Tesztelési módszerek

Kötelező program jelentés

Molnár Ádám (JX6VI7) 2020.05.09.

Motiváció

A feladat kiválasztásánál fontos szempontnak tartottam a kód átláthatóságát, és a program megértését. Az alkalmazás működése nem volt teljesen ismeretlen, ugyanis az ismert Worms World Party játék egy lebutított változatát valósítja meg a projekt, valamint a forráskód dokumentációt is tartalmaz ami külön segítette a program megértését.

Az eredeti projekt repoja: https://github.com/matthijsvk/Worms Game

Feladat

Beadandóm célja a *branch* lefedettség feltérképezése, növelése, valamint a teszt redundancia csökkentése.

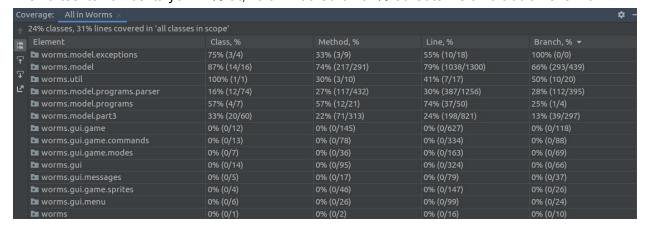
Induló code coverage

A feladatot az eredeti kódbázis lefedettség mérésével kezdtem. Mivel egy natív Java programról beszélünk, első lépésként *Maven* projektet készítettem belőle és hozzáadtam a szükséges függőségeket.

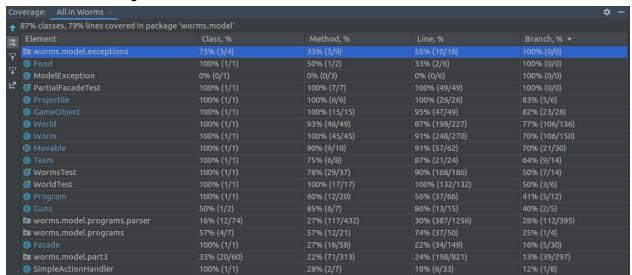
Az eredeti kódbázis *54* tesztet tartalmazott, ami a használt modelleket igyekezett lefedni. Ezek a tesztosztályok a következőek:

- PartialFacadeTest.java
- SimpleActionHandler.java
- WorldTest.java
- WormsTest.java

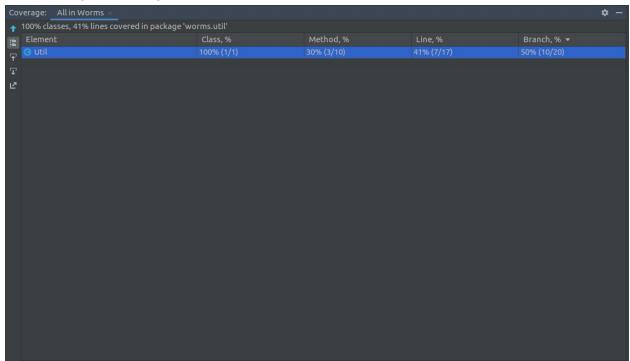
Ezek a tesztek az osztályok 24%-át, valamint a sorok 31%-át fedték le az alábbiak szerint:



Modellek lefedettsége részletesebben:



Utils csomag lefedettsége:



Megvalósítás

Főként azokra a csomagokra koncentráltam, aminek a branch lefedettsége kezdetben 0% volt. Így esett a választásom a worms.gui.game.commands csomag lefedettségének növelésére. A tesztek írására junit4-et használtam, és törekedtem a keretrendszer által szolgáltatott Mock objektumok használatára a rekurzió és a redundancia elkerülése érdekében. A tesztek írása során igyekeztem arra, hogy minden osztályhoz egyértelműen megtalálható legyen a hozzá írt teszt.

A csomagban található *Command* osztály egy *abstract* osztály, *final* metódusokkal, amiket nem tudtam mockolni, mert nem lehet példányosítani, így ezt az osztályt nem sikerült teljesen lefedni.

A *Move* osztály a játékban szereplő karakterek mozgatására való és az itt található függvények több branch-et is tartalmaznak (pl. *startFalling*). Ezekhez nem volt elég egy tesztet létrehozni, hogy a metódust teljesen lefedjem.

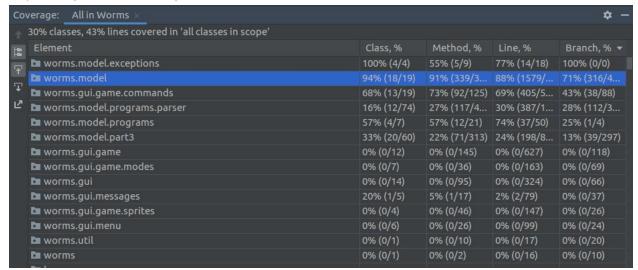
Az *AddNewWorm* illetve a *Jump* osztályokban található metódusok is több branchet tartalmaznak, ezért itt is több tesztre volt szükség.

Refaktoráltam az eredeti *Facade* osztályhoz írt teszteket, mivel ezek főként a *Worm* és *World* osztályok metódusait hívták meg, valamint *mock* adatokat sem használtak és egy-egy teszt a teljes hívási útvonalon ténylegesen végigfutott, tehát valójában ez az osztály nem csak és kizárólag a *Facade* osztály függvényeit tesztelte. A refaktorálás után a redundancia is csökkent, de kódlefedettségből veszítettem, mivel az eredeti tesztek a *Utils* csomag függvényeit is tesztelték, az újak pedig csak a *Facade* hatáskörében lévő függvényeket ellenőrzik.

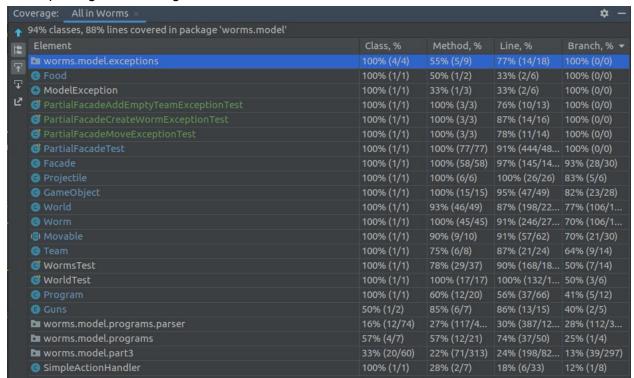
Eredmények

A tesztek számát sikerült 54-ről 176-ra növelni, a commands package branch lefedettségét 0%-ról 43%-ra, valamint a model package lefedettségét 66%-ról 71%-ra javítottam. A util package lefedettsége 50%-ról 0%-ra csökkent.

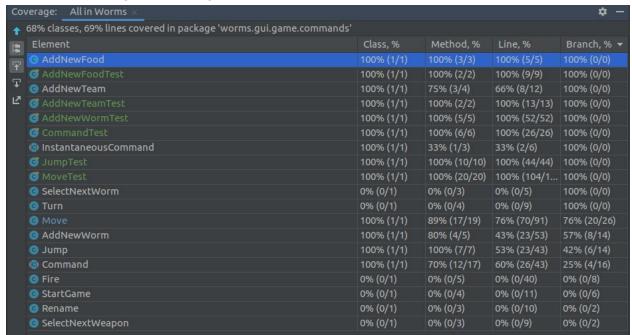
Teljes program lefedettsége:



Model package lefedettsége:



Commands package lefedettsége:



Zöld színnel a hozzáadott új teszt osztályok láthatóak.