

UFO-medical

Semestrální práce z PT

(Standardní zadání)

Datum: 15. 12. 2015

Vypracovali: Michal Štrunc, Jakub Váverka

Mail: [jakub.vaverka@email.cz](mailto:jakub.vaverka@email.cz), [mstrunc@stundents.zcu.cz](mailto:mstrunc@stundents.zcu.cz)

GitHub: https://github.com/Planet-Express/PT

Specifikace problému

Meziplanetární společnost "UFO-medical" zásobuje naši galaxii léky na vesmírný mor. Všech 5 zdrojových centrál, dokáže vyprodukovat dostatečné množství léků pro všechny planety, které zásobují. Zásoby jsou distribuovány jednotlivým planetám v závislosti na přijaté objednávce. Zásoby léků jsou pak rozváženy na měsíční bázi. Samozřejmě čas počítáme ve „hvězdných“ měsících, kdy má každý měsíc 30 dní (24 pozemských hodin), protože na každé planetě trvá lunární měsíc jinak dlouho. Naše společnost zásobuje dohromady 5000 planet. Každá planeta má 100 tisíc až 10 milionů obyvatel. Jejich počet na jednotlivých planetách je určen normálním rozdělením pravděpodobnosti. Dá se ale říci, že nejvíce planet má kolem 3 milionů obyvatel. Každý obyvatel potřebuje k přežití 1 balení léku na měsíc. Každá planeta je schopna pokrýt měsíčně 20-80% zásob ze svých vlastních zdrojů. Toto číslo se ale každý měsíc mění, protože výroba léku není jednoduchá. Pokud tedy společnost nedokáže daný měsíc na planetu dovézt potřebné léky, tak zemře část obyvatel, pro které domácí produkce nestačí. Pokud počet obyvatel planety klesne pod 40 000, tak se stane pro společnost nezajímavou a přestane sem léky dovážet. Galaxie má přibližnou rozlohu 800x800 LY (světelných let) a pro zjednodušení se dá zobrazit ve dvou rozměrech. Vzdálenosti mezi jednotlivými planetami jsou minimálně 2 LY. Mezi planetami vedou obchodní cesty. Každá planeta má obchodní cestu s pěti nejbližšími planetami, bohužel ne všechny cesty lze zabezpečit, a tak 20% cest hrozí útok vesmírných pirátů, kteří ukradnou veškerý náklad. Z dlouhodobých statistik společnost zjistila, že šance na přepadení lodi na nebezpečné cestě je 10%. Společnost disponuje transportními loděmi, které jsou schopny převézt 5 milionů balení léku. Jejich rychlost je 25 LY/den a nakládka i vykládka nákladu trvá 1 den. Loď se po vyložení nákladu vrací do své mateřské základny.

Dodatky

• Příslušná data si vygenerujte.

• Simulaci distribuce léků simulujte po dobu jednoho roku (12\*30 dní) po dnech.

• Počet transportních lodí sice není omezen, lodě ale samovolně nezanikají, po vyložení celého nákladu se vrací nejkratší cestou zpět do mateřské centrály a jsou připraveny na další cestu.

• Začínáte ve dni 1 prvního měsíce, předpokládejte, že k tomto dni mají planety dostatek léků na aktuální měsíc.

• Planety odesílají objednávky na následující měsíc vždy začátku aktuálního měsíce (tedy objednávku na říjen odesílají 1. září).

