

Tóth Botond

3.beadandó/2.feladat

2021.12.02.

MQH41V

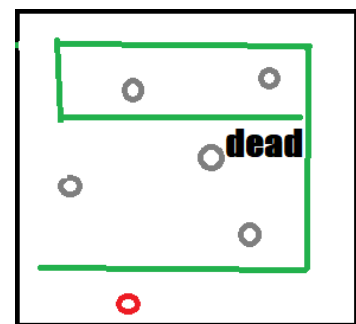
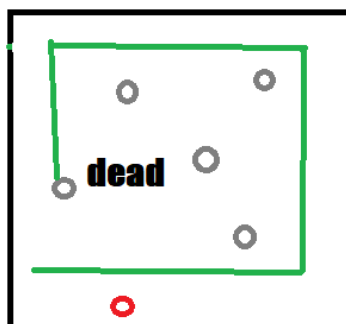
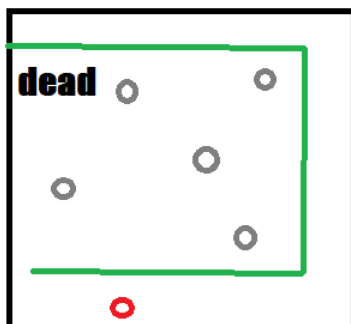
mqh41v@inf.elte.hu vagy tothbotond00@gmail.com

Feladat

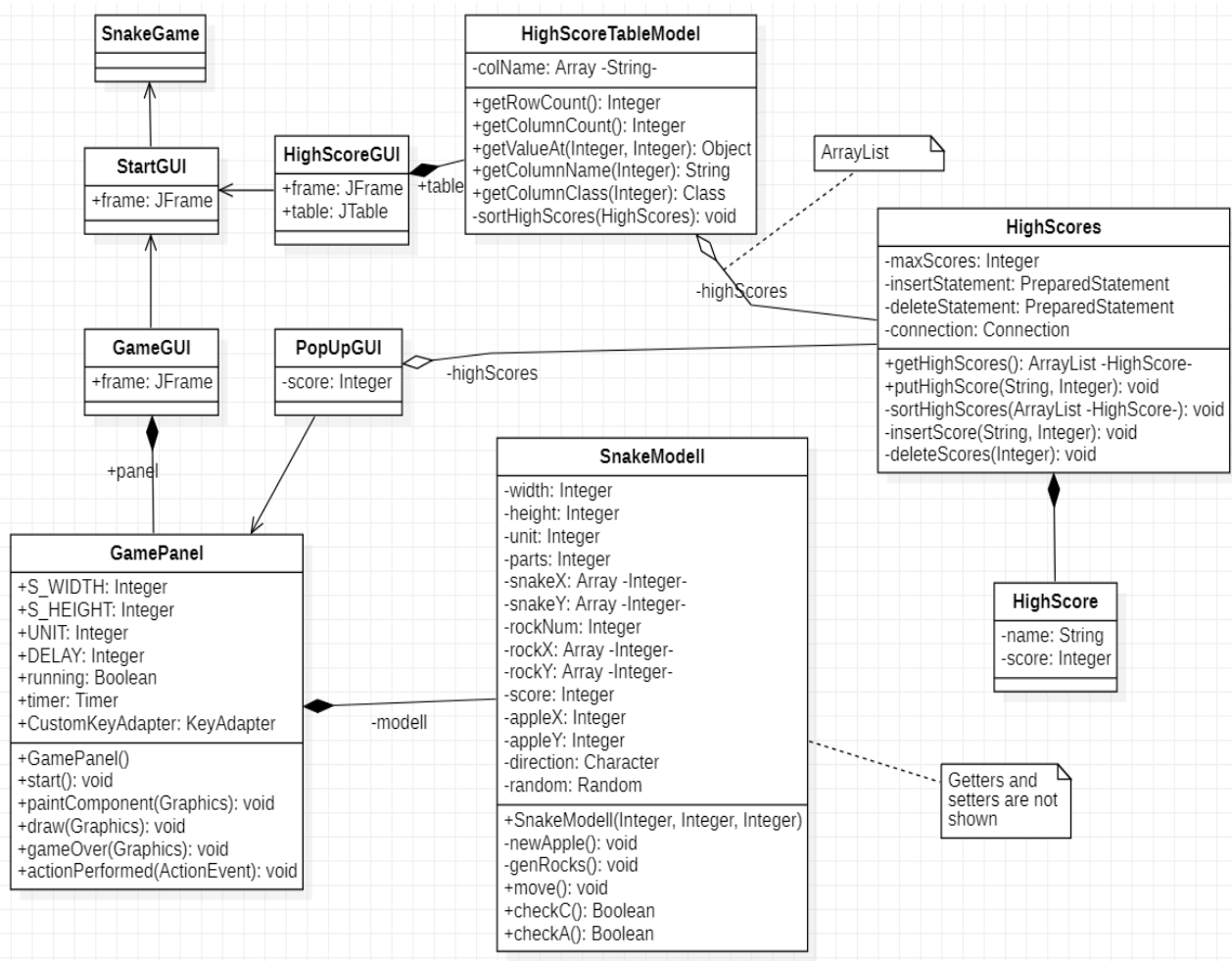
Kezdetben egy 2 egység (fej és csörgő) hosszú csörgő kígyóval kell felszednünk a sivatagos játéktéren megjelenő élelmet. A játéktéren egyszerre 1 elemőzsia lehet véletlenszerűen elhelyezve olyan mezőn, melyen nem a kígyó található. A kígyó a játéktér közepéről egy véletlenszerűen választott irányba indul. A továbbiakban a felhasználó a billentyűzet segítségével válthat majd irányt. Élelemhez érve, a kígyó mérete egy egységgel nő. A játékot nehezítse, hogy a sivatagban kövek is találhatók melyeknek, ha nekimegy a kígyó, akkor véget ér a játék. Abban az esetben is elveszítjük a játékot, ha a kígyó saját magának megy neki, vagy a pálya szélének. Ezekben az esetekben jelenjen meg egy felugró ablak, melyben a játékos a nevét megadva el tudja menteni az adatbázisba az eredményét, mely a játék során a kígyó által elfogyasztott élelem összege. Egy menüpontban legyen lehetőségünk a 10 legjobb eredménnyel rendelkező játékost megtekinteni, az elért pontszámukkal, továbbá lehessen bármikor új játékot indítani egy másik menüből

