Ukázková aplikace TraceGPS

Michal Altair Valášek | michal.valasek@altairis.cz | www.altairis.cz | www.aspnet.cz

Jak lépe oslavit uvedení nové verze platformy Microsoft .NET (3.5) a vývojových nástrojů (Visual Studio 2008), než vytvořením nové aplikace, která novinky odpovídajícím způsobem využívá?

Poslední dobou si hodně hraju s GPS. Jednou z funkcí, které většina GPS má, je záznam prošlé trasy do logu. Napadlo mne podívat se na mapě, kde všude vlastně s čubou trajdám, a proto jsem vytvořil aplikaci **TraceGPS**. Ta umí vzít záznam prošlé trasy v NMEA formátu a zobrazit ho na mapovém podkladu přes AMapy API (http://amapy.atlas.cz/). Využívá přitom řadu nových funkcí z Microsoft .NET Frameworku verze 3.5.

Architektura aplikace

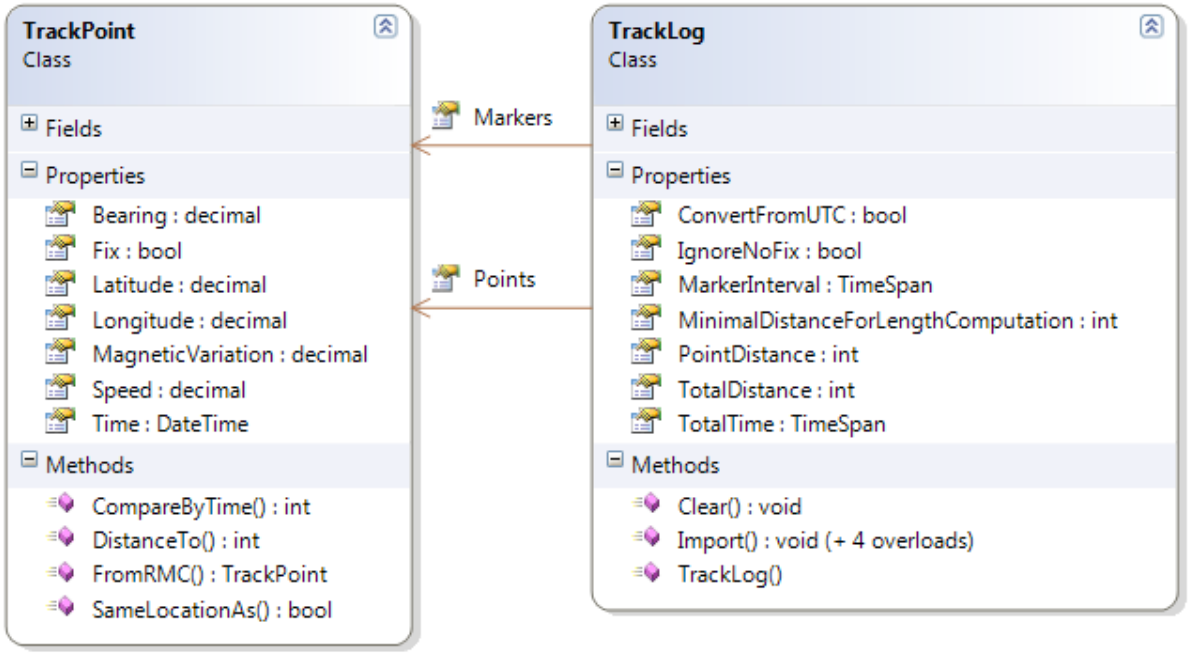
Aplikace je velmi jednoduchá a sestává v podstatě ze dvou částí. První se stará o parsování logu v NMEA formátu a jeho převod na strongly-typed kolekci pozičních bodů. Jedná se o třídy umístěné ve složce `~/App_Data/Altairis.GPS/` (v logice kódu jsou umístěny v namespace `Altairis.GPS`). Druhou polovinu aplikace tvoří uživatelské rozhraní a mashup s AMapy API.

