

Zadání projektu

Navrhněte a implementujte aplikaci pro zobrazení linek hromadné dopravy a sledování jejich pohybu.

Zadání definuje podstatné vlastnosti aplikace, které musí být splněny. Předpokládá se, že detaily řešení si doplní řešitelské týmy. Nejasnosti v zadání řešte, prosím, primárně na k tomu příslušném Fóru.

Specifikace požadavků

- Základní požadavky
 - aplikace zobrazí mapový podklad, na který poté přenáší informace o linkách
 - základní mapový podklad je tvořen pouze liniemi (přímé čáry mezi body křížení)
 - každá linie (čára) reprezentuje jednu ulici (může mít jméno)
 - tento základní koncept můžete jakkoliv rozšířit
 - mapový podklad se načte po spuštění ze souboru (formát si zvolte sami)
 - je možné mapový podklad přibližovat a oddalovat (zoom)
 - systém hromadné dopravy je členěn do linek (např. autobusová linka č. 41)
 - každá linka je definována seznamem zastávek (konečná - průběžné - konečná)
 - zastávka vždy leží na některé z ulic
 - každá linka obsahuje jednotlivé spoje (spoj je jedna kompletní cesta z jedné konečné zastávky do druhé konečné zastávky)
 - linka má jízdní řád, který obsahuje informace o jednotlivých spojích
 - linky a jejich jízdní řády se načítají ze souboru (formát je na vašem uvážení, lze využít některý z dostupných formátů, např. [GTFS](https://developers.google.com/transit), <https://developers.google.com/transit>)
- Pohyb vozidel (spojů)
 - systém obsahuje vlastní hodiny, které lze nastavit na výchozí hodnotu a různou rychlost
 - po načtení mapy a linek začne systém zobrazovat jednotlivé spoje, které jsou právě na cestě (způsob zobrazení je na vaší invenci, postačí značka, kolečko, ...)
 - symbol spoje se postupně posunuje podle aktuálního času a jízdního řádu (aktualizace zobrazení může být např. každých N sekund); pohyb spoje na trase je tedy simulován
 - aktuální polohu na mapě postačí dopočítat podle délky trasy mezi zastávkami, jízdního řádu a vnitřních hodin aplikace; v tomto režimu tedy spoje nemají zpoždění
 - po najetí/kliknutí na symbol spoje se zvýrazní trasa v mapě a zobrazí itinerář spoje (např. ve spodní části čára se zastávkami, časy odjezdů ze zastávek a aktuální pozice spoje)

- Interaktivní zásahy
 - možnost definovat ztížené dopravní situace (stupně provozu)
 - stupeň provozu se týká celé ulice rovnoměrně
 - čím vyšší stupeň, tím pomalejší průjezd
 - vlivem vyšších stupňů provozu dochází ke zpoždění spojů
 - možnost uzavření ulice a definování objízdné trasy
 - objízdná trasa se vytvoří manuálně (např. postupným naklikáním cesty)
 - pro celou objízdnou trasu se nastaví pevná hodnota zpoždění
 - objízdá trasa může vynechat některou ze zastávek
- Inspirace
 - <https://mapa.idsjmk.cz/>
 - <https://tracker.geops.ch>
 - <https://www.miamigov.com/files/sharedassets/public/trolley/brickell-trolley-map.pdf>
- Rozšíření pro tříčlenný tým
 - Vytvořte modul na sběr aktuálních (skutečných) časů jednotlivých spojů z volně dostupných dat (např. <https://www.miamidade.gov/transit/WebServices/MoverTrains/>). Tento modul napojte na vytvořený systém -- simulovaný čas se změní na čas reálný, zobrazená poloha by měla co nejvíce odpovídat skutečnému stavu.

Součást odevzdání

- Připravte mapový podklad obsahující alespoň 20 ulic.
- Připravte předem alespoň 3 linky, každá má alespoň 10 spojů.

Další požadavky

- Zadání definuje podstatné vlastnosti aplikace, které musí být splněny. Předpokládá se, že detaily řešení si doplní řešitelské týmy.
- Při návrhu postupujte tak, aby výsledný program byl dobře použitelný. Hodnotit se bude nejen to, zda splňuje výše uvedené požadavky, ale také JAK je splňuje.
- Návrh musí oddělit vlastní model aplikace od GUI (navrhněte vhodné rozhraní).
- Návrh programu zdokumentujte ve zdrojovém kódu ve formátu pro program "doxygen".
- Používejte vhodné návrhové vzory (Design Patterns) - viz literatura.

