

3D Vizualizácia mapy

Projekt na predmet Tvorba informačných systémov
Návrh

Vedúci projektu: • Peter Náther

Členovia vývojárskeho tímu: • Timotej Jurášek
• Martin Miklis
• Jakub Motýľ

Verzia dokumentu 1.2

Obsah

1. Špecifikácia požiadaviek

1.1. Úvod

1.1.1. Predmet špecifikácie

1.1.2. Slovník pojmov, skratky

1.1.3. Odkazy

1.2. Cieľová skupina používateľov

1.3. Dostupnosť

1.4. Vstupy a výstupy

1.4.1. Vstupy aplikácie

1.4.2. Výstupy aplikácie

1.5. Funkcie

1.5.1. Kontextový diagram

1.5.2. Detailný popis

- 1.5.2.1. Načítanie údajov
- 1.5.2.2. Zobrazenie 3D povrchu
- 1.5.2.3. Nastavenie detailu renderu
- 1.5.2.4. Nastavenie výškového rozdielu vrstevníc
- 1.5.2.5. Rotácia modelu
- 1.5.2.6. Približovanie modelu
- 1.5.2.7. Posúvanie modelu
- 1.5.2.8. Zobrazenie 3D trás
- 1.5.2.9. Prevýšenie
- 1.5.2.10. Priestorová dĺžka
- 1.5.2.11. Energetická náročnosť

1.6. Škálovanie a hardvérová náročnosť

1.7. Možné rozšírenia funkcionality

2. Konceptuálna analýza

- 2.1. Používatelia systému
- 2.2. Use case diagram
- 2.3. Cieľ používateľa
- 2.4. Entitno-relačný diagram
- 2.5. Návrh používateľského rozhrania

3. Analýza technológií, dekompozícia a dátový model

- 3.1. Analýza technológií
- 3.2. Model vstupného súboru Omap
- 3.3. Dátový model
- 3.4. Podrobná špecifikácia komponentov
- 3.5. Testovacie scenáre

1. Špecifikácia požiadaviek

1.1. Úvod

1.1.1. Predmet špecifikácie

Táto špecifikácia požiadaviek na softvér (ďalej ŠPS) popisuje používateľské, funkčné a parametrické požiadavky prvej verzie systému pre zobrazovanie a prácu s 3D modelom mapy. ŠPS je určená pre tím, ktorý bude výsledný softvér implementovať. Špecifikácia je súčasťou zmluvy medzi objednávatelom a dodávateľom. Bude slúžiť ako východisko pre vyhodnocovanie správnosti softvéru.

1.1.2. Slovník pojmov, skratky

OOM	OpenOrienteering Mapper
Základná verzia aplikácie	Verzia implementujúca len najdôležitejšie súčasti aplikácie, popísané a označené v tomto dokumente
Rozšírená verzia aplikácie	Nadstavba nad základnou verzou aplikácie, pripravená v prípade ideálneho splnenia časového plánu a implementujúca rozširujúce funkcionality popísané a označené na konci tohto dokumentu

1.1.3. Odkazy

OpenOrienteering Mapper	https://openorienteering.github.io/
Verejný repozitár projektu	https://github.com/TIS-BoardSmashers/3DMapVisualization
Záznamy komunikácie vývojárskeho tímu s klientom	https://github.com/TIS-BoardSmashers/3DMapVisualization/tree/master/Communication
xmap	Dokumentácia tohoto formátu nie je k dispozícii. Ide o formát založený na XML. Viac v komunikácii s tvorcami OOM na

	https://github.com/TIS-BoardSmashers/3DMapVisualization/blob/master/Communication/mail02.txt
gpx	http://www.topografix.com/gpx/1/1/

1.2. Cieľová skupina používateľov

Cieľové skupiny používateľov pre 3D vizualizáciu mapy budú hlavne učitelia zemepisu predovšetkým na stredných školách, technologicky zdatní nadšenci orientačného behu, turistiky, prípadne ktokoľvek, kto chce využiť túto vizualizačnú pomôcku.

1.3. Dostupnosť

Aplikácia bude dostupná primárne pre operačný systém Windows ako offline desktopová aplikácia.

