

OVRT Projektértékelés

Csapat adatok

Értékelt projekt: JPED - South Park

Értékelő csapat neve és tagjai:

KetyerePorontyok

Tagok:

- Tóth Botond (projektmenedzser)
- Péter Roland
- Szkocsovszki Zsolt
- Terhes Balázs

Tervezési minták

Talált tervezési minták

- Singleton
 - JsonSerializerStrategy.cs
 - XmlSerializerStrategy.cs
 - JsonDeserializerStrategy.cs
 - XmlDeserializerStrategy.cs
- Strategy
 - JsonSerializerStrategy.cs
 - XmlSerializerStrategy.cs
 - JsonDeserializerStrategy.cs
 - XmlDeserializerStrategy.cs
- Builder
 - PartsDictionary.cs
- Observer
 - Statistics.xml.cs
 - SumStatistics.cs
 - MeanStatistics.cs
 - MaxStatistics.cs
 - MinStatistics.cs
 - ChangeablePartStatistics.cs

Lehetséges tervezési minták

- Template method minta: A Statistics osztályok, amikre az Observer Design Patternt alkalmazták, mindegyike tartalmaz egy OnNext metódust, aminek az eleje egy null check mindenhol. Ezt kiemelhették volna a közös CharacterStatistics.cs osztályba, így alkalmazva a Template Method minta előnyeit.
- Strategy minta bővítése: a már meglévő fájl betöltő és elmentő stratégiák mellé fel lehetne venni egy háttér és egy rajz betöltő, illetve elmentő stratégiát is (ezek most külön-külön a MainWindow.xaml.cs fájlban vannak definiálva). Így egy helyre, egy scope-ba kerülhetne az összes mentés és betöltés funkció, átláthatóbb lenne a kód.
- Null Object minta: nagyon sok helyen van kézi null check az alkalmazásban, sokat javítana egy null object minta alkalmazása az olvashatóságon, egyúttal csökkentve a kódsorok számát

Refaktoring ötletek

- Kódkommentezést érdemes lehetne újragondolni, hiányos, ahol van, ott pedig nem sok tényleges információt ad (pl. `Statistics.xaml.cs` - "Interaction logic for `Statistics.xaml`")
- Konstansok, enumok használata - túl sok helyen van beégetett szöveg, érték, pl.: `CharacterVisual.cs:65.` sor - a komponensek típusa lehetne enum, string helyett, `CharacherVisual.cs:43.` sor - beégetett fájl elérési útvonal
- Kódolási stílus egységesítése - kód olvasásakor többféle stílus felfedezhető, pl. lokális-globális változók használata, néhol string, néhol saját enum használata
- Adattagok láthatóságának szigorúbbá tétele - nagyon sok public adattag található a kódban, IDE alapján nagy részük lehetne private vagy protected
- Saját hibaosztályok teljes hiánya - legtöbb helyen egy "return null" statementtel vannak levédve a hibás működést eredményezhető kódrészletek, érthetőbb lenne ilyen esetekben egy-egy megfelelően elnevezett és kezelt kivétel
- Kódból kinézve úgy tűnik, hogy egy Firebase kiegészítő funkció is be volt tervezve az alkalmazáshoz (`FirestoreSaveContext.cs`, `FirestoreLoadContext.cs`), azonban ezek nem lettek implementálva, nyugodtan törölhetőek lennének
- `App.xaml.cs`-ben kikommentezett kód található, törölhető lenne, git history-ból könnyen visszakereshető, ha bármire kellene a későbbiekben
- Karakter építéséhez tartozó fájlok adatbázisban történő tárolása (új ruhák, hátterek stb. hozzáadásával könnyen kezelhetetlenné válhat a projekt, valamint ez a git repo méretének felesleges növekedését okozza minden committal)
- Több helyen maradtak végül nem használt usingok. Ezeket érdemes lett volna commitolás előtt kiszedni. A `Ctrl + R + G` billentyűkombináció segítségével ez automatikusan is végrehajtható.
- Vannak olyan metódusok, melyek annyiból állnak, hogy létrehoznak egy változót és visszatérnek vele. Ebben az esetben a változó létrehozása felesleges, vissza lehet térni egyből a neki adott értékkel. Ilyen található például a `CharacterVisual.cs` `ToBitmapSource` metódusában.
- Találhatóak olyan metódusok, melyekre 0 referencia van. Ezeket érdemes lehet utólag törölni.
- Bizonyos helyeken (pl. `Statistics.xaml.cs` 76. sorában a foreach ciklus változója) lehetne beszédesebb változóneveket használni, mert az olvasó számára nem feltétlenül nyilvánvaló, hogy mit takarnak az elnevezések.

Futtatási nehézségek

A projekt elindításához semmilyen segédlet nem található meg. Nincs pl. egy readme fájl, ami felsorolná a szükséges környezeti paramétereket, eszközöket, a többszöri sikertelen indítás után a hibaüzenetekből kell kikeresni, hogy mi miatt nem akar éppen elindulni az alkalmazás.

A .Net Core 3.1 verziójának telepítése után említésre méltó futtatási nehézséggel nem találkoztunk, sikerült a csapattagoknak elindítani.

Talált hibák

- Új projekt létrehozásakor vagy betöltésekor nem frissülnek a statisztika adatok.
- Randomizált karakternél az üres elem is kiválasztásra kerülhet.

Fejlesztési ötletek

- Elkészített karakter képként történő kiexportálása
- A karakterek valamilyen felületen (pl. Facebook) történő megosztása, az alkalmazáson belülről
- Mások által készített karakterek megtekintése, értékelése

Értékelés

Minták használata

A használt minták logikusak, érthetőek, a projekt szempontjából hasznosak

- Egyáltalán nem hasznos, erőltetettek a minták
- Inkább nem hasznos, de bele lehet magyarázni
- Inkább hasznos, logikusak a minták, azonban néhány részük javításra szorulna
- Maximálisan hasznos, minden minta minden komponense hasznos

A projekt felépítése

A projekt mennyire átgondolt, jól felépített.

- A felépítés rossz, összezsapott
- Nagyjából érthető, azonban vannak benne kevésbé átgondolt részek
- A projekt érthető, azonban a felépítésén lenne mit javítani
- A projekt minden része jól felépített

Rosszul alkalmazott minta

- Van
- Nincs