

Projekt

Základním principem projektu bude udělat grafické RPG/dungeon, které bude klasického stříhu s uživatelem pohybujícím se po čtvercovém poli (viz obrázek např. ze hry Gargoyles Quest). Mapa má omezenou velikost např. 50x50 polí. Ve hře je několik map, mezi kterými může uživatel přecházet. Mapa je vykreslována pomocí ASCII znaků na terminál.

Uživatel může sbírat předměty a bude plnit úkoly, které se dozví od NPC se kterými může interagovat. Tato interakce bude velmi jednoduchá. Bude to pevný text, který NPC „řekne“ hráči a v ten moment se do jeho seznamu úkolů přidá úkol. Ten může být typu najdi určitý předmět nebo zabij konkrétní příšeru. Po zabití příšery dostane hráč nějaký artefakt (např. lebku). Pokud s nalezeným předmětem nebo artefaktem dojde za NPC, dostane od ní odměnu a body do celkového výsledku hry.



Můžete libovolně navýšit funkcionalitu o další obvyklé prvky RPG/dungeonů. Uživatelské rozhraní může být různé. Určitě bude třeba vypisovat různá menu s možnostmi, co lze dělat, co mám v inventáři atp. Samotný průchod mapou může být řešen různě. Základní myšlenka je, že budeme vidět náhled aktuální mapy vykreslený pomocí ASCII.

Struktura aplikace

Strukturu aplikace lze rozdělit na dvě části: “frontend”, to budou třídy, které se budou starat o zobrazení plánu hry, menu, vypsání inventáře hrdiny atp. Druhou částí bude aplikační logika, která bude poskytovat informace frontendu, který bude prakticky dost “hloupý”. Požádá např. Aplikační logiku “vrať mi popis věcí v inventáři” a aplikační logika musí jasně odpovědět vrátit např. Seznam textů”.

Doporučená rozšíření pro více lidí v týmu

- ❑ Celou hru lze uložit a načíst poslední uložený stav. Samozřejmě ukládání neprobíhá do souboru, ale pouze do operační paměti počítače. Není tedy možný load mezi jednotlivými spuštěními aplikace. Jen v rámci jednoho běhu.

Na co si dát pozor

- ❑ Nejde o kvantitu, ale o kvalitu! Tj. Podstatou projektu je objektový návrh. Design hry, rozsáhlost atp. Je také hodnocena, ale klíčový je objektový návrh. Takže soustřeďte se hlavně na to, abyste věci správně ukládali, dobře rozčlenili funkcionalitu (a neměli jednu třídu, která řeší celou aplikační logiku) atp.
- ❑ Opravdu extrémně častou situací je, že někdo udělá většinu kódu do jedné třídy, někdy i jedné metody, která má pak délku stovky řádků. Toto není akceptovatelné. Délka metody jsou jednotky řádků, maximálně nějaké nižší desítky.
- ❑ Doporučuji důsledně oddělovat třídy, které budou komunikovat s uživatelem a které se budou starat o data a aplikační logiku. Samozřejmě můžete si udělat např. třídu Map, která bude primárně ukládat informace o mapě, ale bude obsahovat metodu "printMapTile(...)" vypisující třeba aktuální mapové políčko. Ale tato třída nebude řešit kdy ho má vypisovat, je bude pasivně čekat až jí o to požádá třída řeší herní smyčku.
- ❑ Ostatní věci viz přednáška o velkých projektech.

Hodnocení a obhajoby

První obhajoba (zápočtový týden, 10 bodů)

- ❑ Tato obhajoba je povinná, pokud chcete projekt řešit. Nelze ji přeskočit.
- ❑ Musí vám fungovat hlavní smyčka aplikace ve které komunikujete s uživatelem (vypsání menu, reakce na vstup od uživatele...)
- ❑ Musí vám fungovat základní ukládání mapy do dynamických datových struktur + možnost pohybu po mapě.
- ❑ Prezentovat budete diagram tříd popisující aktuální stav projektu + plán, jak doděláte zbytek (na konci se dozvíte hodně věcí o návrhových vzorech, jejichž využití jen naplánujete)
- ❑ Dále budete prezentovat funkční kód, který si cvičící zahraje.
- ❑ Obhajoba není konzultace. Neřešte na ní, jak máte věci udělat a neměly by na ní padat věty typu "my jsme nevěděli jak...". Silně doporučuji vývoj hry průběžně konzultovat, abyste se vyhnuli různým nepochopením a problémům. Je ideální konzultovat vždy svého cvičícího u kterého budete projekt obhajovat. Názory na různá řešení se mohou lišit.

Druhá obhajoba (zkouškové období, 15 bodů)

- ❑ Musí vám fungovat kompletní hra a musí být rozdělena do logických bloků.
- ❑ O generování map se stará samostatná množina tříd, o herní logiku další, o komunikaci další. Většina těchto problémů je řešena pomocí návrhových vzorů, které se budeme učit. Vhodné použití návrhových vzorů je jeden z klíčových aspektů hodnocení.
- ❑ Hodnotíme také různé extra funkce, které daný tým implementoval (klidně za bonusové body)

Obě obhajoby

- Hodnotíme, jak tým spolupracoval prostřednictvím gitu. Očekává se, že řešení projektu zabere řádově vysoké desítky až stovky commitů. Dále musí být z gitu patrné kdo kód programoval. Body budeme rozdělovat podle práce prezentované v gitu. Někdy můžete aplikovat párové programování, ale není to standardní řešení celého projektu.