1.4. Vstupy a výstupy

1.4.1. Vstupy aplikácie

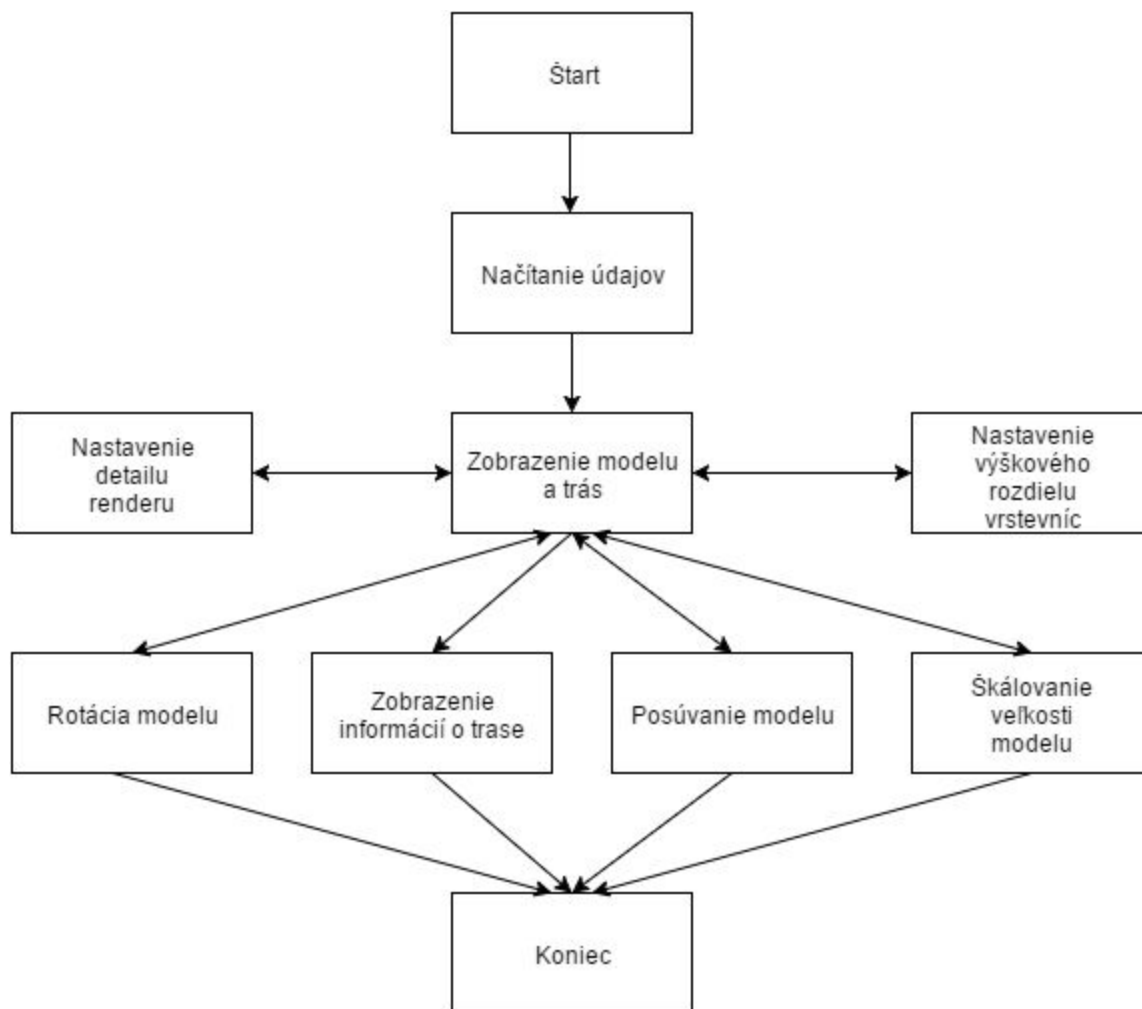
Vstup do aplikácie je výstup z OOM, kde sa mapy vytvárajú. Vstupné súbory obsahujúce vrstevnice s ktorými bude aplikácia pracovať sú formátu *xmap*. Vrstevnice sú reprezentované ako polygonálne cesty alebo kubické Bezierové spliny v 2D. Dajú sa čítať priamo z *xmap* formátu, ktorý je založený na XML a teda dá sa parsovať pomocou existujúcich XML parserov. Objekty sú v súbore označené ako *<object>*. Každý objekt má svoje *id*, podľa ktorého sa dá rozlíšiť o aký typ objektu v mape ide. Keďže vrstevnice nemajú priradenú výšku a nedá sa rozlíšiť čo je prepadlina a čo kopec, vrstevnice prepahlín budú označené špeciálnym symbolom. Takisto výškový rozdiel susedných vrstevníc nie je súčasťou súboru, bude zadávaný počas behu v aplikácii.