Chtěl jsem, aby jednotlivé části na sobě nebyly přímo závislé, aby parsování NMEA dat bylo možné využít i pro jinou aplikaci, aby knihovna byla univerzální a ne jednoúčelová pro AMapy. Z tohoto důvodu jsem využil novou funkcionalitu, konkrétně extension methods. Samo parsování pak využívá některé další vymoženosti, v .NET 3.5 jsoucí.

Parsování NMEA log souboru

Záznam ve formátu NMEA je to nejjednodušší, co z GPSky můžete dostat. Jedná se v podstatě o hrubý záznam komunikace, jedná se o data, která GPSka posílá do počítače například přes sériový port.

NMEA log je reprezentován dvojicí tříd:



Třída `Altairis.GPS.TrackPoint` reprezentuje jeden konkrétní zaznamenaný bod trasy. Má celou řadu vlastností, které v podstatě odpovídají polím RMC sekvence (`$GPRMC`[1]). Při vytváření těchto vlastností jsem hojně využíval nové funkcionality v novém .NETu, a to *automatic properties*.

Pokud jste v předchocích verzích .NETu potřebovali vytvořit jednoduchou property, která jenom ukládala do privátní proměnné svou hodnotu, museli jste napsat něco takového:

```
private int vlastnost;

public int Vlastnost {
  get { return this.vlastnost; }
  set { this.vlastnost = value; }
}
```

Nový .NET vám umožňuje shorthand zápis ve zkrácené podobě, která je výrazně přehlednější, zejména je-li takových vlastností hodně:

```
public int Vlastnost { get; set }
```

Třída `TrackPoint` oplývá kromě vlastností i několika metodami. Statická metoda `FromRMC` slouží k vytvoření instance třídy z textové věty z logu, potažmo přijaté z GPS.

[1] Popis parsování RMC sekvence najdete například na http://www.codepedia.com/1/The+GPRMC+Sentence

Dále pak máme dvě jednoduché metody, které budeme využívat později: `SameLocationAs` a `CompareByTime`. První jmenovaná kontroluje, zda se dva TrackPointy nacházejí na stejných zeměpisných souřadnicích (a ignoruje ostatní parametry), druhá slouží k seřazení bodů dle časové posloupnosti.

Nejkomplikovanější je metoda `DistanceTo`, která umí spočítat vzdušnou vzdálenost mezi dvěma body, určenými pomocí zeměpisných souřadnic[2].

Třída `Altairis.GPS.TrackLog` reprezentuje celý záznam a obsahuje funkcionalitu k jeho parsování. Kromě jiného má tato třída vlastnosti `Points` a `Markers`. Při parsování se odpovídající pole naplní instancemi třídy `TrackPoint`, odpovídajícími bodům pro vykreslení trasy (`Points`) a značek (`Markers`) na mapě. Idea této operace je následující:

NMEA logfile obsahuje výpis RMC (a jiných) sekvencí z GPS. Je věcí nastavení odpovídajícího software, jak často bude data do logu ukládat, ale může tak být zbytečně často, třeba jednou za deset sekund. Pro účely zpracování a zobrazení na mapě ale potřebujeme jenom některé záznamy – ty, u nichž se od posledního měření poloha změnila o více než `PointDistance` metrů.

Vhodnou hodnotu vlastnosti `PointDistance` bude nutné určit experimentálně, v závislosti na přesnosti použité GPS a také na rychlosti pohybu (bude jiná pro pěší chůzi a jiná pro jízdu autem). Pokud nastavíte vzdálenost jako příliš velkou, bude trasa „zubatá“ a nebude přesně kopírovat vaši cestu. Pokud ji nastavíte příliš krátkou, může být vinou chyby zaměření naopak příliš „kudrnatá“ a její vykreslení bude komplikované a pomalé.

Dále pak budeme chtít na mapě zobrazit značku (marker) a v pravidelných časových intervalech. Délku intervalu lze stanovit pomocí vlastnosti `MarkerInterval`. I zde platí, že vhodnou hodnotu je nutné určit experimentálně.

