

Programozási technológia

3. beadandó feladat

Seregi Péter
G9SIY0

Feladat

Labirintus

Készítsünk programot, amellyel egy labirintusból való kijutást játszhatunk. A játékos a labirintus bal alsó sarkában kezd, és a feladata, hogy minél előbb eljusson a jobb felső sarokba úgy, hogy négy irányba (balra, jobbra, fel, vagy le) mozoghat, és elkerüli a labirintus sárkányát. Minden labirintusban van több kijutási útvonal. A sárkány egy véletlenszerű kezdőpozícióból indulva folyamatosan bolyong a pályán úgy, hogy elindul valamilyen irányba, és ha falnak ütközik, akkor elfordul egy véletlenszerűen kiválasztott másik irányba. Ha a sárkány a játékos szomszédos területre jut, akkor a játékos meghal. Mivel azonban a labirintusban sötét van, a játékos mindig csak 3 sugarú körben látja a labirintus felépítését, távolabb nem. Tartsuk számon, hogy a játékos mennyi labirintuson keresztül jutott túl és amennyiben elveszti az életét, mentjük el az adatbázisba az eredményét. Egy menüpontban legyen lehetőségünk a 10 legjobb eredménnyel rendelkező játékost megtekinteni, az elért pontszámukkal, továbbá lehessen bármikor új játékot indítani egy másik menüből. Ügyeljünk arra, hogy a játékos, vagy a sárkány ne falon kezdjenek.

Terv

Típusok

Entity

A játék egy elemét reprezentáló absztrakt osztály. Eltárolja az elem pozícióját, méretét, háttérképét. A „draw” metódus ezek alapján kirajzolja az elemet a képernyőre. Belőle származik le a Player, a Dragon és a Wall osztály is. A Player és Dragon osztályok az előbb említett adatagok mellett eltárolják a sebességüket is, amit az általuk megvalósított „move” metódusban használnak fel.

Direction

Egy enum, amely a négy irányt (UP, DOWN, RIGHT, LEFT) tárolja.

Board

A játék logikáját megvalósító osztály. A „setupMap” metódussal szöveges fájlokból beolvassa a játéktér elemeit, amelyeket el is tárol. Legfőbb metódusai a „movePlayer” és a „moveDragon”, amelyek a játékos és a sárkány mozgását valósítják meg. Emellett az „isOver” metódus vizsgálja, hogy a játékos elvesztette-e a játékot, az „isWon” metódus pedig azt, hogy megnyerte-e. Az osztály „draw” metódusa meghívja az összes komponens „draw” metódusát, ezáltal kirajzolja őket a képernyőre.

GamePanel

A játék megjelenítéséért és a felhasználói interakciók detektálásáért felelős osztály. Eltárol egy Board objektumot, amelyet a billentyűlenyomások (fel, le, jobb, bal) függvényében manipulál. Tizedmásodpercenként mozgatja a játékost és a sárkányt. Ha a játékos meghal, akkor a játékos nevét elkérve eltárolja a játékos eredményét az adatbázisban. Ha a játékos az adott szintet teljesíti, akkor elindítja a következő pályát (az utolsó, azaz 10. pálya után az első indítja el).

LabyrinthGUI

Egy frame segítségével megjeleníti a GamePanel-t, valamint létrehozza a menüsört. A menüben lehetőség van könnyített játékmód („light”) be- és kikapcsolására. Ezenfelül itt lehet a toplistát is megjeleníteni, illetve lehetőség van itt is kilépni az alkalmazásból.

