Felhasználói kézikönyv

Bevezetés

A Zedországi Államvasutak (ZÁV) a megnövekedett késések miatt drasztikus döntést hozott. A vonatok gyakran azért késtek, mert a pálya foglalt volt, ezért úgy döntöttek eltávolítják a biztosítóberendezéseket. Ez teljesen biztonságos, hiszen jól képzett és tehetséges forgalmisták, mint te fogják a vonatokat irányítani. Természetesen ez egy nagy felelősség. Bármi incidens történik az a te és csakis a te hibád nem a vasút társaságé.

Ez egy fiktív történet. Bármilyen párhuzam valós személyekhez vagy vasút társaságokhoz kizárólag a véletlen műve.

Pályák

A beépített pályákat a főmenüben lévő számozott gombokkal lehet elérni. Egyedi pályákat a „Play from file” gombbal lehet elérni. A játékban a bal oldali bejáratokon vonatok fognak érkezni. A bejáratok feletti pöttyök mutatják, hogy milyen színű és mikor érkezik a vonat. 3 pötty jelenti azt, hogy azonnal érkezik a következő.

Vonatok

5 színű vonat van. A piros, kék és zöld vonatok a legegyszerűbbek. Nekik meg kell állni a velük megegyező színű peronoknál. A fekete vonatok nem állnak meg egyik peronnál sem. A fehér vonatok bármely peronnál megállhatnak.

Játékmenet

A vonatokat a váltókra kattintással lehet más irányba terelni. Két vonat nem ütközhet össze, különben veszítesz. A vonatokat le lehet lassítani azzal, hogy nyomva tartod rajtuk az egered. Ha egy vonat elhagyja a pályát anélkül, hogy megállt volna (kivéve a fekete vonatokat) akkor is veszítesz.

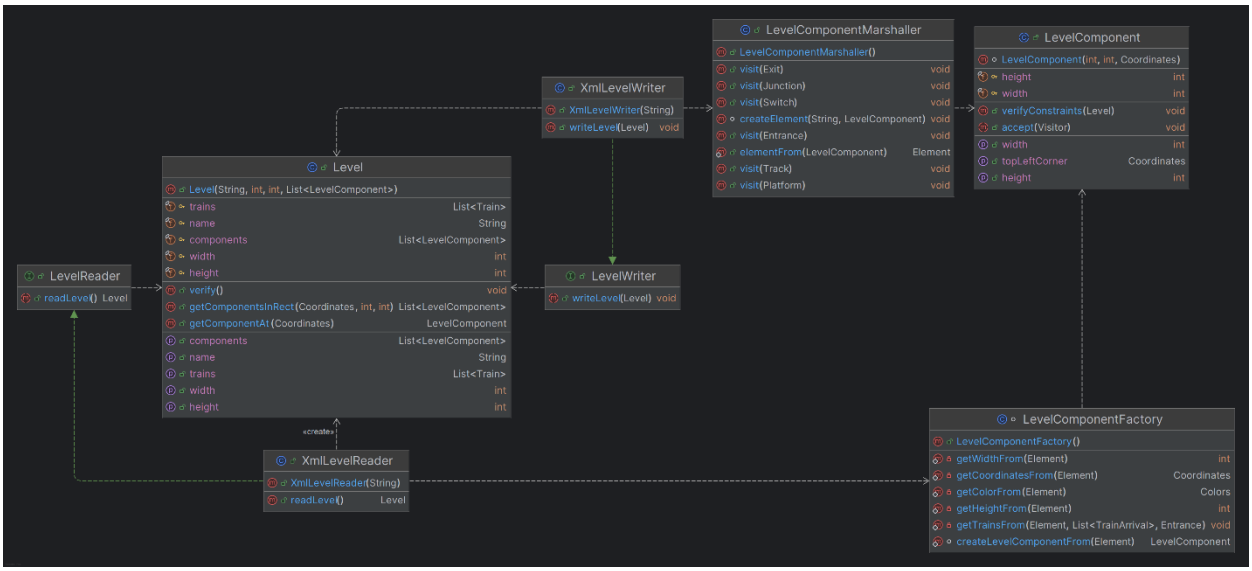
Pálya szerkesztő

A pálya szerkesztő segítségével létre tudod hozni a saját pályádat. Importálhatsz meglévő pályát vagy kezdheted az alapoktól. A jobb oldali listából választhatsz pálya elemeket, amiket lerakhatsz. Ha valamit rossz helyre tettél akkor csak kattints rá és töröld ki a jobb oldalon lévő gombbal. Bátran kísérletezz!

