

České vysoké učení technické v Praze  
Fakulta elektrotechnická

BAKALÁŘSKÁ PRÁCE

2016

Martin Průcha

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student: **Martin Průcha**

Studijní program: Kybernetika a robotika

Obor: Systémy a řízení

Název tématu: **Navigace v budovách**

### Pokyny pro vypracování:

1. Prostudujte systémy navigace pro návštěvníky v rozsáhlých a složitých budovách jako jsou zejména úřady, nemocnice, ap. Zaměřte se na systémy využívající jednoduché principy (značky, tabulky, mobilní telefony, případně náramky).
2. Stanovte jejich výhody a nevýhody z tech. hlediska. Zaměřte se zejména na parametry: náročnost, modularita, instalace, rozšíření, snadnost pro uživatele, vhodnost pro osoby se sníženou orientací.
3. Navrhněte systém pro navigační a částečně lokalizaci založený na:
  - a) Snadnost nasazení a aktualizace (například vhodné umístění NFC tagů)
  - b) Snadnost využití návštěvníkem (například pomocí mobilního telefonu)
  - c) Tzv. objektová mapa poskytující informace pro hledání cesty
  - d) Přesnost navigace a lokalizace vs. celková náročnost a cena systému
4. Na omezeném prototypu ověřte vhodnost tohoto návrhu a zhodnoťte jeho technické přínosy a nedostatky.

### Seznam odborné literatury:

- [1] Beginning NFC, O'Reilly, 2014, ISBN: 978-1-449-37206-4
- [2] MS.NET C#, WPF (WWW stránky, dokumentace, knížky pro programování)
- [3] Xamarin Android Application Development, Diptimaya Patra, 2014
- [4] Xamarin Cross-platform Application Development - Develop production-ready applications for iOS and Android using Xamarin, Jonathan Peppers, PACKT Publishing, 2014

Vedoucí: Ing. Petr Novák, Ph.D.

Platnost zadání: do konce letního semestru 2016/2017

L.S.

prof. Ing. Michael Šebek, DrSc.  
vedoucí katedry

prof. Ing. Pavel Ripka, CSc.  
děkan

V Praze dne 11. 1. 2016

## **Prohlášení**

---

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze, dne .....

.....

Podpis

## Abstrakt

---

Tato práce se zabývá možností realizace snadné navigací lidí (zejména návštěvníků) ve složitých a rozlehlých budovách (například úřady, nemocnice), její realizace za použití technologie NFC a implementací algoritmu pro vyhledávání optimálních cest do zvoleného cíle. Výstupem této práce je demonstrační program pro operační systém MS Windows, umožňující vyhledat v budově místo požadované návštěvníkem, a poté najít k němu vhodnou cestu.

**Klíčová slova:** Bakalářská práce, Navigace v budovách, .NET, C#, Prohledávání stavového prostoru

This thesis deals with possibility of easy navigation in complex and vast building (eg. Offices, hospitals), usage of NFC technology for navigation purposes and implementation pathfinding algorithm. The outcome of this work is computer program, for MS Windows operating system, which allows to users looked up place which they want and find suitable path to there.

**Key words:** Bachelor thesis, Inside navigation, .NET, C#, State space scanning

# Obsah

---

<b>Kapitola 1 Úvod</b> .....	<b>1</b>
<b>Kapitola 2 Současné Navigační systémy v budovách</b> .....	<b>2</b>
2.1 Metody určení polohy v budovách.....	2
2.1.1 <i>Optické metody</i> .....	2
2.1.1.1 <i>Reference z obrázků</i> .....	2
2.1.1.2 <i>Optické tagy jako reference</i> .....	3
2.1.2 <i>Wi-Fi</i> .....	3
2.1.2.1 <i>Původ vysílače</i> .....	3
2.1.2.2 <i>Otisk okolí</i> .....	4
2.1.3 <i>RFID</i> .....	4
2.1.4 <i>Bluetooth</i> .....	4
2.2 Existující systémy .....	4
2.2.1 <i>Vytištěné statické mapy</i> .....	5
2.2.2 <i>Indoor navigace od firmy Symbion</i> .....	5
2.2.3 <i>Navigační systém od Emam [5]</i> .....	6
<b>Kapitola 3 Cíle</b> .....	<b>7</b>
<b>Kapitola 4 Návrh řešení</b> .....	<b>8</b>
4.1 Použité technologie .....	8
4.1.1 <i>NFC</i> .....	8
4.1.1.1 <i>NFC zařízení</i> .....	9
4.1.2 <i>.NET Framework</i> .....	10
4.1.2.1 <i>Architektura</i> .....	10
4.1.2.2 <i>C#</i> .....	11
4.1.2.3 <i>Windows Presentation Foundation (WPF)</i> .....	11
4.1.2.4 <i>Xamarin</i> .....	12
4.2 Návrh.....	12
4.2.1 <i>Mapa</i> .....	13
4.2.1.1 <i>Blok mapy</i> .....	13
4.2.1.2 <i>Objekt mapy</i> .....	13
4.2.1.3 <i>NFC tagy v mapě</i> .....	13

