Xml – projektmunka

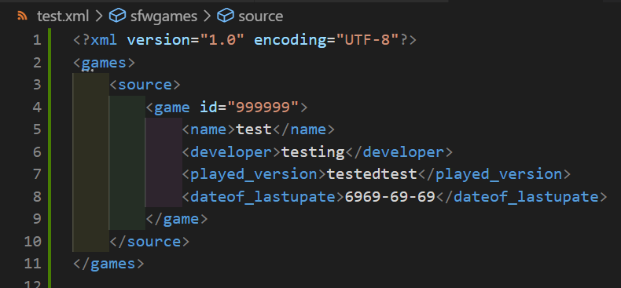
2023.05.07 20:15

1. Alapok

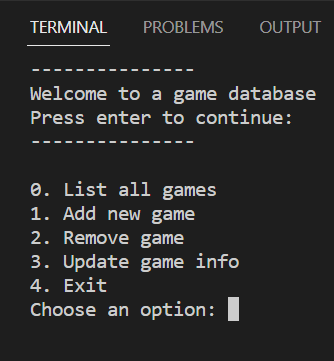
Alap gondolat, hogy egy könnyű és szimpla játék információ tárolása. Mivel sok játék van, ahol nem emlékszem mikor vagy melyik verziójával játszottam, így ez egy hasznos projektnek tűnt. Habár elsőre nem tűnik nagynak, de egy specifikus weboldal, ahol ID alapján vannak indie játékok, ez a program, ha majd automatizálva lesz, egy megváltás lehet.

Mivel Java mindenhol elfut probléma nélkül, könnyen bővíthető, így a legjobb valószínűleg egy szimpla Java programban és Xml fájlban tárolni az adatokat. Hogy lehessen használni olyan környezetben, ahova csak terminállal férünk hozzá, így tökéletesen futtatható a program.

2. XML formátum

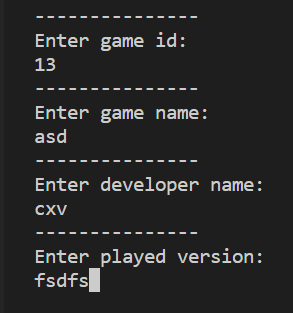
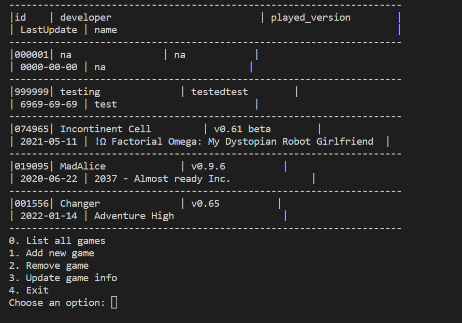
 Első körbe a <games>-en belül megadható minden, a <source>-ban meg lehet adni, ha esetleg több forrásból szeretnénk tárolni a jövőben. <game>-ben megadunk neki egy egyedi ID-t, ennek köszönhetően nincs kettő ugyan olyan. Ezen belül megadhatunk bármit és könnyen bővíthető is a szerkezet.

* <name> - Játék neve
* <developer> - Fejlesztő neve
* <played\_version> - Utolsó verzió, ami használva volt
* <dateof\_lastupdate> - Ez automatikusan az aznapi dátumot tárolja amikor utoljára módosítva volt az adat.



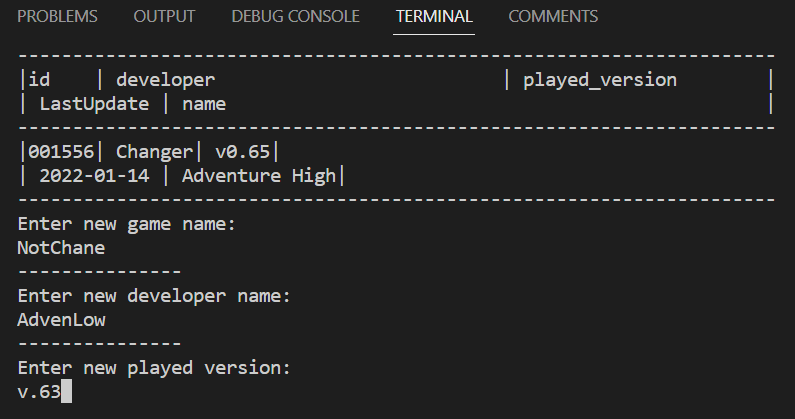
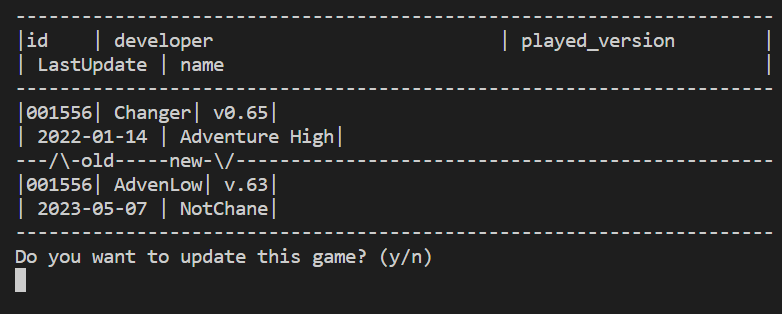
3. Program működése