1.4.2. Výstupy aplikácie

Výstupom aplikácie je interaktívna 3D scéna s modelom terénu a vyznačenými trasami, s ktorou sa dá v aplikácii pracovať, a jej funkcionality je popísaná nižšie v časti *Funkcie*. Žiadne súbory sa z aplikácie neexportujú.

1.5. Funkcie

1.5.1. Kontextový diagram



1.5.2. Detailný popis

1.5.2.1. Načítanie údajov

Aplikácia načítava údaje o vrstevniciach, ich špeciálnych symboloch (prepadnutá vrstevnica/vystúpená vrstevnica) a trasách z jedného vstupného súboru.

1.5.2.2. Zobrazenie 3D povrchu

Aplikácia zobrazuje 3D model povrchu skonštruovaný z trojuholníkových plôch na základe vrstevníc zadaných vstupným súborom.

1.5.2.3. Nastavenie detailu renderu

Užívateľ si v aplikácii môže za behu nastaviť detail renderovania modelu.

1.5.2.4. Nastavenie výškového rozdielu vrstevníc

V aplikácii je možné za behu nastaviť výškový rozdiel medzi susednými vrstevnicami pre realistickejšie zobrazenie, prípadne pohodlnejšiu manipuláciu.

1.5.2.5. Rotácia modelu

Model sa dá rotovať okolo všetkých osí x, y, z.

1.5.2.6. Približovanie modelu

Model sa dá približovať a oddalovať.

1.5.2.7. Posúvanie modelu

S modelom sa dá pohybovať po všetkých osiach x, y, z.

1.5.2.8. Zobrazenie 3D trás

Na modeli sa zobrazujú priestorové trasy z bodu A do bodu B, načítaných zo vstupného súboru.

1.5.2.9. Prevýšenie

Aplikácia pre jednotlivé trasy počíta a zobrazuje celkové prevýšenie trasy.

1.5.2.10. Priestorová dĺžka

Aplikácia pre jednotlivé trasy počíta a zobrazuje ich dĺžku v priestore.

1.5.2.11. Energetická náročnosť

Na základe prevýšenia a priestorovej dĺžky aplikácia počíta a zobrazuje energetickú náročnosť trasy.

1.6. Škálovanie a hardvérová náročnosť

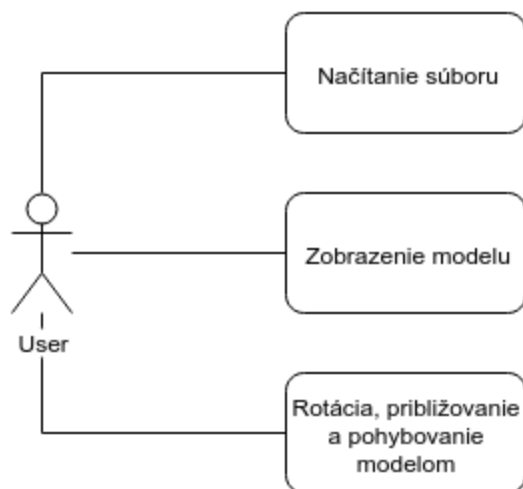
Škálovanie bude možné vďaka možnosti nastavenia kvality renderovania, ktorá priamo ovplyvní počet plôch v 3D vizualizácii. Hardvérová náročnosť bude teda závisieť hlavne od úrovne detailov, no s extrémne rozľahlými mapami sa v základnej verzii nepočíta.