Fejlesztői dokumentáció

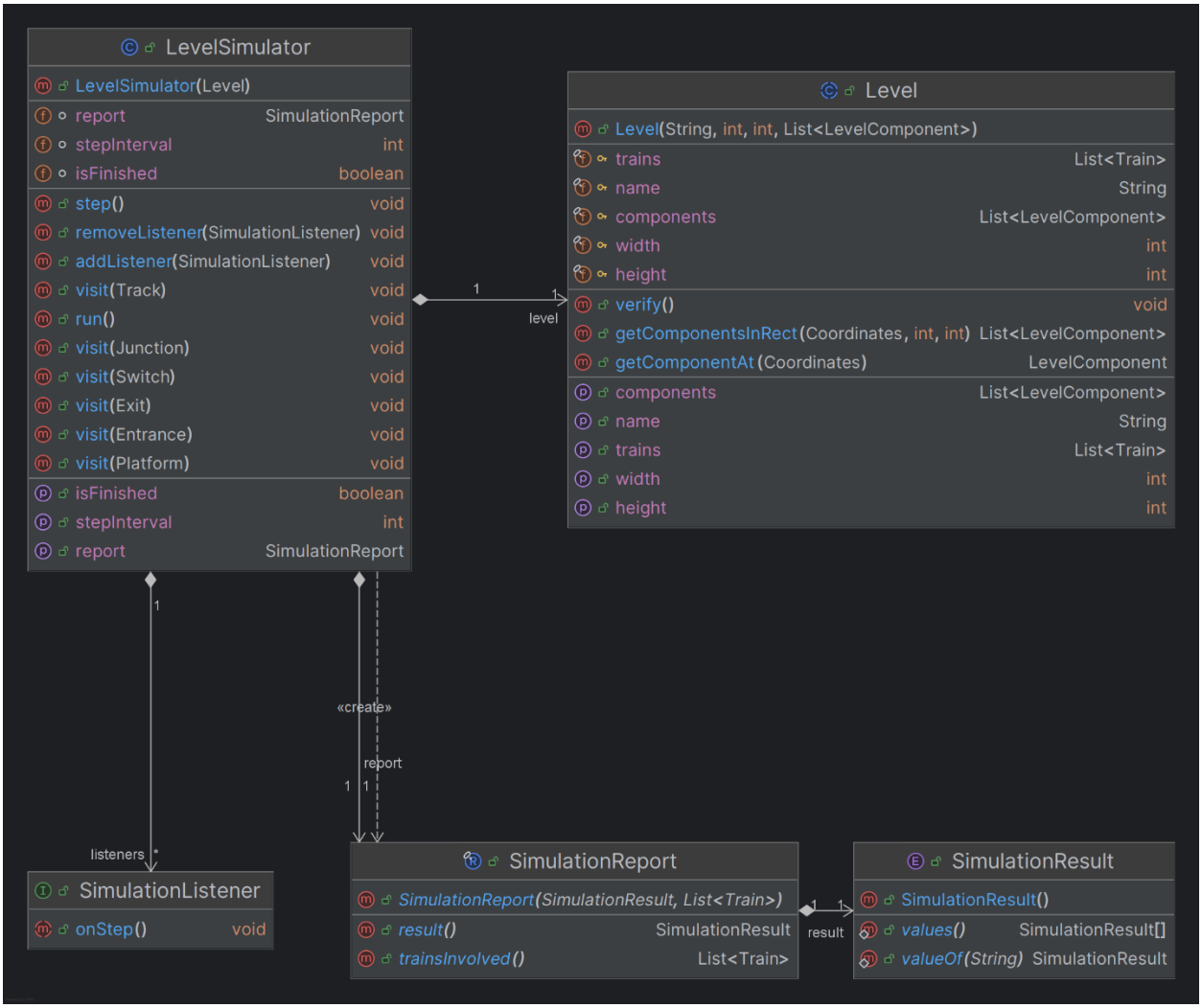
Osztálydiagramok

A fájl kezelés osztálydiagrammja



A modell osztálydiagrammja

A szimuláció osztálydiagrammja



Modell

A pályát leíró modellt teljesen el akartam különíteni a szimulációtól és a megjelenítéstől. Gondot okozott, hogyan adjak egyedi megjelenést a Level-ben lévő heterogén kollekcióban lévő komponenseknek. Erre a Visitor mintát alkalmaztam. Sajnos ez igényel valamennyi kód ismétlést, mert minden osztálynak kell egy metódus a Visitor interfészben és az osztályban is, de még így is elegánsabb, mint instanceof-al megkülönböztetni az egyes komponenseket.

Szimuláció

A szimuláció felelőssége tetszőleges sebességgel léptetni a modellt és a modell állásából eldöntse véget ért-e a szimuláció és sikeres volt-e.

Ellenőrzés

A pályák beolvasása után egy ellenőrzés lépés következik, amiben az egyes komponensek követelményei le lesznek ellenőrizve, valamint a komponensek közötti kapcsolatok is létrejönnek. Jelenlegi implementációja nem túl hatékony egymásba ágyazott ciklus, aminek $O(n^2)$ komplexitása van, ahol n a komponensek száma. Mivel ez csak egyszer fut le betöltéskor elfogadhatónak tartom.

Nyelvi eszközök

Próbáltam a nyelv minél több elemét felhasználni fejlesztés közben, például szálkezelést és rekordokat a szimulációhoz, valamint számos helyen használtam lambda kifejezéseket.

Külső könyvtárak

- A program XML kezeléséhez a JDOM2 nevű könyvtárat használja.
- A program egységteszteléshez a JUnit5.8.1 és a testng könyvtárat használja.
- A függőségek pontosabban az „src.iml” fájlban vannak definiálva.

Dokumentálás

A metódusok dokumentálásához Javadoc kommenteket használtam a forráskódban.