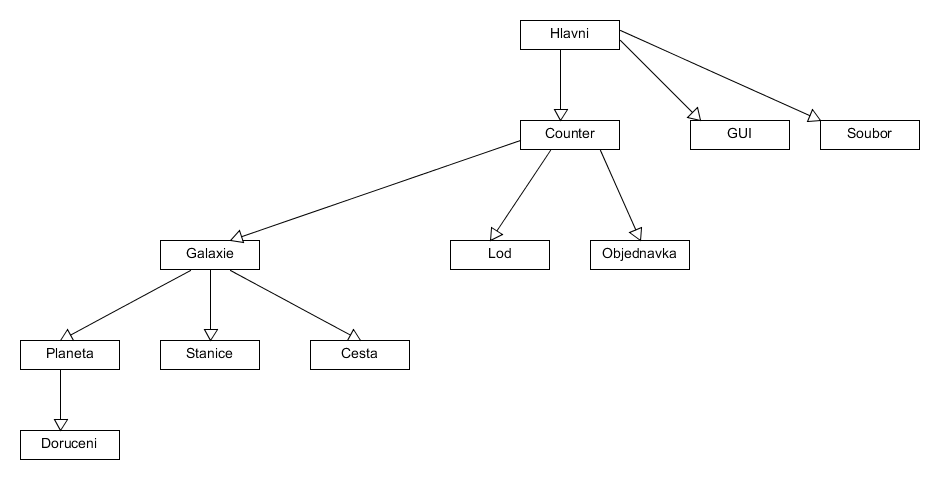
• Pokud společnost nestíhá doručit objednané léky do konce předcházejícího měsíce, tak je vůbec neodesílá (tj. objednávka na planetu nedorazí v půlce měsíce, během kterého byla potřeba).

Analýza problému

Ze zadání vyplývá, že budeme řešit grafový problém, konkrétně problém nejkratších cest. K řešení tohoto problému využijeme Dijkstrův algoritmus. Napadlo nás, že nejkratší cesta nemusí být vždy ta nejbezpečnější . Uvědomili jsme si, že kvůli podobným velikostem cest, bude efektivnější cestovat přes nejmenší počet cest spíše než přes cestu nejkratší. Minimalizujeme tak možnost přepadení a zvýšíme efektivitu dodávání léků. Simulace poměrně velká a dost náhodná, proto se budou konečné výsledky špatně porovnávat a to nám znepříjemní budoucí opravy. Začali jsme uvažovat jak budeme moci konečné rozesílání zefektivnit. Nejprve jsme se chtěli vyhýbat pirátským cestám úplně. Pokaždé když jí objevíme bychom jí naprosto z grafu odebrali a provedli Dijkstrův algoritmus znovu. Vzhledem k tomu, že ale zjistíme jestli je cesta nebezpečná, až potom co nás okradou a šance na přepadení je velice malá, jsme se rozhodli tuto vlastnost vynechat. Navíc by byla velice nepředvídatelná a mohl by nám například vzniknout nespojitý graf. Rozhodli jsme se tedy alespoň využívat na rozesílání nejmenší možný počet lodí. To docílíme tak, že při nakládání lodi zjistíme jestli můžeme po cestě na danou planetu obsloužit požadavky nějakých jiných planet, pokud ano, loď dostane větší náklad a obletí jednou cestou klidně i desítky planet.

Návrh programu

Popis struktur

Všechny datové strutkury jsou vytvořeny ve třídě Galaxie, ta využívá přepravky jako je třída Planeta, Stanice a Cesta, aby reprezentovala celý vesmír. Planeta má v sobě souřadnice, seznam všech blízkých sousedů a hlavně také uloženou cestu jak se k ní dostat z nejbližší Stanice. Třída Cesta je poměrně zbytečná, používáme jí především k určení jestli je přechod mezi planetami bezpečný a nebo ne. Třída Stanice dědí Planetu a funguje velice podobně, až na to že v sobě má dok, ve kterém jsou v průběhu simulace uskladněny přebytečné lodě. Celou tuto strukturu zobrazuje třída GUI. Je navržena podle návrhového vzoru jedináček a stará se o hlavní uživatelské rozhraní. Využívá k tomu 1JavaFX a veškeré vykreslování probíhá v hlavním vlákně programu. Nejsložitější třída je pravděpodobně Counter. Tato třída se stará o průběh celé simulace. Je to taková časová osa. Právě v ní se simulují jednotlivé dny a měsíce. Nechtěli jsme aby nám výpočty zastavovali celé uživatelské rozhraní a tak jsme použili 2 vlákna. První hlavní vlákno využívá již dříve zmíněné GUI a druhé právě tato třída Counter. Tato třída využívá přepravky Loď a Objednávka a každý den je podle potřeby upravuje. Na začátku měsíce zjistí potřebné objednávky, dále každý den, pokud může,