1.7. Možné rozšírenia funkcionality

Za funkcionality, ktorá nie sú súčasťou základnej verzie ale rozširujúce v prípade ideálneho splnenia plánu považujeme funkcie:

- porovnávanie trás na základe priestorovej dĺžky a prevýšenia
- porovnávanie trás na základe energetickej náročnosti
- viacero zobrazených trás súčasne s možnosťou zobrazenia informácií o jednotlivých trasách
- možnosť importu gpx súboru s trasou (trasa nebude súčasťou exportu mapy)
- obmedzenie manipulácie modelom v zmysle zakázania stavu kedy sa kamera nachádza vo vnútri modelu
- vykresľovanie typu terénu na 3D modeli (les, lúka, kamene,...)
- vyhľadávanie najkratšej priestorovej cesty medzi dvoma bodmi v 3D modeli

2. Konceptuálna anal



2.1. Používatelia systému

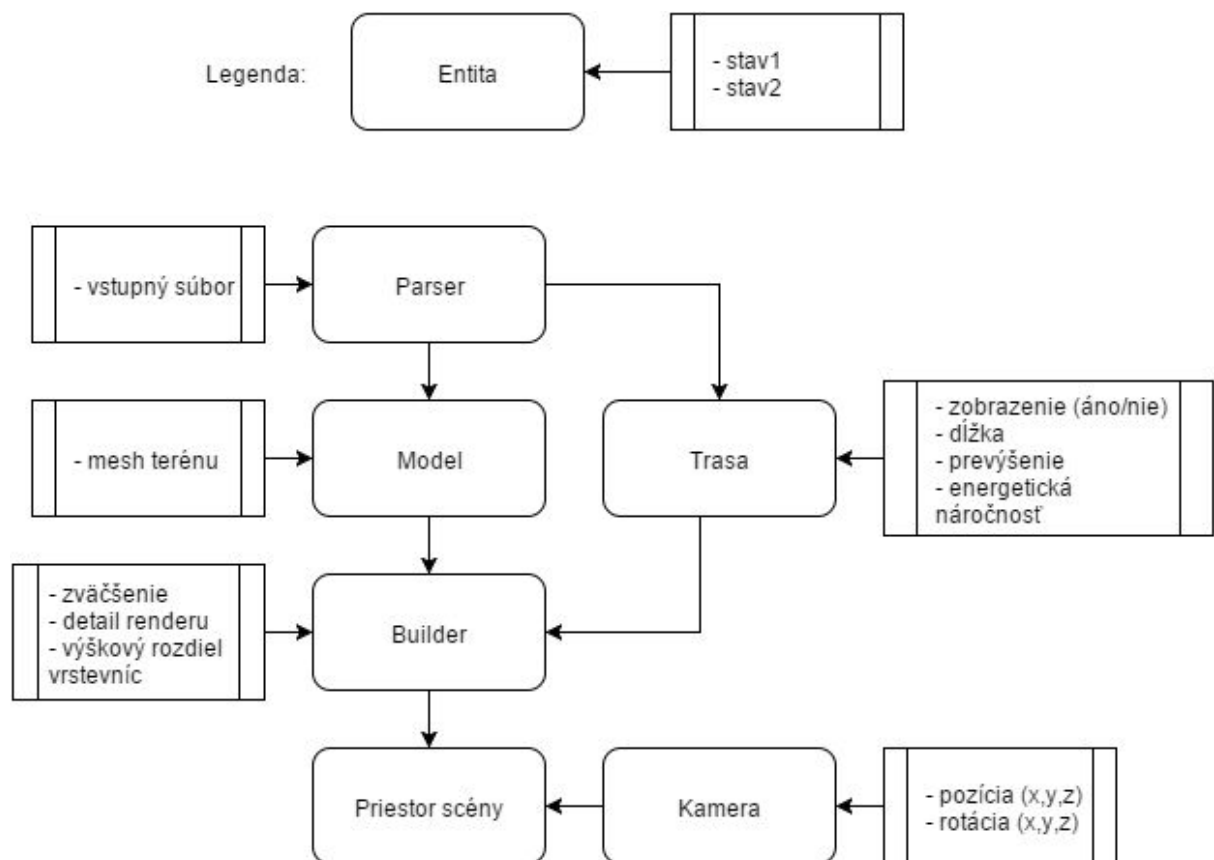
V našej aplikácii je iba jeden modelový používateľ, ktorý má prístup ku všetkým funkciám.

