

Západočeská univerzita v Plzni
Fakulta aplikovaných věd
Katedra informatiky a výpočetní techniky

ZPG:Semestrální práce

1 Zadání

1.1 Povinná část

Situace Aplikace umožňuje procházet terénem na vymyšlené planetě.

Implementace

1. Implementace bude provedena v prostředí .NET s využitím Managed Direct3D.
2. Pro implementaci může být použito rozhraní D3DUT (nutno rozhodnout při prvním odevzdání, není možno se později vrátit do Managed Direct3D). Za implementaci v D3DUT ve Windows je možno získat až 5 bodů, v OS Linux až 10 bodů.
3. povinná část - nutná podmínka udělení zápočtu terén zadaný polem alespoň 128x128 výšek vykreslený pomocí trojúhelníkové sítě
4. výšková mapa bude načtena z binárního souboru, jedna výška je reprezentovaná jedním bytem, výšky jsou uloženy v řádcích od západu k východu, řádky jsou uloženy od severu k jihu

Interaktivní průchod terénem

1. pozorovatel chodí po terénu, výška očí pozorovatele viz dále
2. výšku terénu mezi body zadanými výškovou mapou je třeba interpolovat
3. pokud pozorovatel dorazí na konec výškové mapy, nemůže jít dále
4. Za žádných okolností nesmí pozorovatel vidět nebo vstoupit "dovnitř" terénu
5. pozorovatel se pohybuje rovnoměrnou rychlostí vzhledem ke skutečnému času (viz níže) i pozorovatel může pomocí klávesnice chodit vpřed, vzad a dělat úkroky doprava a doleva (nezaměňovat s pohybem na sever, jih, západ a východ!)
6. jednotlivé pohyby vpřed a vzad lze libovolně kombinovat s úkroky do stran, vzniká tak vektor pohybu, který určuje směr, nikoliv rychlost, tzn. hráč se stále pohybuje konstantní rychlostí.
7. pozorovatel se může pomocí myši otáčet, rychlost otáčení je závislá pouze na rychlosti pohybu myši
8. při chůzi dopředu/dopředu/vpravo/vlevo je nutné brát v úvahu natočení pozorovatele vpravo/vlevo
9. natočení pohledu pozorovatele nahoru/dolů nemá na směr chůze vliv
10. při rozhlížení nahoru/dolů se pozorovatel nemůže přetočit, tj. natáčení ve vertikálním směru lze pouze v rozsahu (-90; +90) stupňů, kde 0 je přímo vpřed
11. výchozí pozice pozorovatele je uprostřed mapy
12. veškeré animace a pohyb pozorovatele jsou závislé na reálném čase, nikoliv na snímkovém kmitočtu, tj. změna doby nutné na zobrazení jednoho snímku nesmí způsobit změnu rychlosti pohybu či animace v zobrazovaném světě

13. scéna je osvětlena sluncem

- (a) naše planeta velmi rychle rotuje, jeden astronomický den trvá dvě minuty, přičemž "světlo" je jednu minutu, "tma" také jednu minutu
- (b) během dne se mění intenzita slunečního svitu, za úsvitu a za soumraku je nejmenší, v poledne největší
- (c) během noci je terén osvětlený tak, aby bylo něco vidět
- (d) přechod mezi dnem a nocí musí být plynulý
- (e) zobrazení počítadla snímků za vteřinu (FPS)
- (f) výpočet počtu snímků za předchozí vteřinu počínaje aktuálním snímkem, tj. počet předchozích snímků kteréžto se vejdou do okénka o velikosti jedné vteřiny
- (g) snímková frekvence nesmí klesnout pod 20 fps
- (h) snímková frekvence je reálné číslo

14. geometrie a údaje : výšková mapa o rozměru 128x128 se zobrazí jako terén 254x254 m, tj. výška terénu je vzorkována po 2 metrech

15. hodnota 0 ve výškové mapě znamená nadmořskou výšku 0, hodnota 255 nadmořskou výšku 25,5 m