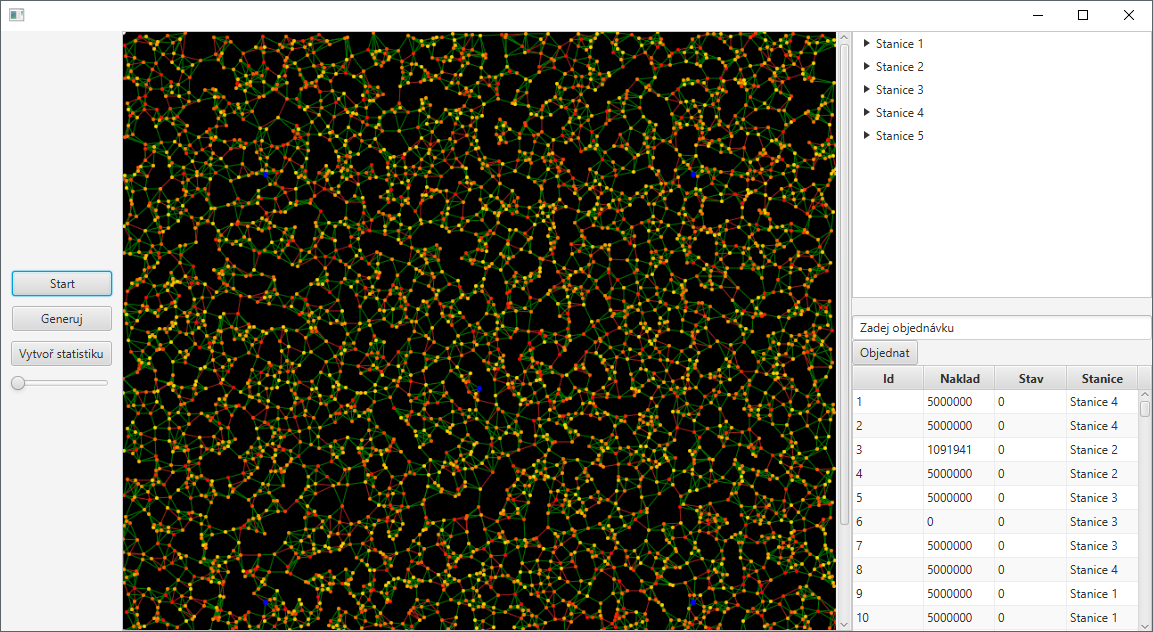
Zjednodušený UML diagram

každou objednávku obslouží nejmenším počtem lodí. Pokud by se náhodou stalo, že lodě v příslušné stanici chybí, loď vytvoří a zapíše do datových struktur. Také se snaží lodě rozesílat co nejefektivněji, to znamená, že pokud má loď volné místo a po cestě by letěla přes neobslouženou planetu, zbytek místa na lodi se zaplní jinou objednávkou. Třída Counter po vyslání lodí vypočítá posun všech současně letících a případně rozhodne zda byli některé okradeny. Lodě které doletěli na cílové planety, jsou nastaveny do vykládacího stavu a příští den letí buď dál a nebo se vrací do stanice. Na konci každého měsíce tato samá třída rozhodne, kolik obyvatel zemře a rozešle příkazy jednotlivým planetám. Dále náš program využívá třídu Soubor, v té se nachází jakákoliv práce se soubory, ať už je to načítání nebo ukládání. Také se v ní vytváří konečná statistika průběhu. Kvůli konečným výpisům jsme museli přidat novou strukturu a tou je třída Doruceni, která v sobě ukládá úspěšně doručené objednávky. Díky tomu můžeme na konci roku vypsat která loď zásobila jakou planetu a ve který čas. Dále se v programu nachází pár Observable a Cell tříd, ty jsou využity na vykreslení pěkných tabulek v pravé části uživatelského rozhraní.