Amikor elindítjuk a programot, egy kisebb menü fogad minket, ahol kiválaszthatjuk, hogy mit szeretnénk tenni. Ekkor a Scanner egy számot vár és amíg nem adunk meg egyet, addig nem megy tovább. Természetesen a 4-esre kilép a program. XML részleteket a következő pontban fogalmazom pontosabban.

Az 1-es opcióra egy sima loop funkció lép életbe, ami lekéri az összes az adatokat a fájlból, majd kiírja. Itt nincs semmit különleges.

Az első és második opció hozzáad és eltüntet adatokat az adatbázisból. Mind a kettő funkció el van látva ID ellenőrzéssel, szóval nincs kettő ugyan olyan és természetesen nem lehet nem létező elemet törölni.

A harmadik funkció kér egy ID-t majd lekérdezi a hozzá rendelt adatokat, szóval ezek után meg lehet adni a kívánt változtatásokat.



Hogy egy kis optimalizálás legyen a terminál dinamikusabban egy külön funkcióban van benne és mivel sokszor kell frissíteni a fájlt, igy a mentés is egy külön funkcióba lett írva.

4. Adatkezelés

A program, ahogy említve volt a 3. bekezdésben, XML dokumentum objektum modell-t (DOM) használ. Kutatás alapján erre a kicsi, de hasznos projekthez tökéletes. A DOC formátum egy nyelv és platform független objektum standard modell, ami HTML, XML és hasonló formátumú fájlok interakcióját végzi. Tehát kezeli az írást, olvasás és módosítását a fájloknak. XML a gyakori strukturált adatszerkezet helyett, ahol táblázat-szerűen vannak az adatok, félig strukturált adatszerkezetet használ. A másik különlegesség, hogy minden String, ha számot szeretnénk tárolni akkor szöveggé kell alakítani mentés elött, majd a szöveget olvasáskor vissza kell alakítanunk számmá.

Java-ban egy DucumentBuildFactory-val egy DocumentBuilder-t hozunk létre. Ez arra kell, hogy a Builder tudjon foglalkozni az XML fájlunkkal, amit meg is adunk neki, hogy foglalkozzon vele. Mivel XML-ben minden adat node így különböző adatokra különbözőképpen hivatkozunk. Ha sikeresen ki is nyertünk adatokat valahogy írnunk is kell.

Először létre kell hozni element-eket azzal a névvel, amivel menteni szeretnénk. Ezeket következőnek összekapcsoljuk ahogy szeretnénk, hogy legyen tárolva. Majd ezeknek mind meg kell adni egy értéket. Ennek az új elemnek, amiben adatokat tárolunk, megadhatunk egy attribútum értéket, majd ezt adjuk tovább mentésre. Mentéshez kell egy TransformerFactory ami ad nekünk egy Transformert. Ennek kell megadni StreamResult-al a kimeneti fájlunkat és mellé a DOMsourceunkat. A DomSource segít, hogy egy Document formátumot átalakítsunk, amiben az adataink vannak és tovább adjuk mentésre. Ennek a végén kapjuk meg a végleges „új” XML fájlunkat.

Megjegyzés:

1. Az eredeti itt: <https://github.com/DiamondPRO02/nsfw-game-manager>
2. A beküldött verzió itt elérhető: <https://github.com/DiamondPRO02/Program-modszertan-Java/tree/master/src/ZH_project>
3. XML or the relational model <https://www.ibm.com/docs/en/db2/11.5?topic=overview-comparison-xml-relational-models>