**ZORK – požadavky**

\*scénář příběhu je nutné si vymyslet - nic složitého nečekám

\* v b’ase projektu jsem schválně udělal pár chyb -> bude třeba je najít a opravit

\* přepracujte CommandLineUi na Singleton (z principu by nám něměly v JVM existovat 2 instance této třídy)

\* dbejte na to, aby hra nespadla nesprávným zadání uživatelského vstupu, pokud uživatel zadá něco špatně, měl by o tom být informován

\* a měl by mít možnost hrát dále

\* data hry, které se mění by měla být umístěna tak, aby šlo jednoduše implementovat resetování hry (základ tam je)

\* Command pattern využijeme k zapouzřování logiky nad příkazy (základ tam už je)

\* je třeba stávající třídy upravit tak, abychom zajistili immutabilitu jednotlivých příkazů a zároveň si do příkazu propagovali data

\* pokud toto nedodržíme, nedokážeme udělat reset bez přeinicializování commandů, což je zbytečné

\* implementujte příkaz ‘jdi’, který dokáže procházet prostory, vždy do východů dané místnosti

\* implementujte třídu reprezentující nepřítele (Enemy), použijte kombinaci Flyweight + Builderu na neměnné části nepřítele

\* určitě bude generovat nějaké poškození (rozsah) a bude mít životy

\* implementujte třídu reprezentující zbraň (opět rozsah poškození)

\* k tomu příkaz ‘nasaď’, kterým zbraň sebereme z místnosti, pokud vezmeme další zbraň, stávající zahodíme do aktuální místnosti

\* k tomu příkaz ‘útok’, který z rozsahu náhodně vybere poškození nasazené zbraně a zasadí ho nepříteli

\* pokud zbraň nasazená není, tak nelze útočit

\* pokud tímto nepřítele nezabijete, útočí nepřítel

\* nepřítel útočí i v případě, že jsme v místnosti a snažíme se provádět jiné příkazy než útok (nemůžeme mu tedy vysbírat loot aniž by na nás neutočil)

\* implementujte příkaz ‘restart’, který resetuje data hry do původního stavu

\* implementujte příkaz ‘konec’, který ukončí hru

\* hra skončí dále končí, pokud uživatel porazí finálního bosse (nějaký special Enemy)

\* ten bude ve speciální místnosti, která bude zamčená -> bude třeba získat klíč

\* ten budeme muset vložit do inventáře (fixní kapacita)

\* tzn. příkazy ‘seber’ a ‘polož’ z inventáře

\* pokud v inventáři detekujeme speciální předmět (klíč), umožníme provést příkaz jdi do zamčené místnosti

\* můžete si vymyslet další předměty jako štít, nějaké lektvary na životy a podobně, pak ale bude třeba příkaz ‘použij’

\* za to budou pochopitelně nějaké ty body navíc

\* upravte log4j2.xml a kód tak, aby se do souboru prikazy.log zaznamenavaly zapsane (jen do souboru, ne do konzole)

\* použití zalogování přes logovací framework je možné vidět v CommandLineUi

\* uživatel hraje přes příkazovou řádku, je tedy nežádoucí, abychom naše vlastní vývojářské informace vypisovali do System.out

\* je tedy nežádoucí, aby uživatel v konzoli vidět aktuálně vylogovanou hlášku "application started"

\* právě k tomu slouží logovací frameworky = veškerá konfigurace v log4j2.xml

\* pozor na to, aby tam byl zalogovaný i čas zadání příkazu (třeba upravit přes pattern)

\* implementujte Observer pattern na třídu představující inventář (inventář bude Publisher)

\* pokud přidáme/odebereme předmět z inventáře, registrovaný Listener jen vytiskne aktuální obsah inventáře

\* implementujte testy pro jednotlivé příkazy (jednotlivé implementace třídy Command)

**sečteno podtrženo**

\* příkazy: jdi, konec, restart, nasaď, útok, seber, polož, rozhlédni se

\* snažte se, ať jsou výpisy smysluplné a ať se dá hra hrát - nechceme, aby uživatel musel pořád psát "nápověda"

\* tj. při vstupu do místnosti by bylo fajn, abyste vypsali nepřátelé, předměty, východy atd.

\* při souboji je fajn, aby se uživatel vždy dozvěděl, kolik má života on i nepřátelé atd.

Složitost hlavího scénáře si představuju tak, že se po startu hry zjevíme v nějaké místnosti, bude tam např. klacek s malým poškozením, půjdu do místnosti, tam bude nepřítel, buďto slabý nebo silný -> podle toho ho buď porazíme nebo ne. Z místnosti slabého nepřítele získám loot pro poražení silnějšího, tomu v místnosti seberu klíč pro místnost s final bossem. Porazím -> konec hry.

Samozřejmě se fantazii meze nekladou, dejte si ale pozor, ať nejdřív uděláte to co máte v zadání a pak se starejte o ‘rozšíření’.

\*\*DEADLINE: 14.1.2024 23:59\*\*<br>

\*\*Odevzdat na git. repo\*\*