2.2. Use case diagram

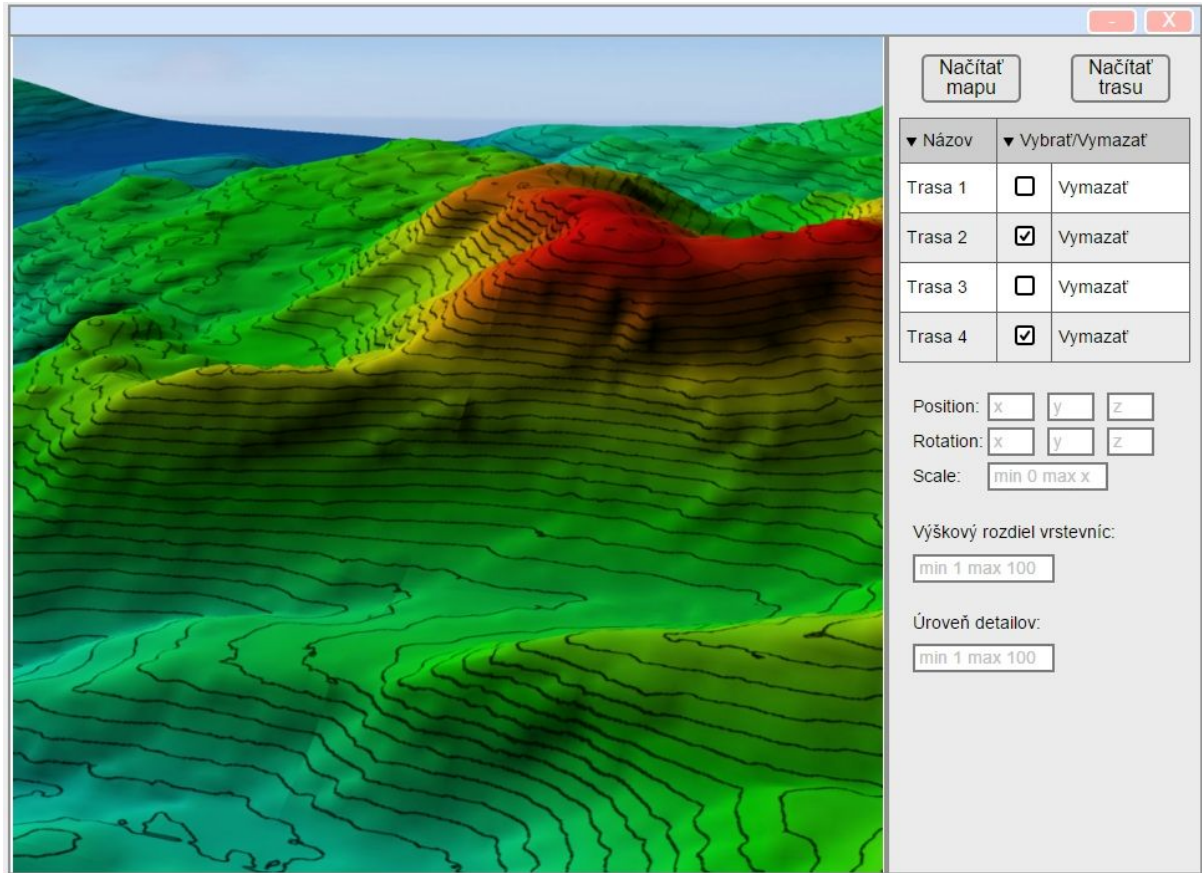
2.3. Cieľ používateľa

Prostredníctvom interakcie s aplikáciou prezentovať priestorovú vizualizáciu terénu a trás.