Megoldás terve

A modellben számon kell tartanunk a kígyónk minden pontját így a mozgást relatívan egyszerűen implementálhatjuk, hiszen a kígyó előre mozgásával minden elem eggyel el fog tolódni. A dolog amiért a pont jár fontos, hogy jó helyen generálódjon le, tehát nem szabad olyan helyre pakolnunk az elemet ahol éppen a kígyónk tartózkodik. Figyelni kell 3 fontos dologra: a kígyó nekiment-e a falnak, a kígyó nekiment-e egy kőnek és hogy a kígyó nekiment-e saját magának. Ezeket az ütközéseket egy mátrixban könnyen tudjuk felügyelni. A pontok számlálása egyszerű feladat, ezt majd egy SQL adatbázisba kell frissítenünk, és ezt az adatbázist majd később megjelenítenünk ha a felhasználó látni szeretné azt.



Osztálydiagramm



Osztályleírások

Modell:

SnakeModell: Az egész játékot reprezentáló osztály. Három fontos elem változása és reprezentálása az osztály feladata: az alma, a kövek és a kígyó. Az alma létrehozás kulcsfontosságú feladat : figyelni kell a kígyó és a kövek pozíciójára, nehogy belegeneráljuk valamelyik elembe. A kígyónál ütközési pillanatokat kell figyelni, almával való érintkezés esetén pontnövekedés, egyéb más objektummal (vagy a pálya szélével) való ütközés pedig halál. A kövek a modell elején legenerálódnak (lehetőleg nem a kígyóba) és statikusan a helyükön maradnak a játék alatt. A kígyó mozgása még egy fontos feladat. A kígyó a farkától eggyel előrébb csúszik, a feje pedig abba az irányba mozog ami legutoljára le lett nyomva.

HighScore: Egy nevet és egy pontszámot tartalmazó osztály. A játékosok pontjainak eltárolására és reprezentálására alkalmas.

HighScores: Az SQL adatbázisba való beszúrásért és lekérdezésért felelős osztály. A létrehozott JConnection-el írni és olvasni tud az adatbázisból. Fontos, hogy ez az osztály maximum "x" db elem megjelenítésére és tárolására szolgál (tehát tudunk benne pontosan 10 sort tárolni).

HighScoreTableModel: Egy ArrayList alapján előállítja a megfelelő formátumot a HighScoreGUI számára. Egy AbstractTableModel-ből származtatjuk.

UI:

GamePanel: A játék kirajzolásáért egy JPanel fog felelni. A JPanelen egy Timer fog futni ami egy megadott időközönként rajzolja majd újra a pályát. A jelenlegi állapot a modell alapján fog megjelenni és külső inputkora tudjuk majd a modellt közvetlenül befolyásolni. Ha a megoldás dob egy Game Over-t, akkor a rajzolás ciklusa megáll és a játék pontjait elmenthetjük és új játékot kezdhethetünk egy menüből.

