

# Csák Attila Gergő

s7b01e

## 3. Labirintus (Labyrinth)

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 játékoskal 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, mentsük 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.

## Megoldási Terv

A játék fő mechanikája a játékos és a sárkány érvényes mozgása a labirintusban. Az volt a terv, hogy egy adott labirintuscella mindig tudja, mely utak járhatók egy entitás számára, és hogy az adott út melyik cellába vezet.

Ezek miatt úgy döntöttem, hogy Tile objektumokat hozok létre, amelyek minden kardinális irányba tárolnak egy referenciát egy másik Tile-ra (vagy null értéket, ha fal van), miközben a másik Tile is tárolja a saját szomszédjait.

A labirintus generálásához azonban valami másra volt szükség, ezért két külön modellt gondoltam ki: az egyiket az entitások áthaladására optimalizáltam, a másikat pedig a generáló algoritmus egyszerűbb implementálásához.

A játékban változásokat előidéző események a felhasználói bemenetek és a sárkány mozgása voltak, ami azt jelentette, hogy a fő eseménykezelőknek a billentyűleütésekhez és a sárkány időszakos mozgását kezelő Timer eseményhez kellett kapcsolódniuk.

Az adatbázishoz az SQLite-t tűnik a legpraktikusabbnak ebben az esetben, mivel hogy nagyon kicsi mennyiségű adattal dolgozunk, és nagyon kevés írásra és lekérdezésre lesz szükségünk. Nem kell külön adatbázist futtatni szerveren, csak kódból le lehet generálni meg olvasni egy fájlt.

## Megvalósítás

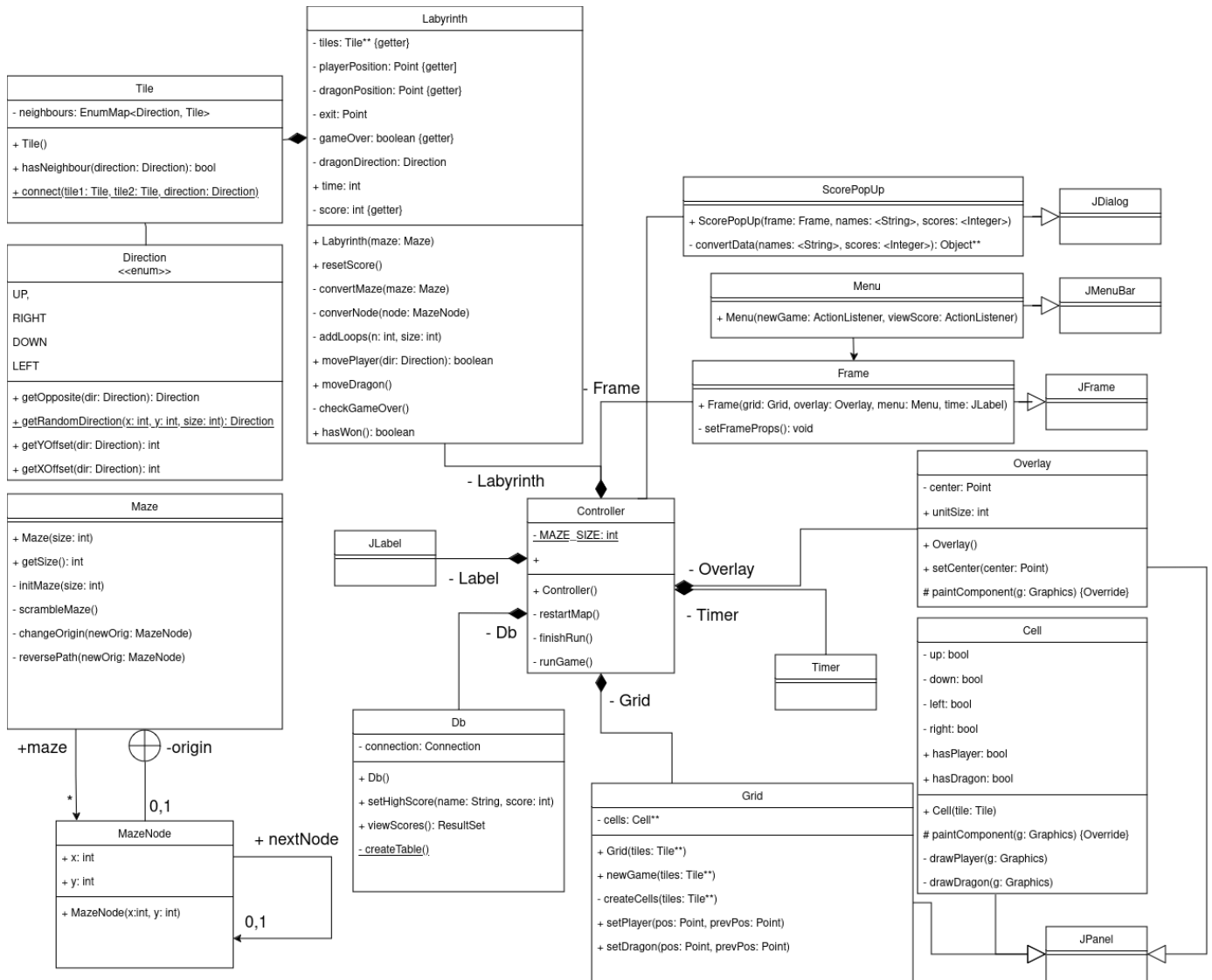
A labirintus generálása némi nehézséget okozott. Találtam egy egyszerű algoritmust online, amit Origin Shift néven emlegettek. Ez egy kezdetben tökéletes labirintust igényel, amit manuálisan kell elkészíteni, majd meg kell “zavarni”, hogy véletlenszerű legyen. Ezt kicsit módosítottam, hogy az összezavarás ne csak véletlenszerű bejárással történjen, hanem az összes MazeNode kb ugyanannyiszor legyen meglátogatva. Emiatt ciklikusan megyek át minden MazeNode-on, hogy mindenhol legyen valamennyi randomizálás. A MazeNode objektumok megkönnyítik ennek az algoritmusnak a végrehajtását. Az algoritmus lefutása után egy tökéletes labirintust kapunk, de a feladat azt kéri, hogy több érvényes kijárat is legyen a Labirintusból. Így, miután átalakítottam a Maze és MazeNode objektumokat Labyrinth és Tile objektumokká, írtam egy függvényt, amely hurkokat ad hozzá a Labirintushoz (addLoop). Alapvetően ez annyit tesz, hogy egy élt hoz létre két véletlenszerűen kiválasztott szomszédos Tile között.

