

Kígyó (Snake)

Varga Boldizsár - YE1BRV

Feladat leírása

Kezdetben egy 2 egység (fej és csörgő) hosszú csörgőkígyóval 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.

Terv

1. Osztályok

- **Snake:** A kígyó mozgását, növekedését, hajta végre, és ütközéseit ellenőrzi.

Adattagok:

body: A kígyó testét tároló lista.

direction: A kígyó haladásának iránya.

headImage: A kígyó fejének képe.

bodyImage: A kígyó testének képe.

Metódusok:

move(): A kígyó mozgását végzi el.

grow(): A kígyó növekedését végzi el.

hitSnake(): Ellenőrzi, hogy a kígyó saját magának nekimegy-e.

hitWall(int width, int height): Ellenőrzi, hogy a kígyó falnak ütközik-e a pálya szélén.

- **Food:** Az élelem generálását és elhelyezéséért felelős.

Adattagok:

foodposition: Az élelem pozícióját tároló Point.

Metódusok:

generateFood(): Generál egy új élelem pozíciót, figyelembe véve a kígyó és kövek elhelyezkedését.

getPosition(): Point: Visszaadja az élelem pozícióját.

- **GameBoard:** A játékkeret és grafikus megjelenését kezeli.

Adattagok:

snake: A játékban szereplő kígyó.

food: Az élelem a játékban.

stones: A kövek a játékban.

backgroundImage: A játék háttérének képe.

Metódusok:

generateStones(): Generál néhány követ a játékkeren.

drawStone(): Kövek kirajzolása.

drawFood(): Élelem kirajzolása.

drawSnake(): Kígyó kirajzolása

- **GameGUI:** A játék grafikus felületét, menüjét, a játék teljes logikáját kezeli.

Adattagok:

stones: A játékban található kövek listája.

snake: A játékban szereplő kígyó.

food: Az élelem a játékban.

timer: Az időzítő a kígyó mozgásához.

seconds: Az eltelt időt méri.

gameTime: Az időzítő a játék idejének méréséhez.

Metódusok:

viewHighScores(): Megjeleníti a legmagasabb pontszámokat.

startNewGame(): Elindít egy új játékot.

createMenu(): Létrehozza a menüt a GUI-ban.

checkHits(): Ellenőrzi, hogy a kígyó ütközik-e saját magával vagy a fallal.

checkStoneHit(): Ellenőrzi, hogy a kígyó ütközik-e kővel.

checkFoodEat(): Ellenőrzi, hogy a kígyó megeszi-e az élelmet.

updategameTime(): A játék során az eltelt idő folyamatos kiírása.
GameOver(): A játék végén kiírja a pontszámot és elmenti a legmagasabb pontszámokat.

- **HighScore:** A játékos nevét, pontszámát és időt tárolja.

Adattagok:

name: A játékos neve.
score: A pontszám.
seconds: Az eltelt idő másodpercben.

Metódusok:

getName(): String: Visszaadja a játékos nevét.
getScore(): int: Visszaadja a pontszámot.
getSeconds(): int: Visszaadja az eltelt időt másodpercben.

- **HighScores:** A legmagasabb pontszámok tárolása az adatbázisban.

Adattagok:

maxScores: A maximális tárolható legmagasabb pontszámok száma.
insertStatement: Az új pontszámok beszúrására szolgáló prepared statement.
deleteStatement: A pontszámok törlésére szolgáló prepared statement.
connection: Az adatbáziskapcsolat.

Metódusok:

HighScores(int maxScores): Konstruktor, inicializálja az HighScores osztályt és kapcsolódik az adatbázishoz.
getHighScores(): ArrayList<HighScore>: Visszaadja a legmagasabb pontszámokat.
putHighScore(String name, int score, int seconds): int: Frissíti a legmagasabb pontszámokat az új pontszámmal.
sortHighScores(ArrayList<HighScore> highScores): Rendezi a legmagasabb pontszámokat csökkenő sorrendben.
insertScore(String name, int score, int seconds): Beszúr egy új pontszámot az adatbázisba.
deleteScores(int score): Töröl egy pontszámot az adatbázisból.

- **Direction:** Felsorolás, ami a kígyó mozgásának irányait tartalmazza.

Implementációs Fejezet

Pálya Generálása

A pálya dinamikusan generálódik a GameBoard osztályban. A kígyó, az élelem és a kövek elhelyezkedését figyelembe véve így zajlik:

1. Kígyó: A kígyó kezdeti pozíciója a GameGUI osztályban kerül meghívásra, a kígyó pozícióját is figyelembe vesszük a köveknél és az ételnél.
2. Élelem: Az élelem létrehozása a Food osztályban történik. Az élelem lehetséges pozícióit egy listában tároljuk, figyelembe véve, hogy nem ütközhet a kígyóval vagy a kövekkel. Az étel pozícióját véletlenszerűen választjuk ki ebből a listából.
2. Kövek: A kövek létrehozása a GameBoard osztályban történik. A kövek véletlenszerűen kiválasztott pozíciókba kerülnek, de nem ütközhetnek sem a kígyóval, sem az élelemmel. A generateStones() metódus felelős ezért, a kígyó, az élelem és a már létező kövek pozícióit is figyelembe veszi.