PopUpGUI: A felugró ablak abban az esetben, ha meghaltunk. El tudjuk menteni a nevünket egy adatbázisba (persze csak akkor ha legalább a 10. játékosnál jobban teljesítettünk). A submit gombbal elküldjük a nevünket és a pontszámunkat az SQL adatbázisba.

GameGUI: A játékot tartalmazó JPanelt foglalja magába az ablak.

HighScoreGUI: A tíz legjobban teljesített neve és pontszáma jelenik meg egy táblázatban.

StartGUI: A kezdőképernyő, az alkalmazás elindításánál jelenik meg, és bezárásával az alkalmazás leáll. Innen nyitható meg a dicsőségtábla és maga a játék.

Érdekes algoritmusok:

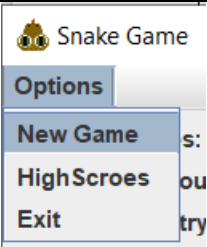

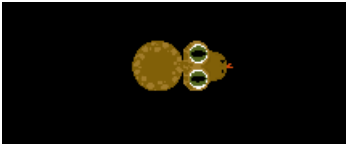


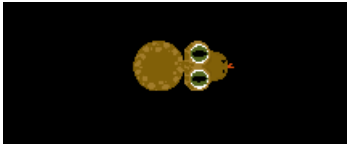




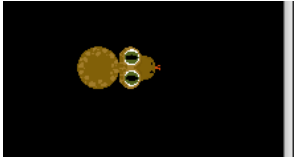

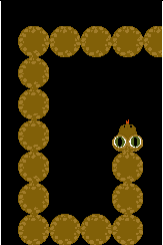

newApple(): Egy alma legenerálása az alapja az egész játéktér legenerálásának. Az egyik legfontosabb pontja ennek az algoritmusnak az ellenőrzések. Sajnos ha véletlenszerűséget szeretnénk belevinni a generálásba nem tudjuk kikerülni a kígyó részeivel és a kövekkel való összevetését a generált koordinátáknak. Éppen ezért szinte kiszámíthatatlan a futási ideje, hiszen nem tudhatjuk mikor fog az algoritmus egy olyan pontot találni amin éppen nem tartózkodik senki. (Persze evidens, hogy szinte lehetetlen, hogy nem talál ilyet, viszont a főciklus while-ja eldönthetetlen ideig fog futni)

move(): A kígyó mozgatása egy érdekes feladat. Sokféle képpen lehet nekiállni a feladatnak (persze a végeredmény ugyanaz lesz). A modellben implementált mozgásfajta a faroktól indul és egészen a fejig tart. A ciklus lényege, hogy a kígyó teste kövesse magát még el nem ér a fejéig (tehát kb. úgy mozog mint egy kukac). Ha elértük a fejét, akkor lényegében már csak a fix irányba kell elmozdítanunk azt és ebből fog következni, hogy a kígyó minden része megy majd a feje után (megj.: ezt természetesen fordítva is lehetett volna implementálni).


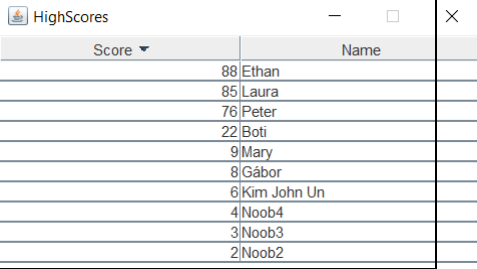
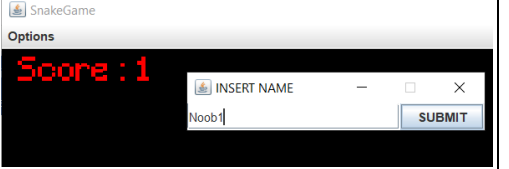

checkC(): A move() mellett a másik kulcsfontosságú függvénye a modellnek. Minden egyes mozgásnál figyelnie kell a modellnek az ütközéseket. Ezt nem lehet egyszerűbben implementálni, mint hogy minden lehetséges komponenssel ellenőrizni kell a koordináta egyezést. Az almával való érintkezésért a **checkA()** felel.

Tesztelési terv

Modell:

Teszt	Bemenet	Várt kimenet
A játékos elindítja a játékot		
A játékos a kígyóval mozog.	 pl. A „fel” gomb lenyomása	
A játékos a kígyóval helytelenül mozog.	 A „bal” gomb lenyomása	
A játékos felvesz egy almát a kígyóval		
A játékos neki megy egy kőnek		
A játékos neki megy egy falnak		
A játékos nekimegy saját magának		

SQL:

Teszt	Bemenet	Várt kimenet
A játékos nem menti el a nevét		
A játékos elmenti a nevét de nincs benne a top 10-ben		
A játékos elmenti a nevét és bekerül a top 10-be	