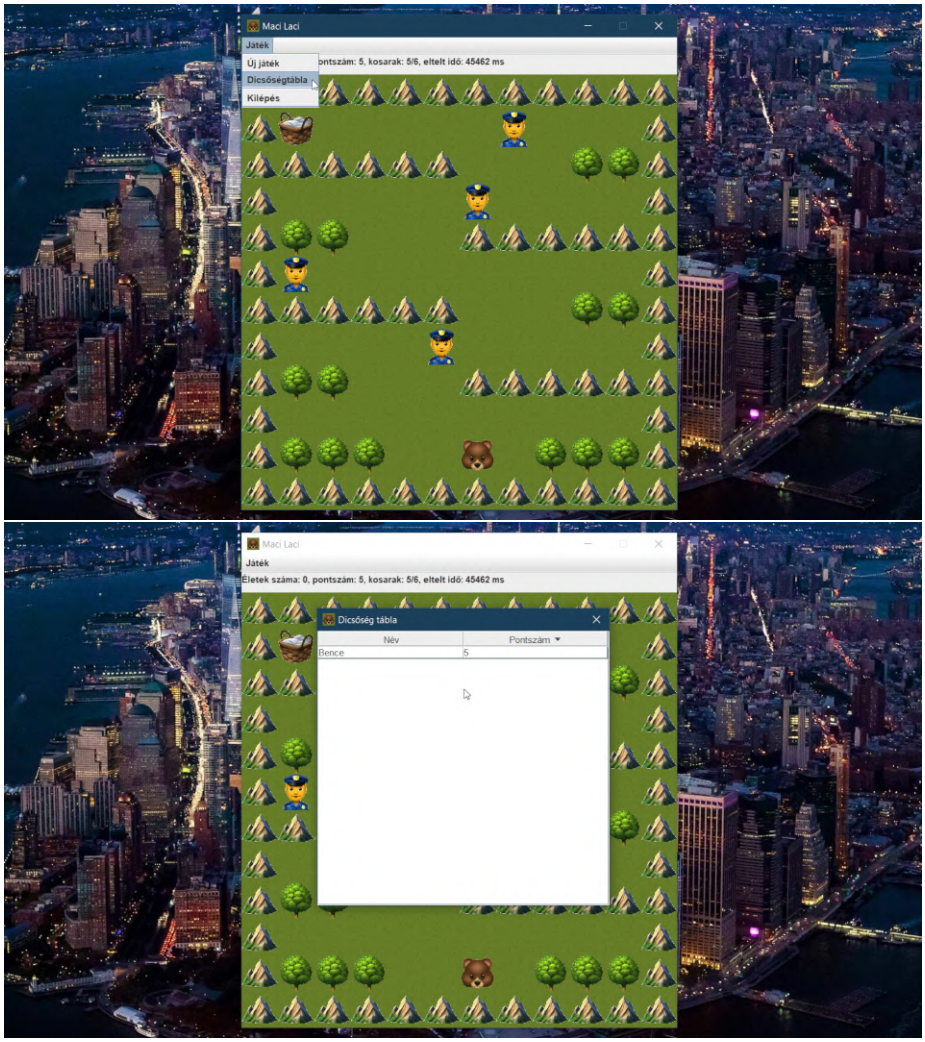
- Összes kosár begyűjtése után új pálya betöltése

- Őr közelébe lépés


Maci Laci
Játék
Életek száma: 2, pontszám: 0, kosarak: 0/7, eltelt idő: 16588 ms

- Játék vége, eredmény mentése

- Dicsőségtábla



A program osztálydiagramja

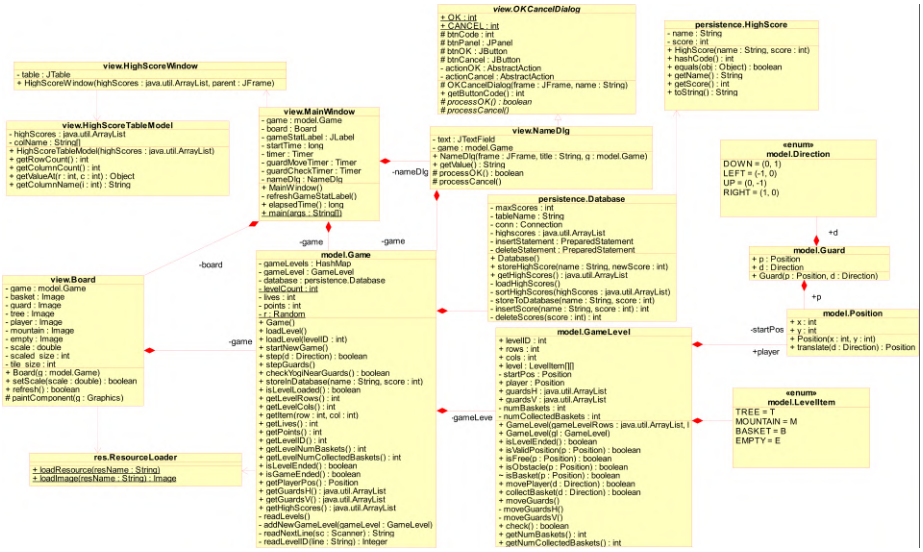


Figure 1: A program osztálydiagramja