Esemény-kezelő Párosítások

A játék esemény-kezelését a GameGUI osztály végzi, amely A JFrame-t és az alapvető Swing komponenseket hozza létre. Az esemény-kezelés így zajlik:

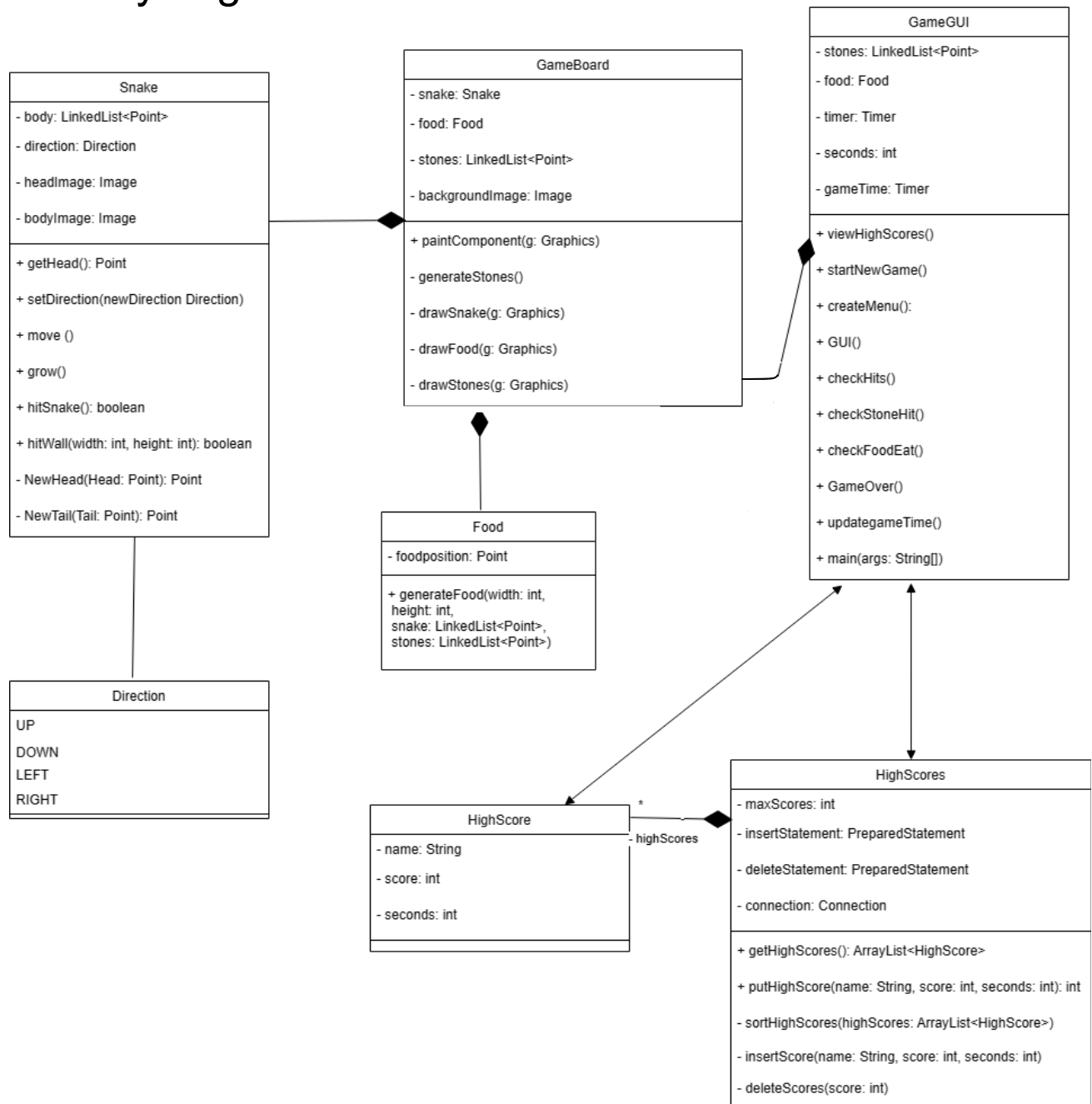
Billentyűleütések kezelése

A játékos irányítása a KeyAdapter-en keresztül történik, amit a GameGUI osztály tartalmaz. A keyPressed() metódusban észlelhetők a lenyomott billentyűk, és ennek alapján módosítjuk a kígyó irányát.

Menü gombok kezelése

A "View High Scores" és "New Game" menü opciókat a createMenu() metódusban állítjuk be, és ezekhez ActionListener-ek kerülnek. Az ActionListener-ek a viewHighScores() és startNewGame() metódusokat hívják meg, amelyek kezelik a kapcsolódó eseményeket.

Osztálydiagram



Tesztelés

Fehérdobozos tesztesetek

Leírás	Metódus	Elvárt kimenet
A kígyó falnak ütközik	hitWall(20, 15)	true
A kígyó a saját testébe fordul	hitSnake()	true
	move()	A kígyó az aktuális irányban mozog egyet
Direction = UP newDirection = Direction.DOWN	setDirection(Direction newDirection)	Az irány nem változik

Feketedobozos tesztesetek

Leírás	Elvárt kimenet
Játék indítása	Létrejön egy ablak, a 2 hosszúságú kígyó a pálya közepéről random irányba kezd haladni
Jobbra mutató nyíl billentyű lenyomása	A kígyó jobbra fordul(ha éppen nem balra haladt)
A kígyó élelemnek megy	Az élelem eltűnik, a kígyó halad tovább, 1-el nagyobb lett a mérete
A kígyó egy kőnek, saját magának, vagy a pálya szélének megy	Játék vége, párbeszédablakban meg lehet adni a nevet.
Game menüpontból „View High Scores”-ra kattintás	Jelenlegi toplista megjelenése párbeszédablakban
Game menüpontból „New Game”-re kattintás	Új pálya rajzolódik ki, a kövek, élelem véletlenszerűen új pozícióba kerülnek