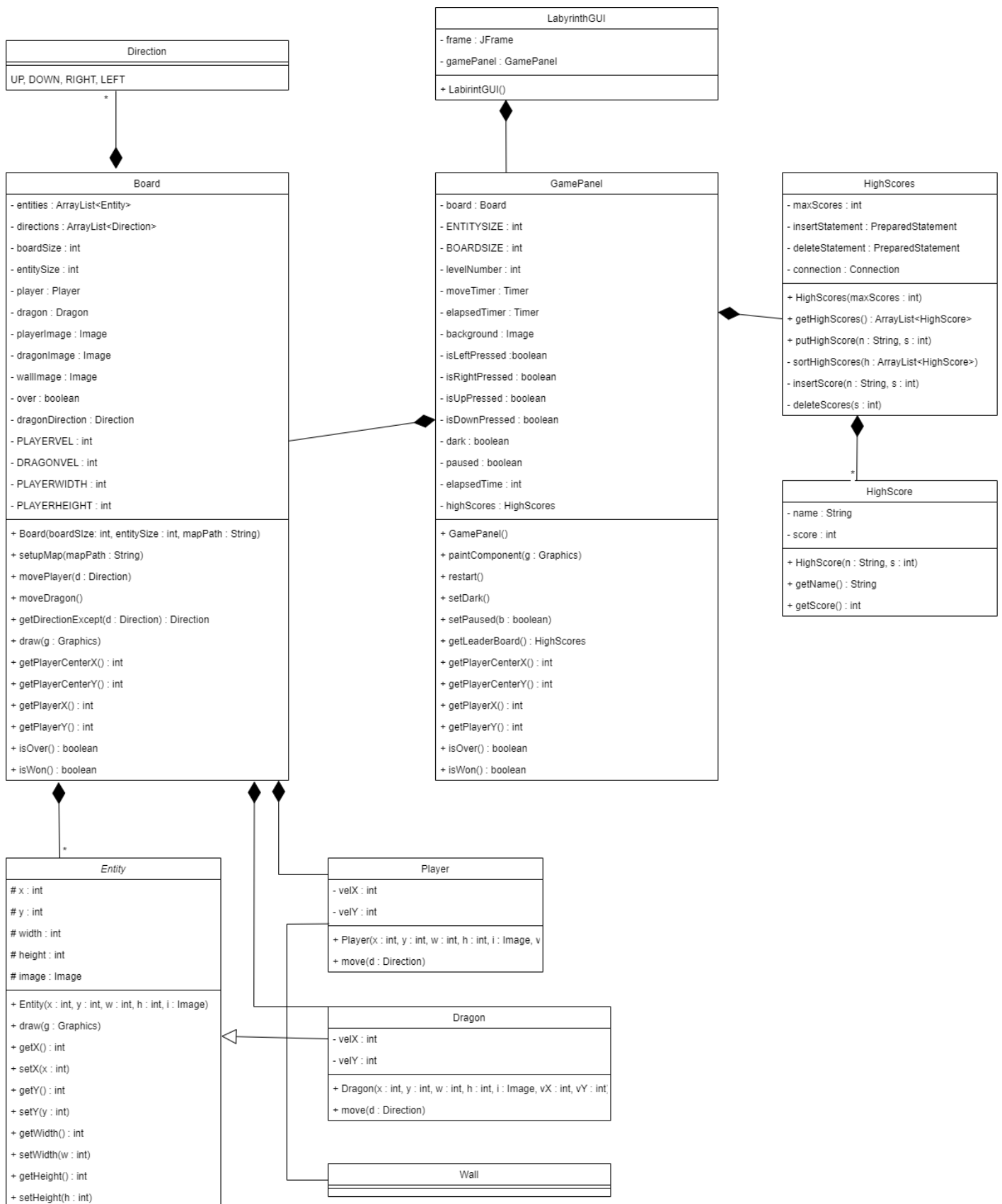
HighScore

Az adatbázis egy rekordját reprezentáló osztály, amely eltárolja a játékos nevét illetve eredményét.

HighScores

A rekordokat az adatbázisba helyező, valamint onnan lekérő osztály.

Osztálydiagram



Implementációs fejezet

A sárkány mozgatása:

A Dragon önmagában is rendelkezik egy „move(Direction d)” metódussal, azonban mozgatását annak logikájával a Board osztály „moveDragon()” metódusa végzi.

A mozgítás lépései:

1. A sárkány mozgatása az aktuális irányba a sárkány „move” metódusával. Az aktuális irányt a „dragonDirection” adattag tárolja.
2. Határátlépés vizsgálata. Amennyiben a mozgással a sárkány kimegy a pályáról, visszaállítjuk a pozícióját úgy, hogy ne lógjon ki.
3. Ütközések vizsgálata. Végigmegyünk az elemek listáján, és ütközés esetén kiválasztjuk a másik elemet. Ha Player-rel ütközött, akkor a Board osztály állapotát befejezettre állítjuk.
4. Az ütközés után ellenkező iránnyal hívjuk meg a „move” metódust, hogy ne lógjon rá semmire sem a sárkány.
5. Mivel a sárkány ütközött, ezért át kell állítanunk az irányát. Ezt a „getDirectionExcept(Direction d)” metódus segítségével végezzük, amely egy a paramétertől eltérő véletlenszerű irányt ad vissza. Így garantáljuk, hogy a következő mozgítás során a sárkány már egy másik irányba mozogjon.

Események, eseménykezelők (esemény: eseménykezelés)

GamePanel osztály:

1. felfelé nyíl lenyomása: „isUpPressed” igazra állítása
2. felfelé nyíl elengedése: „isUpPressed” hamisra állítása
3. lefelé nyíl lenyomása: „isDownPressed” igazra állítása
4. lefelé nyíl elengedése: „isDownPressed” hamisra állítása
5. balra nyíl lenyomása: „isLeftPressed” igazra állítása
6. balra nyíl elengedése: „isLeftPressed” hamisra állítása
7. jobbra nyíl lenyomása: „isRightPressed” igazra állítása
8. jobbra nyíl elengedése: „isRightPressed” hamisra állítása
9. 10 milliszekundum eltelése: MoveListener osztály „actionPerformed” metódusának meghívása, amely mozgatja a játékost és a sárkányt, valamint vizsgálja a játék esetleges befejeződését.
10. 1000 milliszekundum eltelése: az eltelt másodpercek növelése eggyel

LabyrinthGUI osztály:

1. „New game” menüpont: új játék indítása
2. „Light” menüpont: fény fel-vagy lekapcsolása

3. „Leaderboard” menüpont: toplista megjelenítése
4. „Exit” menüpont: Alkalmazásból való kilépés

Tesztelés

Fehérdobozos tesztek

Teszt leírása	Tevékenység	Elvárt kimenet
játékos y koordinátája > 0	Board.isWon()	false
játékos y koordinátája = 0	Board.isWon()	true
játékos és sárkány nem „ütközik”	Board.isOver()	false
játékos és sárkány „ütközik”	Board.isOver()	true

Feketedobozos tesztek

Teszt leírása	Elvárt kimenet
program elindítása	létrejön az ablak az alapértelmezett méretben, elindul a játék és az időszámláló
balra nyíl lenyomása	ha balra nincs akadály, a játékos balra mozog
felfelé nyíl lenyomása	ha felfelé nincs akadály, a játékos felfelé mozog
a játékos bejut a célba	felugró ablak jelzi, hogy a játékos az adott számú pályát teljesítette
a sárkány és a játékos „összeütközik”	Felugró ablak jelzi a játék végét, és elkéri a játékos nevét.
új játék indítás a menüből	új játék indul az első pályával
toplista megnyitása a menüből üres toplistával	felugró ablak „Leaderboard is empty” szöveggel
toplista megnyitása a menüből nemüres toplistával	felugró ablak csökkenő sorrendben felsorolja a (max 10 db) legjobb játékos nevét és pontszámát