- Pro vytváření menu, dialogů a dalších komponent grafického uživatelského rozhraní (GUI) použijte toolkit Qt minimálně ve verzi 5.5 (nainstalováno v učebnách a na serveru "merlin" v adresáři </usr/local/share/Qt-5.5.1/>). Pokud pro vývoj použijete verzi novější/starší než je na učebnách, předpokládáme, že zajistíte zpětnou/dopřednou kompatibilitu vašich programů. (V případě problémů, počítejte s nutností předvedení překladu/sestavení/běhu aplikace na vašem notebooku. Toto se nedoporučuje.)
Protože toolkity jsou přenositelné na více platform, měly by vaše programy fungovat minimálně na Linuxu i na Windows. (Nejen Java umožňuje psaní přenositelných aplikací.) Je jedno na které platformě proběhne vývoj aplikací, jen dbejte na pravidla psaní přenositelných programů. (Testovat se bude na Linuxu = server merlin, ale Makefile nesmí být závislý na konkrétním adresáři s lokální instalací toolkitu.)
Na vytvoření GUI dialogů a menu můžete použít interaktivní editory/generátory.
- Použijte C++17. (Boost raději nepoužívejte. Pokud použijete některé knihovny z Boost, tak použité soubory přibalte a zkontrolujte si, zda se to vše vejde do max. velikosti odevzdaného archivu. Existuje nástroj BCP pro výběr podmnožiny Boost.).
- Programy budou vytvořeny zadáním příkazu "make" v adresáři s rozbalenými zdrojovými kódy projektu (Pozor - bude se automaticky překládat). Musíte vytvořit soubor Makefile, který nesmí obsahovat absolutní cesty k souborům. Výsledný program se bude jmenovat "icp". Pro testovací účely půjde vše spustit použitím "make run".
- Součástí projektu bude programová dokumentace vytvořená programem Doxygen (konfigurační soubor nastavte na HTML výstup se zahrnutím zdrojových textů: SOURCE_BROWSER = YES). Dokumentace se vygeneruje automaticky (= nezabalovat do archivu) při zadání "make doxygen" a bude uložena v podadresáři doc.

Poznámky k odevzdávání

- Termín odevzdání je uveden v IS.
- Řešení zabalené ve formátu .zip (nebo .tar.gz) odevzdá pouze vedoucí týmu do IS a to s dostatečným předstihem před uzávěrkou. (Nikdo nebude psát, že neodevzdal, protože *** a posílat to poštou.)
Jméno odevzdaného souboru musí mít tvar: **xname01-xname02.zip**, kde xname01 je jméno vedoucího.
- POZOR: Archiv může mít **maximálně 2MB**, aby mohl být vložen do IS.
- Odevzdaný archiv obsahuje minimálně tuto adresářovou strukturu:
./src/* - zdrojové texty
./examples/* - příklady (pro testování)
./doc/ - dokumentace (bude později generován: "make doxygen")
./README.txt - základní přehled o co jde, autoři, poznámky, ...
./Makefile - zavolá rekurzivně make na src/Makefile, případně na generování dokumentace atd.
- Neodevzdávejte nic, co lze vytvořit pomocí "make" (spustitelné programy ani dokumentaci, kterou lze vygenerovat použitím "make doxygen").
Poznámka: Příkaz "make clean" všechny produkty překladu smaže, "make pack" vytvoří archiv pro odevzdání - zkontrolujte, zda obsahuje vše potřebné rozbalením v prázdném adresáři a překladem/spuštěním.

- Všechny zdrojové soubory musí obsahovat na začátku komentář se jmény autorů daného modulu, popisem obsahu modulu, atd. Z komentářů by také mělo být patrné, které části byly napsány ručně a které vygenerovány automaticky.

Poznámky k hodnocení

- Projekt je hodnocen na stupnici 0-100 procent/bodů.
- Po ohodnocení může dojít k přepočtu bodů na jednotlivé členy týmu podle návrhu vedoucího týmu, který je zakódován do jména odevzdaného souboru (formát jména pak je: **xname01-xname02-AA-BB.zip**
Součet $AA+BB=100$, POZOR: pro rovnoměrné rozdělení $AA=BB=50\%$ není třeba uvádět "-50-50" a je to preferovaná možnost).
- Hodnocení zahrnuje kvalitu OO návrhu, kvalitu implementace v C++, dodržování stylu psaní programu (odsazování, kvalita komentářů, vhodné identifikátory) a především funkčnost programu. Pokud budou ve zdrojových textech chybět podstatné komentáře (zvláště jména autorů), bude projekt hodnocen 0b.
- POZOR:
Pokud nepůjde přeložit příkazem `make` = 0b nebo předvedete na svém notebooku a vysvětlíte kde byl problém,
pokud nebudou komentáře se jmény autorů = 0b.
Zcela nevhodně formátovaný a nečitelný kód = 0b.
Plagiáty atd. = 0b + návštěva u disciplinární komise.