A sárkány és a játékos mozgása hasonlóan működnek egymáshoz. A logikai oldal tárolja ezeknek az entitásoknak a tényleges pozícióját, míg a Cell objektumok, amelyek felelősek a Labirintus grafikai megjelenítéséért, ténylegesen megrajzolják az entításokat. Csak azokat a Cellákat frissítem, amelyeket egy entitás elhagy vagy belép, a többit érintetlenül hagyom, hogy elkerüljem a felesleges újrarajzolásokat. Ahogy a megoldási tervben is tárgyaltuk, a játékos mozgását a felhasználó billentyűlenyomásai váltják ki, ezért egy keyListener-t implementáltam a fő játékkeretre, ami az adott billentyűtől függően logikai számítást indít el a megkísérelt lépés helyességének a kiszámításához. Ha a lépés valid, akkor egy állapotszinkronizálás jön létre a logika és a megjelenítés között, majd egy újrarajzolás.

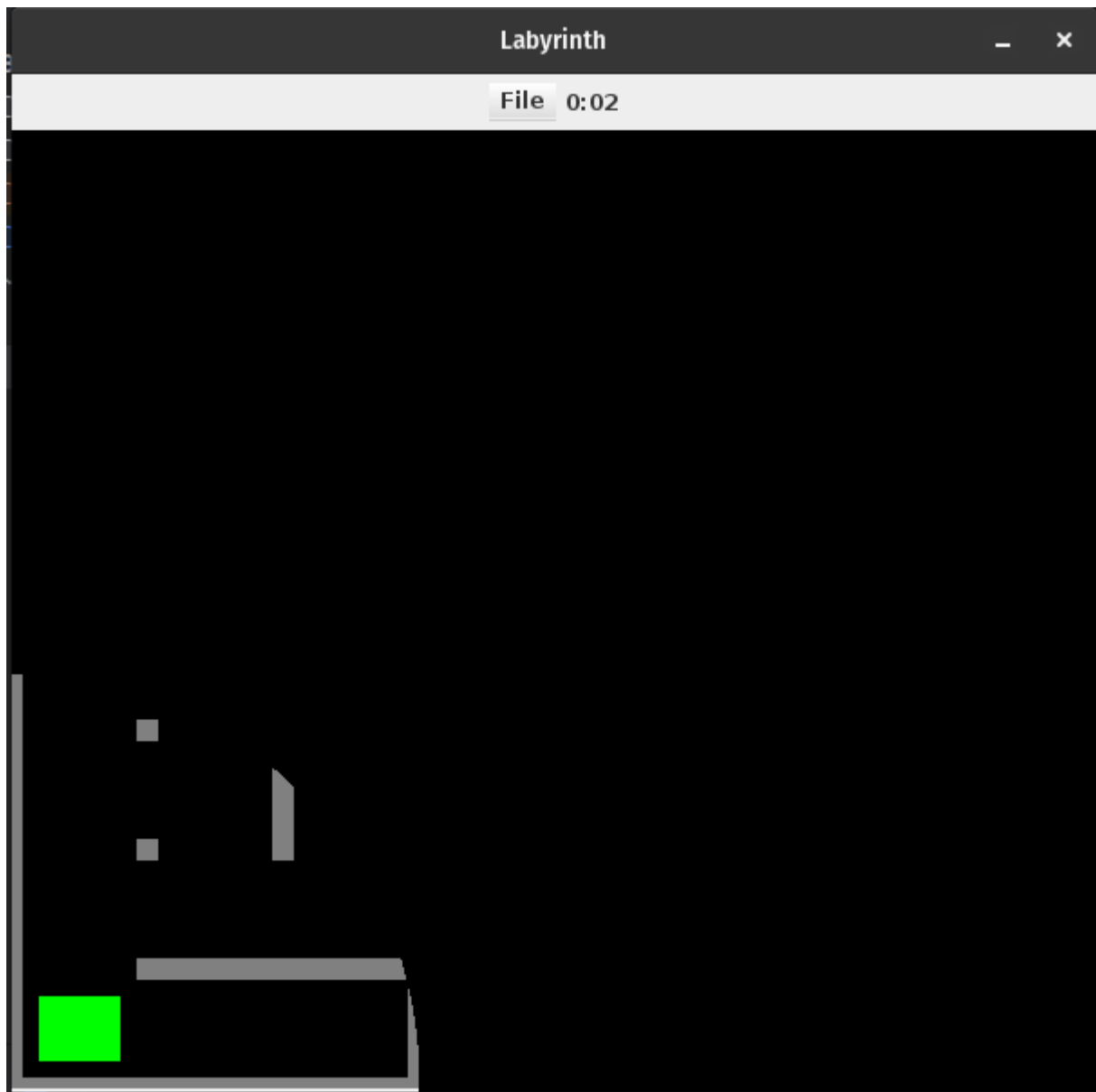
A sárkány esetében egy időzítőt hoztam létre, amely időzített eseményeket indít el, és ezek is egy logikai oldali számítást váltanak ki. Mindkét esetben a logikában érzékelt változások az érintett Cellák újrarajzolását eredményezik.

A ponttáblázat megjelenítéséhez az eseményt az aktivált menüelem váltja ki. Ez egy adatbázis-lekérdezést végez, majd létrehoz és megjelenít egy párbeszédablakot a megjelenített pontszámokkal.

