# Programátorská dokumentace

Doporučuji otevřít projekt ve Visual Studio 2015 – je zde vytvořená složková hierarchie pro třídy. Celý koncept aplikace je postaven na tom, že máme manažera – TanksAppManager, který se stará o to, co se v aplikaci a kde děje. Pro grafickou část aplikace je použita knihovna SFML 2.0

Aplikace má různé „Screens“ - pro každou obrazovku ve hře jednu, TanksAppManager ví, jaká obrazovka se má konkrétně zobrazovat a na té potom volá metody Render() pro vykreslení a HandleEvent() pro zpracování nějaké SFML události.

Nyní některé třídy, které v aplikaci jsou a co mají za úkol:

TanksAppManager

- Inicializace všeho
- hlavní organizátor aplikace, stará se o to, která screen to má právě pod palcem (AppStatusEnum)
- Ukládání a načítání uložených her
- Má v sobě uloženou strukturu pro data všech hráčů ve hře a metodu pro přístup k nim
- Přístup k potřebným věcem ze SFML

GameManager

- Stará se o Game = běžící hra s tanky <=> AppStatus = game
- Uchovává GameComponents – z důvodu aby jednotlivé GameComponents mohly mezi sebou různě interagovat a měly k sobě přístup
- řídí Game samotnou – v jakém statusu právě je (hraje hráč a jaký, je vystřeleno), monituruje výhry a prohry

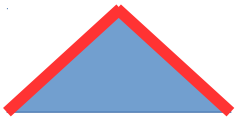
Components/Others, Components/Buttons

- komponenty = něco co se zobrazuje na nějakém screenu a může to i nějak interagovat s uživatelem
- při vytváření nové - pokud podědíme z BaseComponent, měli bychom být dotlačení k tomu, ať komponenta funguje správně

Components/Game

- speciální případ komponent pro screen Game – tedy už běžící hra s tanky
- jsou zde všechny věci pro Game
- oproti normálním komponentám mají přístup ke GameManageru
- Obsahuje 2 pouze datové struktury – TankProperties a Player = obaly pro uchovávání dat o těchto entitách

- pro vytvoření nové GameComponenty by mělo stačit naimplementovat virtuální třídu GameBaseComponent a dopsat si potřebné metody – popis:
    - Render = co se děje v případě, kdy se vykresluje, pro vykreslování okno SFML získáme app_mngr.GetWindow()
    - ProcessEvent() = zpracování nějaké sfmlEvent
    - Type() = mělo by vracet název komponenty – díky tomuto pokud dostaneme obecnou GameComponent víme, která konkrétně to je
    - CollideWith() = co se stane, pokud se mnou něco zkoliduje – co je v parametru. Sám můžu tuto metodu volat na jakékoliv komponentně, kde já jsem ten, kdo koliduje.
    - GetTopCollisionLine = pro všechny x-ové souřadnice vracím horní y-ovou souřadnici, kde už jsem já – viz obr
    - chci určitě zavolat konstruktor předka



Screens

- Jednotlivé screeny, pokud chceme vytvořit novou, naimplementujeme GameScreenable nebo její specifičtější verzi BasicStaticScreen (tu pokud jen chci říct, jaké komponenty obrazovka má a o nic víc se nestarat)
- přidáme je do gameManageru tím, že v TanksAppManager.cpp přidáme do mapy pro nový status hry instanci naší screen

# Uživatelská dokumentace

Ovládání:

- šipky vlevo/vpravo: pohyb vlevo/vpravo
- šipky nahoru/dolů: změna míření kanonu
- pgUp/pgDwn: změna síly výstřelu
- home/end: změna zbraní

Hráč má na začátku 10000. Za každou výhru dostane dalších 10000. Za tyto peníze si hráč může koupit benzín, náboje a jiné – viz Tank Settings.

Pokud hráč nemá náboje zvoleného druhu, zmáčknutím odpalu (mezerník) se dostane na tah druhý hráč.

Hráč, který umře, další kolo začíná na své startovní poloze, hráč, který zabil zůstane tam, kde byl.

Všem hráčům se každé kolo obnoví zdraví do původní polohy.

Hra nemá jako taková konec, no hráč, který už nemá peníze ani náboje prohrál.

Uložení hry v Tank Settings – vybereme si nějaký slot. Pokud je již obsazený, přepíšeme jej aktuální hrou.

Načtení v hlavní obrazovce.