16. Pozorovatel

- (a) výška očí: 1,85 m
- (b) hmotnost: 95 kg
- (c) průměrná rychloprůměrná rychlost chůze: 3 m/s

17. Ovládání uze standardní vstupní zařízení

vstup	význam
myš	otáčení vlevo/vpravo, rozhlížení nahoru/dolů (pohyb myši vpřed = nahoru)
klávesa 'W'	pohyb kupředu ve směru pohledu
klávesa 'S'	pohyb vzad vzhledem ke směru pohledu
klávesa 'A'	pohyb vlevo vzhledem ke směru pohledu (nikoliv otáčení)
klávesa 'D'	pohyb vpravo vzhledem ke směru pohledu (nikoliv otáčení)
klávesa 'U'	zapnutí/vypnutí invertace myši pro pohled nahoru/dolů.
ostatní	dle potřeb, ale nutno popsat v dokumentaci

1.2 Volitelná rozšíření

V této sekci jsou citována zadání **zvolených** volitelných rozšíření.

1. ostatní

- (a) konzole (5 b)
Jednoduchá poloprůhledná konzole obvyklá z počítačových her. Minimálně 3 příkazy s dopadem na scénu. Za vzor je považována konzole ze hry Quake III.

2 Řešení

2.1 Povinná část

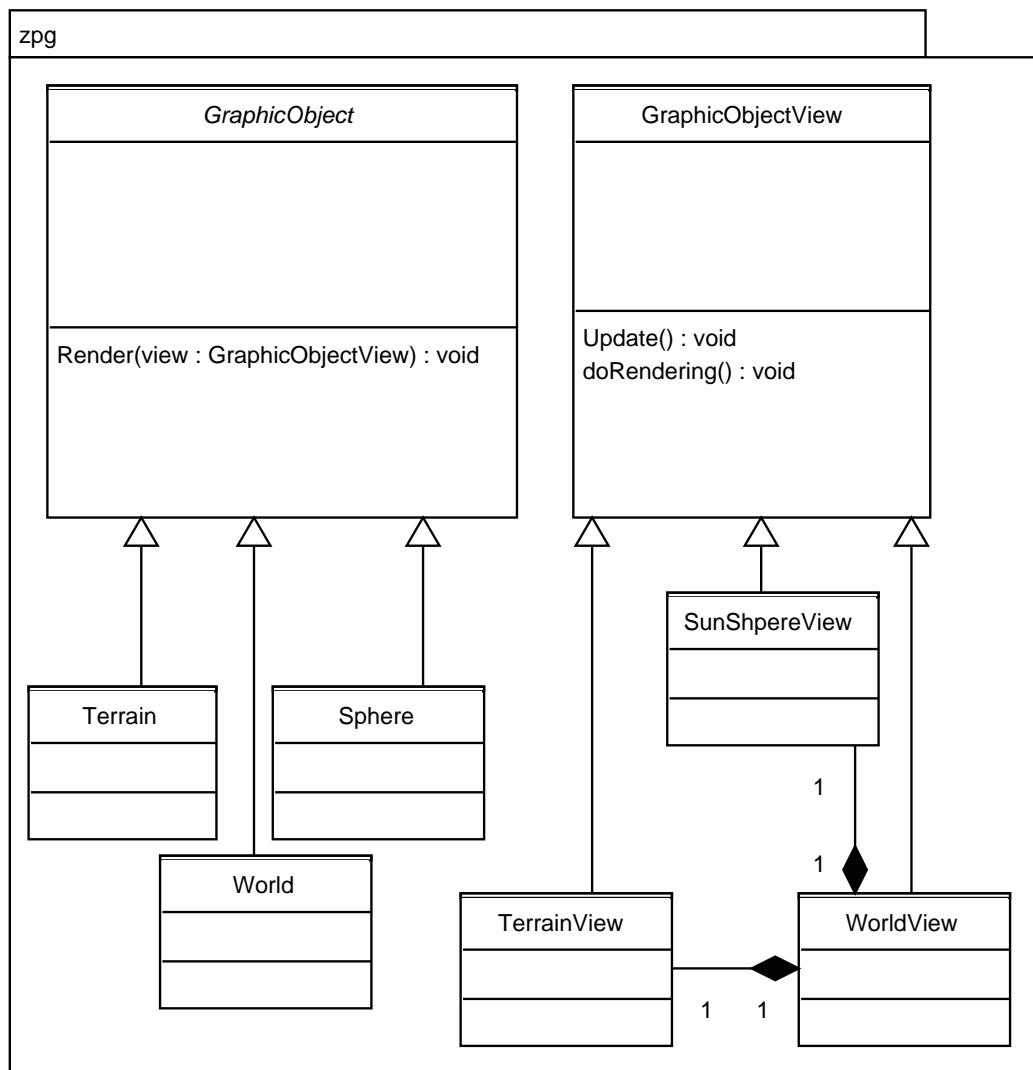
Vypracování povinné části odpovídá požadavkům.