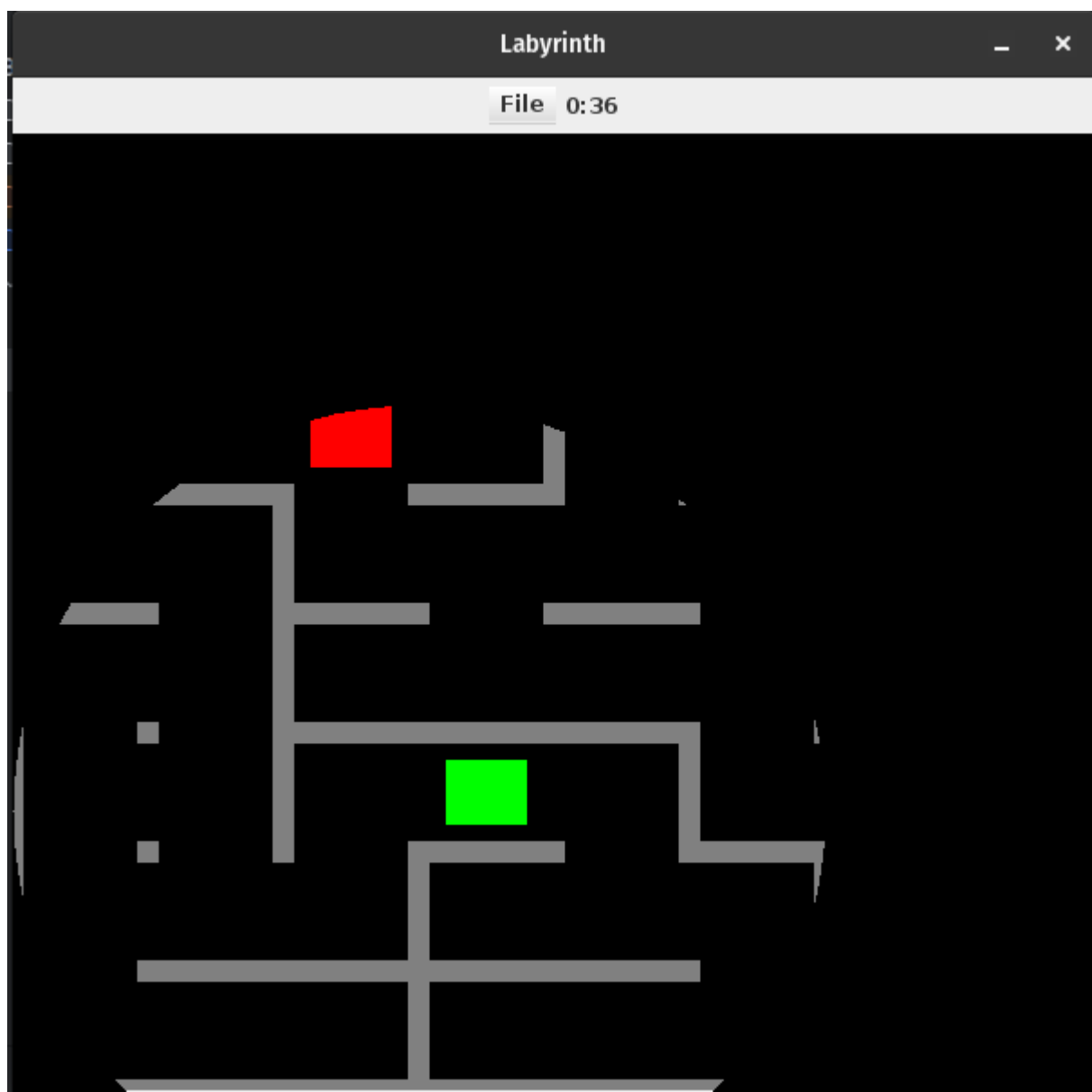
## UML Osztálydiagram:



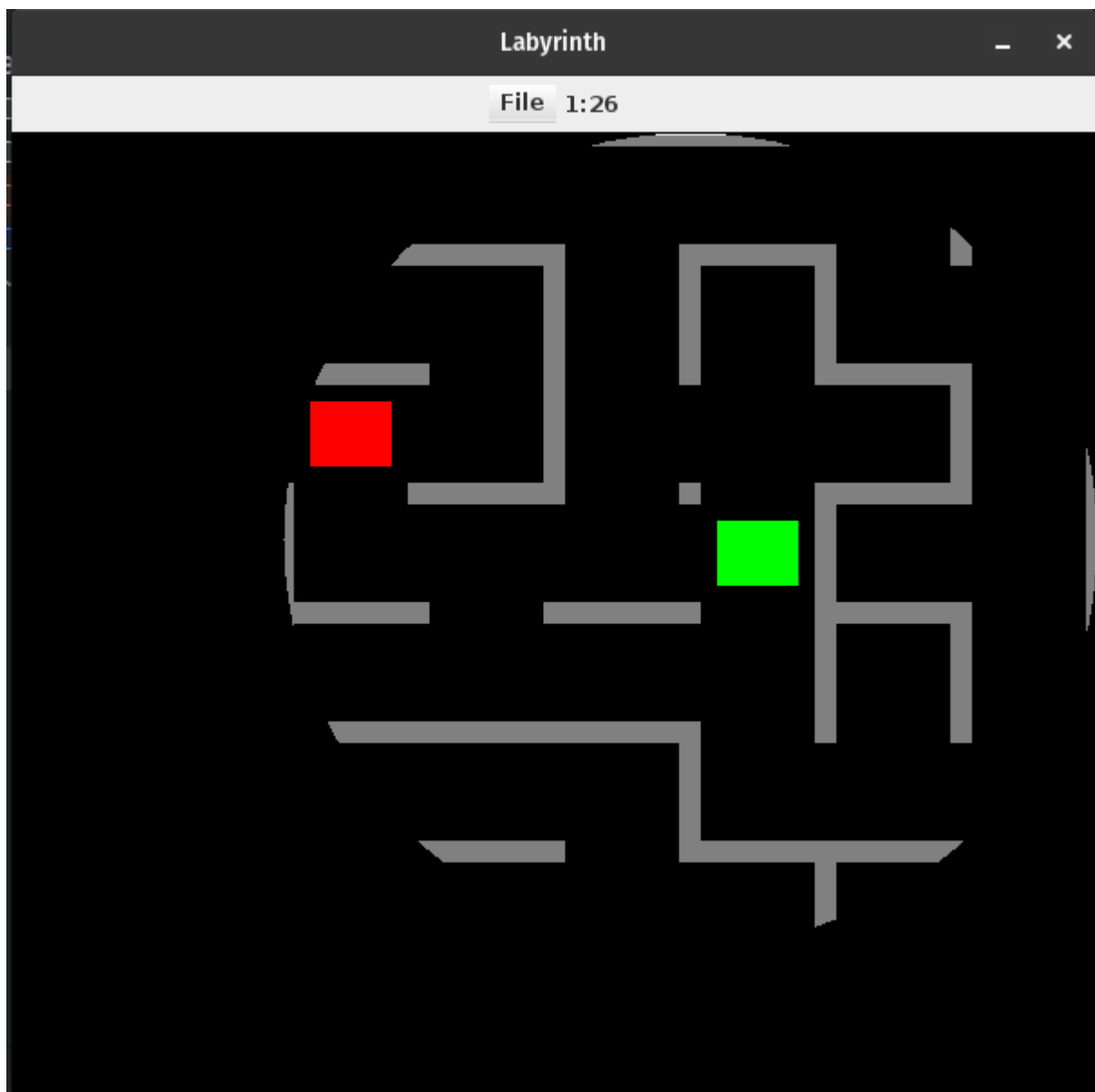
Képek a futtatásról:  
Kiindulo állapot:



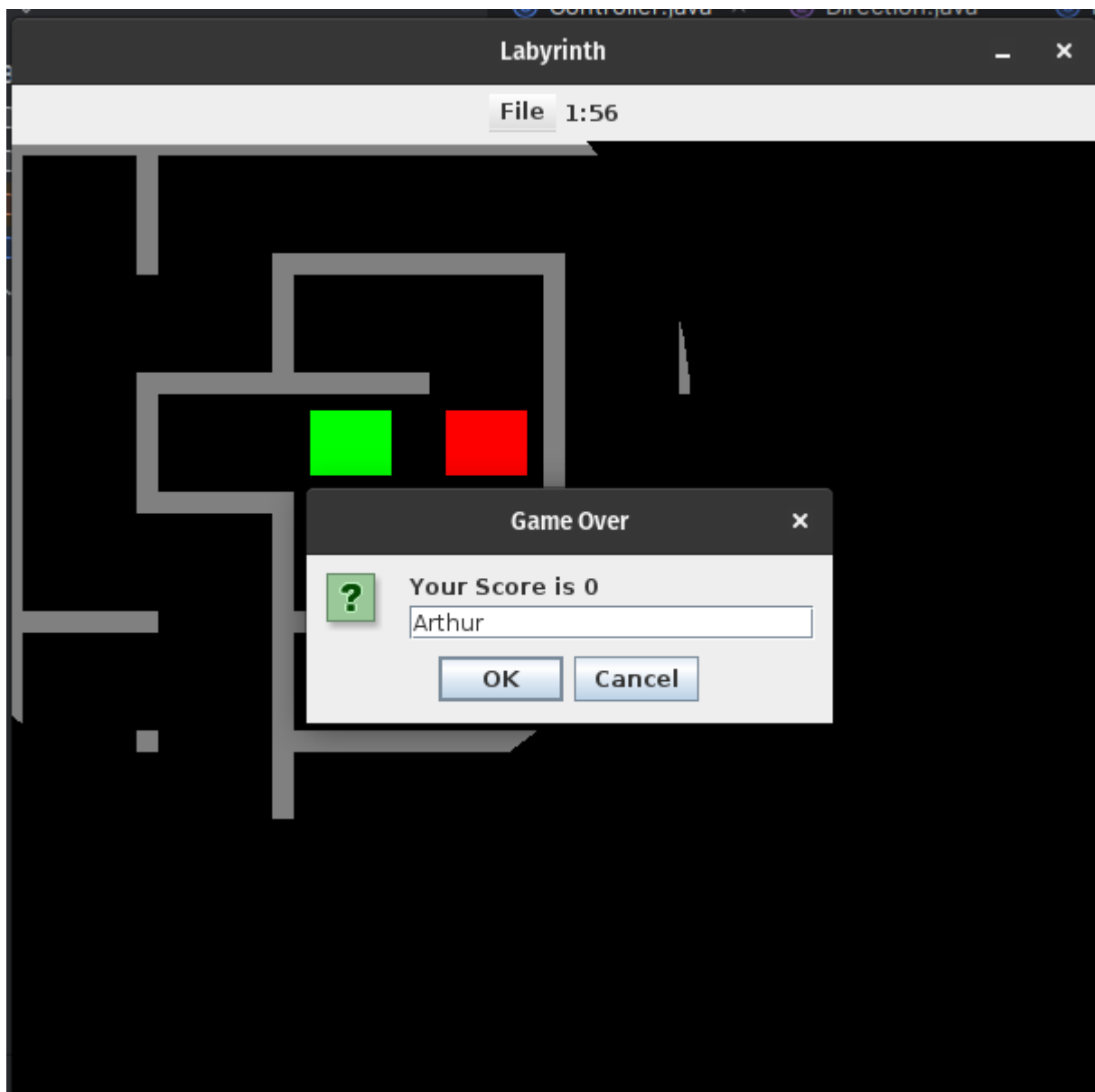
A látható része a labirintusnak követi a játékost.