2.4. Entitno-relačný diagram



2.5. Návrh používateľského rozhrania



- Pri stlačení tlačidla Načítať mapu alebo Načítať trasu vyskočí systémové okno pre výber súboru.
- Pri stlačení tlačidla Vymazať sa vymaže príslušná načítaná trasa
- Checkbox slúži na výber, či sa trasa má alebo nemá zobrazovať na modeli
- Pozícia, rotácia, úroveň zväčšenia, výškový rozdiel vrstevníc, úroveň detailov sa zadávajú do textboxov ako číselné hodnoty
- Objekt sa dá približovať alebo oddalovať pomocou kolieska na myši
- Objekt sa dá rotovať stlačením pravého tlačidla na myši a tahaním rôznymi smermi súčasne aj na viacerých osiach
- Objekt sa dá posúvať stlačením ľavého tlačidla na myši a tahaním rôznymi smermi súčasne aj na viacerých osiach

3. Analýza technológií, dekompozícia a dátový model

3.1. Analýza technológií

Pojmy:

omap - jedná sa o súbory podobné xml. Slúžia teda na ukladanie štruktúrovaných dát, v našom prípade údajov o vrstevniciach v mape

parsovanie - spracovanie dát z textových súborov do dátových štruktúr

mesh - kostra grafického modelu

Pri tvorbe aplikácie je potrebné zvoliť technológie, ktoré podporujú koncepty použité v návrhu tejto aplikácie.

Robiť aplikáciu od úplného základu sme považovali za stratu času, preto sme sa rozhodli použiť niektoré z existujúcich prostredí, ktoré podporujú prácu s modelmi a umožňujú rýchlo vytvoriť príjemné a praktické GUI.

Po úvahách a diskusiách sme sa zhodli, že najvhodnejším prostredím pre tvorbu našej aplikácie bude Unity3D. Unity3D má viacero výhod, od jednoduchého budovania modelov (použitím meshov), ktoré bude jadrom našej aplikácie, cez možnosť používať knižnice C#, ktoré nám dovoľia jednoducho parsovať omap súbory (použitím štandardnej triedy XmlReader), po jednoduchú distribúciu aplikácie na ľubovlnú platformu. Taktiež nám vyhovuje možnosť v scéne pracovať s kamerou, čo nám umožní sústrediť sa na funkcionality, miesto toho, aby sme investovali čas do prepočtov rotácii modelov, pannonovania a zoomovania. Unity3D sa teda javí ako najvhodnejšie vzhľadom na naše požiadavky.

Ďalej sme sa zaoberali použitím prostredia Panda3D. Avšak narazili sme na viaceré problémy. Jedným z problémov boli samotné programovacie jazyky podporované v

prostredí. Python sme pre túto rozsiahlejšiu aplikáciu nevybrali kvôli menej vhodným nástrojom na manipuláciu s kódom.

Pri C++ nám nevyhovovalo, že štandardné knižnice nepodporujú spracovanie xml súborov.