Vlastní modifikace

Při prostudování zadání nás napadlo jak si celou práci značně ulehčit. Rozhodli jsme se většinu výpočtů provést jen jednou a uložit je do souboru. Také nám došlo, že když si stanice efektivně rozmístíme a mezi planetami se budeme pohybovat jen směry od stanice a ke stanici, můžeme si ke každé planetě rovnou poznamenat nejkratší, jedinou, cestu a vyhnout se tak zdlouhavému přepočítávání nejkratších cest každý krok simulace. To nám sice zvětšilo počáteční soubor a prodloužilo počáteční generování vesmíru, značně jsme však ušetřili na časové složitosti. Stačí nám v podstatě jen procházet již vytvořené datové struktury.

Uživatelská dokumentace



*Obr. 1*

C

B

A

Pro spuštění této aplikace nám stačí počítač který má nainstalovanou jakoukoliv Java SE. Pro další vývoj bychom však potřebovali JDK (k sehnání na této adrese http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html) Aplikaci lze spustit bez parametrů z příkazové řádky pomocí příkazu –> java –jar UFO-medical.jar. Po spuštění aplikace nás uvítá hlavní okno (Obr. 1). To je rozděleno do 3 sekcí. Vlevo v sekci A máme možnost spustit simulaci, vygenerovat nový vesmír a vytvořit statistiku z dosavadní simulace a také ovládání přiblížení zobrazení. Uprostřed v sekci B je zobrazen celý vesmír. Modrou barvou je zobrazeno všech 5 stanic. Na škále od žluté po červenou jsou vykresleny všechny planety. Čím více obyvatel tím červenější planeta je. Planety jsou pospojovány cestami, které jsou buď zelené - bezpečné a nebo červené - nebezpečné. Při spuštění simulace si můžeme zobrazit i pozice lodí. Její současná pozice jefialová a pozice na kterou loď letí je růžová. Loď většinou zůstane na cestě a tak jí zobrazíme jako fialovou cestu s růžovým čtvercem na konci, značícím směr cesty. V pravé části rozhraní, v sekci C, je velký strom, reprezentující všechny planety rozdělené pod příslušné stanice. Pod tímto oknem je textové pole do kterého lze zadat velikost objednávky a objednávku odeslat. Dole je poslední prvek a to je tabulka všech lodí. Je zde uvedeno číslo lodě, náklad, její stav a stanice ve které má dok. Při spuštění simulace se tyto hodnoty dynamicky mění. Loď má nejčastější 3 stavy. Má -1 pokud čeká v doku na objednávku, 0 pokud letí z jednoho místa na druhé a 3 když zrovna doletěla a čeká na vyložení.

Závěr

Díky této práci jsme byli přinuceni začít pracovat se sdílenými uložišti jako je Git, což byla z počátku velká výzva. Vyzkoušeli jsme si své první merge a několikrát i neúspěšně. Nakonec jsme ale počáteční obtíže překonali a teď se využití podobných stránek nebojíme. Co se týká simulace překvapili nás nejvíce konečná čísla. Zatím co průměrný počet ukradených léků se drží kolem 20% po průběhu celé roční simulace pořád přežívá 94% populace celého vesmíru což je podle mě velký úspěch. Pokud necháme simulaci běžet dál číslo se sníží po 10 letech asi na 70% a dále již klesá jen velmi pomalu. Zajímavé také bylo pozorovat procenta využití jednotlivých lodí. Většinou je největší potřeba druhý měsíc. Tento jev nastává kvůli tomu, že první měsíc vyletí všechny lodě, ale druhý měsíc začíná když se ještě všechny nestihnou vrátit. To způsobí vytvoření lodí nových. Tyto lodě ale po druhém měsíci skoro nevyužijeme, protože nám klesne celková populace vesmíru a tak nám postupně využití lodí jen klesá.