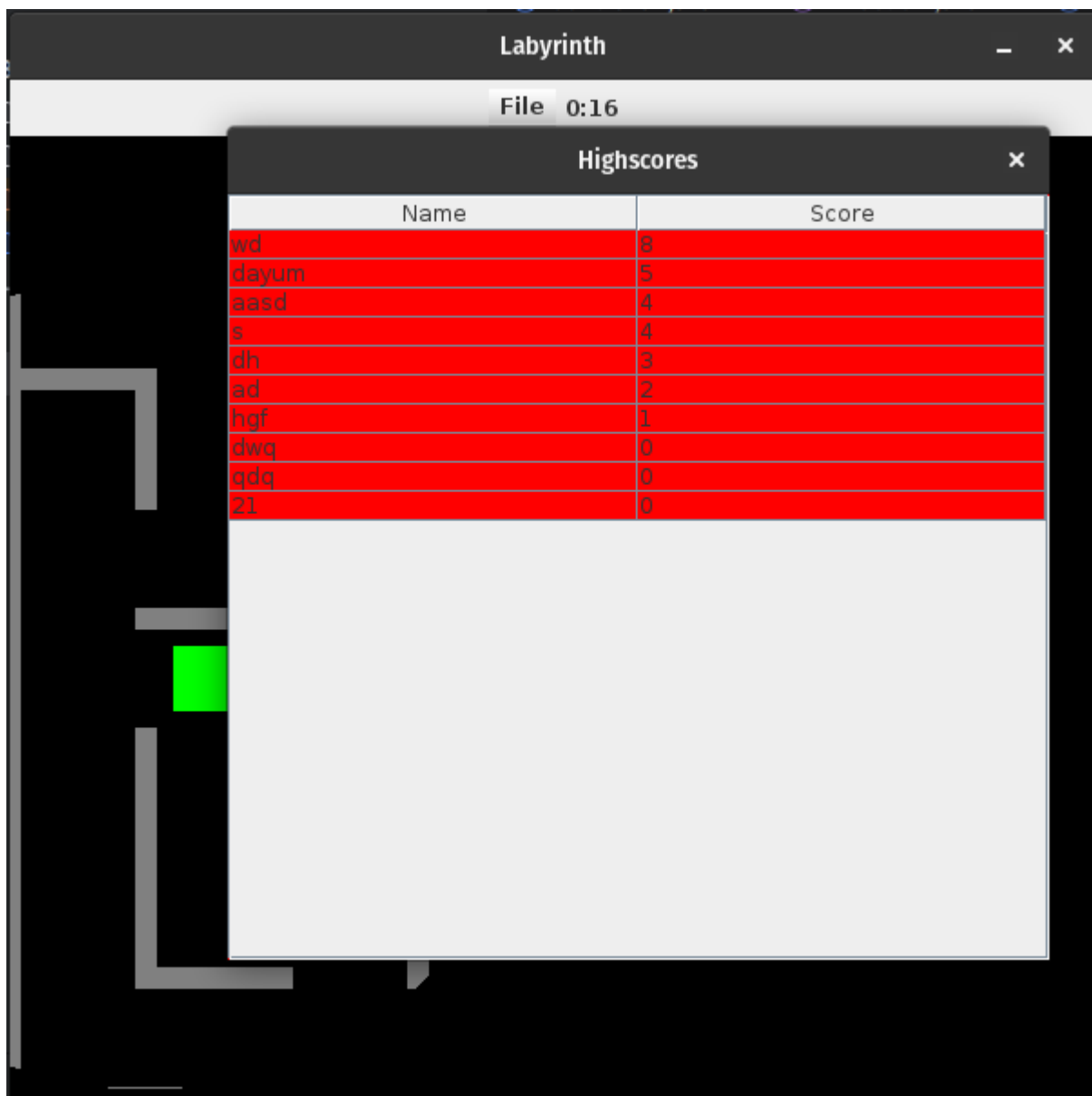
Több kiút van a labirintusból:



Elkapott a sárkány:



A játékosok pontjai megtekinthető:



## Tesztelési terv

Ez a program most kicsit többmindennel foglalkozik egyszerre. Maga a játékon kívül a labirintus generálása meg a pontok adatbázisba való eltárolása is külön tesztelhetők.

## User Stories

AS A	Játékos	
I WANT TO	Mozogni	
1	GIVEN	Fal felé mozgok
	WHEN	Megnyomom a W,A,S,D gombok egyikét
	THEN	A játékos nem mozog a pályán
2	GIVEN	Üres mező felé mozgok
	WHEN	Megnyomom a W,A,S,D gombok egyikét



	THEN	A játékos a megfelelő irányba megy
3	GIVEN	A sárkány cellája mellé mozgok
	WHEN	Megnyomom a W,A,S,D gombok egyikét
	THEN	A játék leáll és elmenthetem a pontszámom
4	GIVEN	Fut a játék
	WHEN	A sárkány a mellettem lévő cellára lép
	THEN	A játék leáll és elmenthetem a pontszámom
5	GIVEN	Fut a játék
	WHEN	A jobb felső sarokba érek
	THEN	Új labirintus generálódik, és nő a pontszámom
6	GIVEN	Érvényes lépést teszek
	WHEN	Megnyomom a W,A,S,D gombok egyikét
	THEN	Mozog velem a pálya látható része
7	GIVEN	A sárkány melletti cella felé mozgok, de van köztünk fal
	WHEN	Megnyomom a W,A,S,D gombok egyikét
	THEN	A sárkány nem kap el, és folytatódik a játék

AS A	Játékos	
I WANT TO	Játszani	
1	GIVEN	Elindítom a játékot
	WHEN	Betölt a pálya
	THEN	Csak a labirintus egy része látható
2	GIVEN	Üres mező felé mozgok
	WHEN	Megnyomom a W,A,S,D gombok egyikét
	THEN	A játékos a megfelelő irányba megy
3	GIVEN	Elindítom a játékot
	WHEN	Betölt a pálya
	THEN	Az időzítő el kezd számolni
4	GIVEN	Elkapott a sárkány
	WHEN	Bezárom a dialógus ablakot
	THEN	Új pályán kezdődik újra a játék
5	GIVEN	Elindítom a játékot
	WHEN	Betölt a pálya
	THEN	Több lehetséges menekülő út van a labirintusban

AS A	Játékos
------	---------

I WANT TO	Pontokat megtekinteni, menteni	
1	GIVEN	Már töltöttem fel pontszámot
	WHEN	Megnyomom a menüből a View Score pontot
	THEN	Megjelenik a pont táblázat csökkenő sorrendben
2	GIVEN	Elkapott a sárkány
	WHEN	Beírom a nevem a dialógus ablakba és OK-t nyomok
	THEN	A játék elmenti a pontszámom a megadott névvel
3	GIVEN	Elkapott a sárkány
	WHEN	Nem írok semmit a dialógus ablakba, vagy Cancel-t nyomok, vagy bezárom az ablakot
	THEN	A játék nem menti el a pontszámom