3.2. Model vstupného súboru Omap

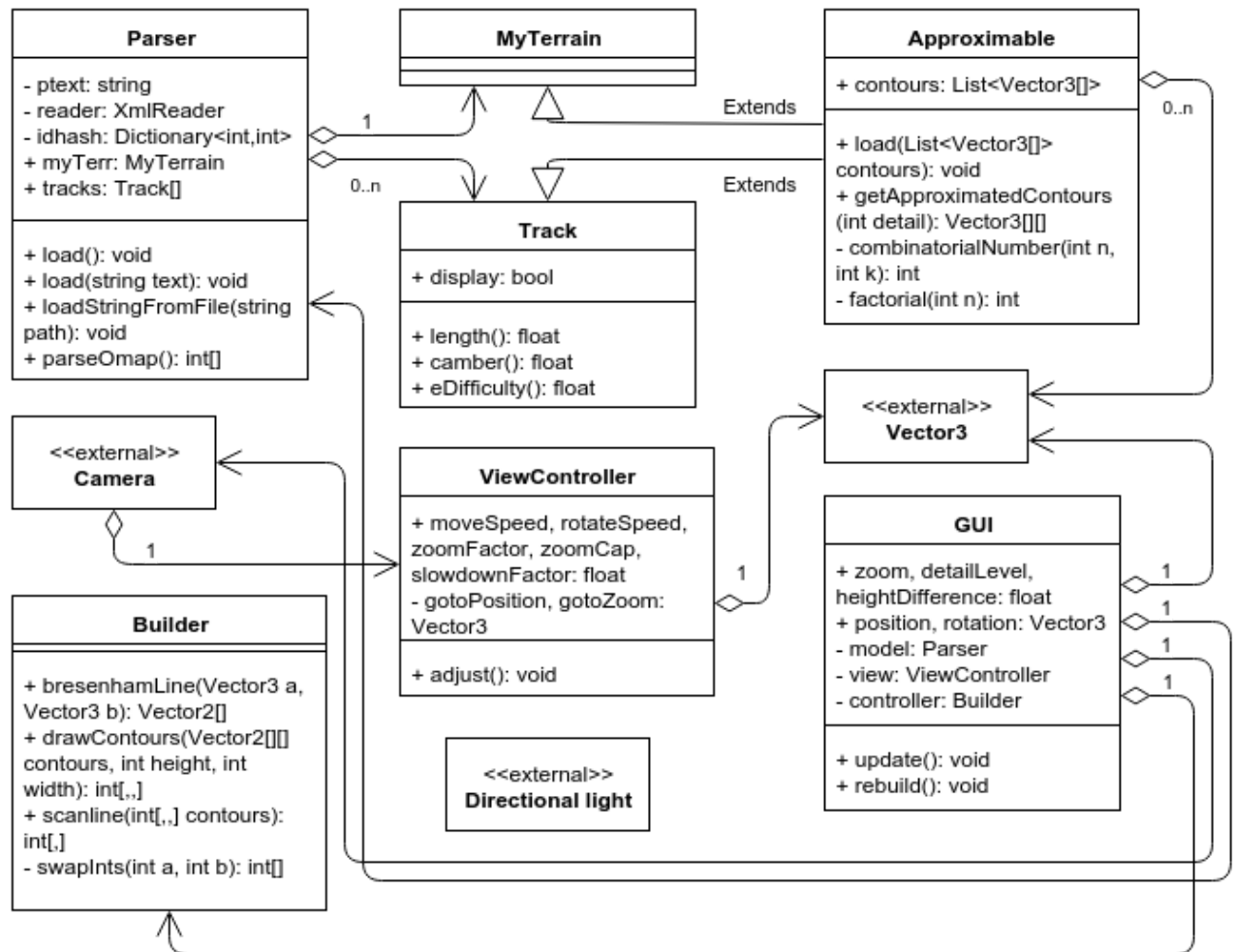
Súbor sa skladá z hlavičky súboru a jadra.

Hlavička obsahuje definície a popisy jednotlivých typov terénu, ktoré sa nachádzajú v súboroch omap. Hlavička je pre nás kľúčom, podľa, ktorého vieme zistiť aké objekty sa nachádzajú v jadre súboru.

Jadro súboru má vlastnú hlavičku, v ktorej sú definované iba terénne objekty použité v konkrétnej mape. Ich ID sú mapované cez hlavičku súboru na konkrétny typ terénu. Nás budú v základnej verzii zaujímať len vrstevnice a trasy. Jedná sa konkrétne o objekty s kódom 101,102,103, 104 a 105. ID sú pridelené v hlavičke jadra súboru, na objekty sa súbor neskôr odvoláva len použitím týchto ID.

Nasleduje samotný výčet všetkých objektov so súradnicami bodov, ktoré tieto terény definujú a upresnením, či sa jedná o hmotný bod objektu alebo len bod používaný na presnejšie dokreslenie kriviek. Toto upresnenie je vyjadrené pomocou flagu. Z tohto zoznamu objektov vyberieme tie, ktoré majú nami požadované ID a zapamätáme si všetky body jednotlivých objektov v objektoch jednotlivých vrstevníc a trás.

3.3. Dátový model



3.4. Podrobná špecifikácia komponentov

Parser - komponent načítavajúci a držiaci údaje potrebné pre chod aplikácie

load([string text]) - načíta zo vstupného súboru oMap potrebné dáta o vrstevniciach. Metóda na načítavanie používa XmlReader a počas načítavania naplní pole vrstevníc v objekte MyTerrain. Okrem toho naplní aj objekty Track údajmi o trasách.