Celkový náhled Program je objektově navržený a odděluje grafické zobrazení od vlastních dat objektu. Vzhledem k omezenému rozsahu práce jsou všechny třídy pouze v jednom jmenném prostoru - *zpg*. Pokud by program měl být dále rozšiřován, bylo by velmi vhodné zavést komplexnější strukturu jmenných prostorů.

Hlavní, spustitelnou třídou je (z historických důvodů) *RenderForm*. Tato třída vznikla úpravami vzorové třídy *zpg.start.RenderForm*. Obsahuje metody pro zpracování vstupů a hlavní *update-render* cyklus "hry". Bylo by vhodné zavést do hlavního cyklu ještě část *wait*, což by umožnilo držet předem definovanou snímkovou frekvenci.

Datové struktury Datová struktura grafických objektů (viz obr. 1), je snad dostatečně pružná libovolný případ užití.

Obrázek 1: Model tříd - GraphicObject

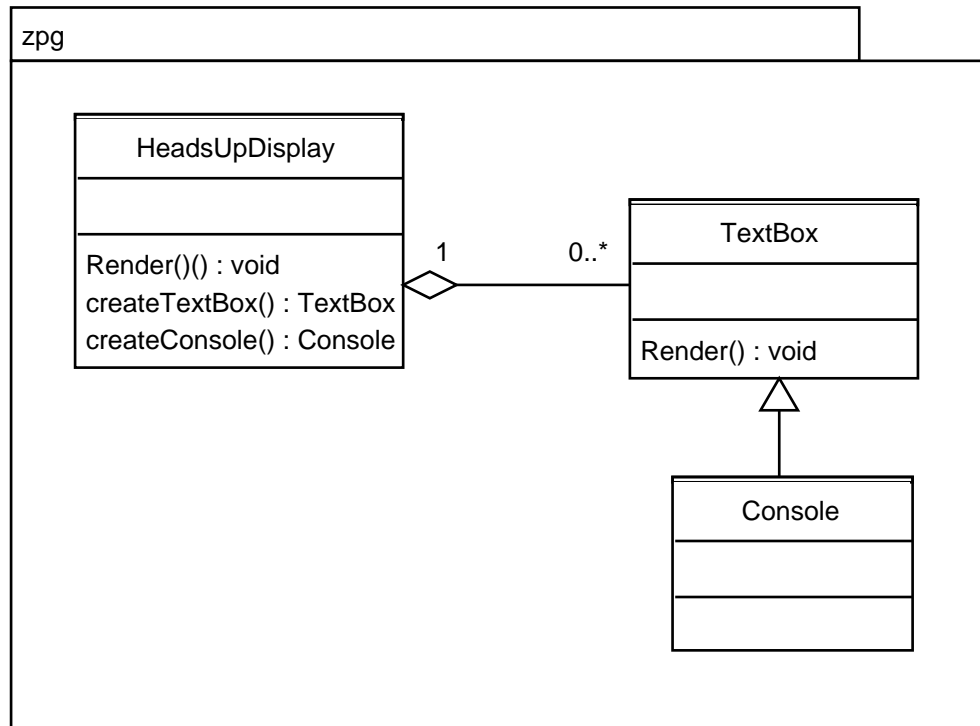


Základem je *GraphicObject* - udržuje grafický vzor - archetyp objektu(např. koule). S *GraphicObject* je těsně svázána třída *GraphicObjectView*, která obsahuje

již jen odkaz na patřičný *GraphicObject* a parametry jeho zobrazení. Tak například objekty *Slunce* a *Míč* by byly reprezentovány jedinou instancí *GraphicObject* - *Koule*, ale pro každý by vznikla instance *GraphicObjectView*. *GraphicObjectView* také obsahuje chování objektu - metodu *Update()*.

