

Pozsár Bence Zsolt  
F2K9QJ  
f2k9qj@inf.elte.hu

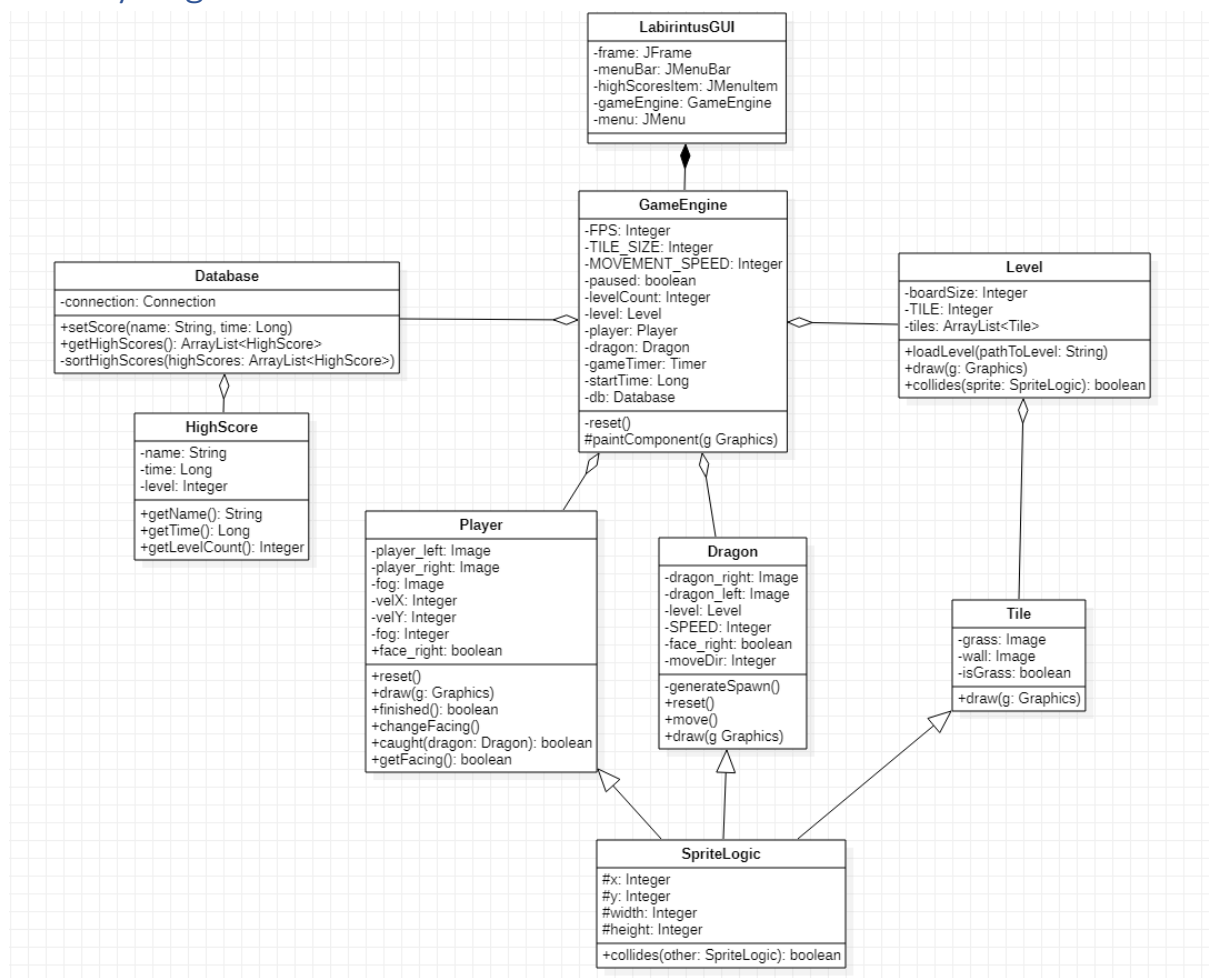
3. beadandó/3.feladat

2021. december 4.

## Feladat

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 játékosal 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.

## Osztálydiagram



## A feladat elemzése

A játék során egy karaktert irányítunk, aki a sötétben 3 egységben látja a játékeret. A játékos célja a bal alsó sarokból eljutni a jobb felső sarokba. A játéktér egy labirintus, ahol nehéz közlekedni a sötétben. Ezen felül a kijutást nehezíti egy sárkány, aki a játékosra valahol mozog. A játék véget ér, ha a sárkány elkap vagy a játékos az összes labirintusból kijutott.

## Implementáció

### Dragon::generateSpawn()

Generál egy véletlenszerű x és y koordinátát 0-tól 10-ig, ami megadja, hogy melyik kockában kezdjen a sárkány. Amíg a sárkány ütközik egy fallal, addig a sárkánynak új kezdőpont lesz generálva.

### Dragon::move()

A sárkányt egy megadott irányba mozgatja. Ha a sárkány falnak ütközik, akkor egy véletlenszerű irányt változtat. Ha eddig balra nézett és jobbra megy, akkor megfordul és fordítva.

### SpriteLogic::collides(other: SpriteLogic): boolean

Megadja, hogy két test ütközik-e egymással. A függvény ütközést jelez, ha a vizsgált test kint van a pályáról.

## Események

Osztály	Eseménykezelő	Esemény	Tevékenység
LabirintusGUI	ActionListener	kattintás	Újrakezdi a játékot
LabirintusGUI	ActionListener	kattintás	Előhozza a toplistát.
GameEngine	NewGameListener	Timer	Egy bizonyos időközönként megvizsgálja, hogy elkapott-e a sárkány vagy be lett-e fejezve a játék, ha igen a játék feljegyzi az adatbázisba az eredményt.
GameEngine	getKeystroke(„DOWN”)	le billentyű	A karaktert lefelé mozgatja
GameEngine	getKeystroke(„UP”)	fel billentyű	A karaktert felfelé mozgatja
GameEngine	getKeystroke(„LEFT”)	balra billentyű	A karaktert balra mozgatja
GameEngine	getKeystroke(„RIGHT”)	jobbra billentyű	A karaktert jobbra mozgatja

## Tesztelés

Teszteteset rövid leírása	Elvárt viselkedés
A nyilak megnyomása	Megmozdítja a karaktert a megnyomott nyíl irányába.
Az ESC billentyű megnyomása	Befagyasztja a játékot az újbóli megnyomásig
Új játék kezdése a menüből	A játék az első pályától újraindul
Toplista megnyitása	Az adatbázisból lekéri az adatokat és kilistázza egy táblázatban.
Falnak vagy játéktér végének ütközés	A játékost visszamozdítja annyival, amennyivel kiment, így úgy tűnik, mintha nem tudna nekimenni a falaknak vagy a játéktér végének.
Jobb felső sarokba érkezés	Áttesz a következő pályára, vagy a 10. után adatbázisba teszi a rekordot a megadott név alapján, majd újratekdi a játékot.
Ütközés a sárkánnyal	Adatbázisba teszi a rekordot a megadott név alapján, majd újratekdi a játékot.