Approximable - má údaje o kontúrach (vrstevnice alebo cesta trasy) z objektu parser pomocou metódy load.

getApproximatedContours(int detail) - vráti kontúry aproximované Bernsteinovým polynómom.

Track - Drží v sebe údaje o trase.

length() - vráti celkovú priestorovú dĺžku trasy

camber() - vráti celkové prevýšenie trasy

eDifficulty() - vráti energetickú náročnosť trasy

Builder - Enkapsuluje metódy pre prácu s terénom a trasami.

bresenhamLine(Vector3 a, Vector3 b) - vráti súradnice úsečky medzi bodmi *a* a *b* rasterizovanej Bresenhamovým algoritmom.

drawContours(Vector2[][] contours, int height, int width) - zakreslí a vráti pole z rasterizovaných úsečiek na mape veľkosti *height* x *width*

scanline(int[,] contours) - preskenuje vodorovne a zvisle vykreslené pole úsečiek a vráti mapu reprezentujúcu výšku na jednotlivých poliach.

ViewController - Združuje ovládacie funkcie kamery. Približovanie, rotácia a pozícia v scéne sa ovláda týmto komponentom.

adjust() - podľa užívateľského vstupu (myšou) upraví vlastnosti kamery v scéne.

GUI - Užívateľské rozhranie zobrazujúce všetky informácie dostupné užívateľovi, umožňuje aj ich editáciu.

update() - obnoví informácie o kamere v rozhraní

rebuild() - pretvorí terén a trasy v scéne na základe zmenených informácií v rozhraní

3.5. Testovacie scenáre

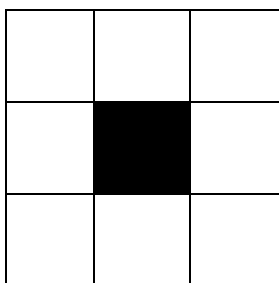
Overenie správnosti DDA algoritmu - testujeme správnosť algoritmu pomocou výsledkov na jednoduchých vstupoch, jedná sa o priamku rovnobežnú s x-ovou osou, uhlopriečku štvorca a jednu úsečku, ktorá je uhlopriečkou obdĺžnika.

Overenie správneho parsovania - Parser vypíše minimálnu a maximálnu hodnotu x-ových a y-ových súradníc. Ďalej skontrolujeme, či sa zo súboru načítali vrstevnice a koľko ich bolo.

Overenie správnej aproximácie kriviek vrstevníc - testy overujú, či správne vytvárame body zo splinov. Pre cyklickú a acyklickú vrstevnicu. Overujeme správnosť Bernsteinových polynómov.

Vizuálne overenie správnosti heightmáp - vizuálne overíme, či naša aplikácia správne vytvorila heightmapu pre súbor.

Napríklad:



Tento vstup má vrátiť mapu, ktorá ma v strede vyvýšenú platformu a okolo rovinu.

Užívateľské testovanie zoomovania, presúvania a rotácie v scéne

Otestovanie správnosti výpočtu náročnosti trasy

Otestovanie správnosti výpočtu dĺžky trasy

3.6. Popis použitých algoritmov

Aplikácia načíta zo súbora údaje o vrstevniciach mapy vo formáte omap.

Tieto dáta, vo forme splinov, aproximuje použitím Bernsteinových polynómov, čím dostaneme konkrétne body ležiace na vrstevniciach.

Do dvojrozmerného poľa vyrasterizujeme pomocou Bresenhamovho algoritmu úsečky týchto vrstevníc.

Pokračujeme dvojitém použitím scanline algoritmu na štvorcovej sieti, tým dostaneme odhad toho, aká je výška jednotlivých plôch medzi vrstevnicami. Následne dáta kvantizujeme, čím zmenšíme ich objem pre rýchlejší beh programu.

Z týchto dát vytvoríme heightmapu ofarbenú odtieňmi šedej, ktorá bude symbolizovať výšky jednotlivých políček dvojrozmerného poľa. Túto heightmapu ďalej posunieme interface-u Unity3D, ktorý vypracuje z nej trojrozmerný objekt, ktorý užívateľovi zobrazíme.