Heads Up Display(HUD) je zapouzdřen do třídy *HeadsUpDisplay*(viz obr. 2).

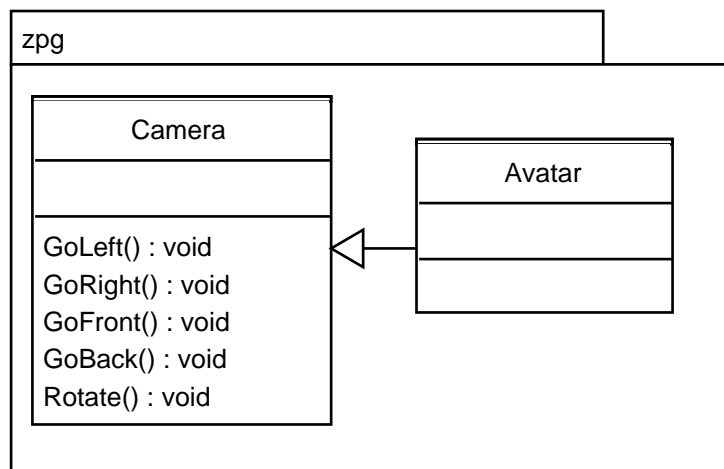
Obrázek 2: Model tříd - HeadsUpDisplay



Jedná se o jednoduchý kontejner Objektů typu *Textbox*. *Textbox* umožňuje zobrazení textových informací do HUD. Zděděním *TextBoxu* vzniká *Console* která již má průhledné pozadí a uživatelský vstup.

Pohyb kamery po terénu zajišťuje třída *Camera*(viz obr. 3)

Obrázek 3: Model tříd - Camera

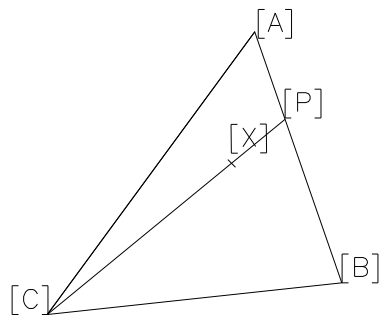


kteřá drží instance od zobrazovacích matic. Jejím zděděním vzniká *Avatar* - kamera navázaná na povrch terénu.

Algoritmy Použité algoritmy jsou vesměs velice jednoduché - většinu "těžké práce" obstará grafická knihovna. Ale přesto :

1. Výpočet normál vertexů pro terén(který by také udělal DirectX, kdyby terén byl mesh object). Algoritmus je přímou implementací vzorce $\sum_{i=1}^n v_i \times v_{i+1}$. Kde n je počet sousedních vertexů a v_i je polohový vektor i-tého vertexu vůči "centrálnímu". n je v obecném případě 8 a je třeba ošetřit okrajové a rohové vertexy.
2. Zjišťování výšky terénu v daném bodě - nutné pro *Avatara* chodícího po terénu. Tento výpočet je realizován dvoupolohovou lineární interpolací v trojúhelníku: Výška v bodě P je poměrnou výškou bodů B a A . Hledaná výška v bodě X

Obrázek 4: Výpočet - Výška terénu



pak poměrnou výškou bodů C a P .

Reference

[Pierson] <http://blogs.msdn.com/coding4fun/archive/2006/11/09/1044115.aspx>

[neXe] <http://nexe.gamedev.net/directKnowledge/>

[riemers] <http://www.riemers.net/eng/Tutorials/DirectX/Csharp/Series1/tut4.php>