Poslední zákeřnost se skrývá ve výpočtu celkové překonané vzdálenosti a na ní závislé průměrné rychlosti. Pro odfiltrování chyb měření musíme použít podobný algoritmus, jako o dva odstavce výše. Vhodná hodnota ale může být odlišná od hodnoty pro potřeby zobrazení na mapě, pročež máme k dispozici samostatnou nastavovací vlastnost jménem `MinimalDistanceForLengthComputation`. Pokud vám vychází podezřele velká délka trasy, pak se vám projevuje chyba měření a počítáte, jako kdybyste chodili „cik-cak“ – je třeba zvýšit uvedenou vzdálenost.

Vlastní import logu probíhá, nepříliš překvapivě, pomocí metody `Import`. Ta má několik komfortních overloadů, které vám umožní načítat data ze streamu, souboru nebo `TextReaderu`. Vzhledem k tomu, že logy mohou být poměrně objemné, ale zato se dobře komprimují, přidal jsem do třídy též podporu pro logy komprimované pomocí GZip nebo Deflate, včetně autodetekce[3].

Vykreslení do mapy

K vykreslení markerů a trasy do mapy využívám mashup s mapami na portálu Atlas.cz – AMapy API. Podrobný popis tohoto rozhraní včetně příkladů najdete na adrese http://amapy.atlas.cz/api/.

[2] Popis použitého algoritmu (včetně teorie, která je za ním) najdete například na http://mathforum.org/library/drmath/view/51879.html

[3] Obšírnější pojednání o využití a detekci těchto komprimačních algoritmů najdete v mém článku *GZIP komprese v .NET: Jak ji poznat a využít?* na serveru ASPNET.CZ: http://www.aspnet.cz/Articles/167.aspx

Většina relevantního JavaScriptu je napevno zapsána na konci souboru `~/Default.aspx`. Dynamicky se generuje pouze obsah kolekcí `points[]` a `markers[]`.

Bylo by skvělé, kdyby sám bod (třída `TrackPoint`) měl schopnost vytvořit svoji reprezentaci v klientském skriptu pro AMapy API. Mohli bychom samozřejmě vytvořit nějakou metodu, která by tuto funkcionalitu dodala (jde přece jenom o vrácení jednoho stringu). Nicméně tím bychom svázali předmětnou třídu s tímto konkrétním mapovým API a dokonce i s naší konkrétní aplikací – a já jsem chtěl, aby byla pokud možno univerzální.

Zde nám přicházejí na pomoc *extension methods*, další z novinek v .NET 3.5. Extension methods umožnují (lapidárně řečeno) libovolnému typu zvenčí naroubovat metody, aniž bychom museli jakkoliv modifikovat třídu samu, ba dokonce k ní ani nemusíme mít zdrojové kódy.

Stačí vytvořit statickou třídu a v ní statické metody, jejichž první parametr bude uvozen slovíčkem `this`. Vizte následující příklad, který ke stringu přidá metodu `IsUpperCase`, která vrací `true` nebo `false`, podle toho zda je předmětný řetězec psán velkými písmeny (přesněji velkými písmeny a case-neutrálními znaky, jako je příkladně interpunkce):

```
public static class MojeExtensionMethods {
    public static bool IsUpperCase(this string s) {
        return s.Equals(s.ToUpper(), System.StringComparison.Ordinal);
    }
}
```

V dalším kódu pak stačí napsat něco následujícím duchu:

```
string x = "AHOJ";
bool jeUpperCase = x.IsUpperCase();
```

S extension methods je řešení našeho problému snadné. Vytvořím si třídu `AMapyExtensions` a v ní definuji nad typy `TrackPoint` potažmo `TrackLog` metody `ToAMarkerScript`, `ToAGeoPointScript` a `ToAMapyScript`, které mi vygenerují odpovídající JavaScript.

Závěrem

Aplikace sama je bohatě komentována, podrobnější informace tedy získáte nahlédnutím do kódu. K jejímu spuštění budete potřebovat web server podporující Microsoft .NET Framework verze 3.5. Aplikace byla otestována pod pre-release verzemi Beta 2, Release Candidate 0 a pod finální (RTM) verzí. K editaci ukázkového projektu postačí Microsoft Visual Web Developer Express 2008, který si můžete stáhnout zdarma[4].

Licence

[4] Na adrese http://go.microsoft.com/?linkid=7729281