4.2.2 Grafické rozhraní.....	13
4.2.2.1 Interaktivní mapa.....	14
4.2.2.2 Podpora více jazyků.....	14
<b>Kapitola 5 Implementace (Řešení).....</b>	<b>15</b>
5.1 Mapa.....	15
5.1.1 Blok mapy.....	15
5.1.2 Objekt mapy .....	16
5.1.2.1 Potomci objektu mapy.....	17
5.1.3 Základní třída mapy.....	17
5.2 Práce s NFC.....	18
5.3 Grafické rozhraní (GUI).....	19
5.3.1 Postranní panel.....	19
5.3.1.1 Tlačítka.....	19
5.3.1.2 ScrollView.....	21
5.3.2 3D zobrazení mapy.....	21
5.3.2.1 Interaktivita mapy .....	21
5.4 Podpora více národnostních jazyků .....	22
5.5 Ukládání do XML .....	23
5.6 Algoritmus pro nalezení optimální cesty .....	24
<b>Kapitola 6 Používání aplikace.....</b>	<b>25</b>
6.1 Ovládání kiosku (informačního panelu).....	25
6.1.1 Mapa .....	25
6.1.2 Postranní panel.....	26
6.1.3 Navigace a lokalizace po budově.....	26
6.2 Správa systému.....	28
6.2.1 Editace souborů mapy.....	29
6.2.2 Editace národnostních jazyků .....	30
<b>Kapitola 7 Testování .....</b>	<b>31</b>
7.1 Aplikace pro kiosky.....	31
7.2 NFC tagy .....	32
7.3 Demonstrační mapa.....	32

7.4 Lokalizace mobilním zařízením.....	32
7.5 Plánování cesty.....	33
7.6 Netestované části.....	33
<b>Kapitola 8 Závěr.....</b>	<b>34</b>
8.1 Budoucí rozvoj.....	34
<b>Přílohy A – Použité zkratky .....</b>	<b>36</b>
<b>Příloha B – Obsah CD.....</b>	<b>37</b>

## Obrázky

---

Obrázek 2.1 Databáze pořízených obrázků (Zdroj:[2]) .....	2
Obrázek 2.2 Aktuální obrázek.....	2
Obrázek 2.3 Tři příklady optických tagů (Zdroj: [3]) .....	3
Obrázek 2.4 Příklad grafického rozhraní aplikace pro navigaci v budovách od firmy Symbion (Zdroj: [6]).....	5
Obrázek 2.5 Příklad nabízeného kiosku a uživatelského rozhraní interaktivní mapy (Zdroj: [7]) .....	6
Obrázek 4.1 : Pasivní NFC tag.....	9
Obrázek 4.2 Struktura Common Language Infrastructure.....	11
Obrázek 5.1 Vyobrazení použitých souřadnic .....	16
Obrázek 5.2 Náhled na celé grafické prostředí .....	19
Obrázek 6.1 3D mapa s vyznačenou pozicí kiosku.....	25
Obrázek 6.2 Zvýraznění tagu v mapě .....	25
Obrázek 6.3 Okno s informacemi o místnosti.....	26
Obrázek 6.4 Při vybrání určitého typu místnosti se všechny místnosti tohoto typu .....	27
Obrázek 6.5 Zobrazení nově vytvořené cesty .....	27
Obrázek 6.6 Aktualizovaná pozice pomocí tagu.....	28
Obrázek 6.7 Ukázka programu XML Notepad .....	28
Obrázek 7.1 MeeGo Pad T01 .....	31
Obrázek 7.2 Dotyková obrazovka.....	31



## **Kapitola 1 Úvod**

---

Dostatečně přesná navigace lidí ve složitých a zejména rozlehlých budovách je velice obtížný problém [1], jelikož jsou zde klasické technologie pro venkovní navigaci jako GPS nedostupné. Možností řešení tohoto problému je několik např. za pomoci Wi-Fi vysílačů rozmístěných po budově, nebo využití kombinace Wi-Fi a senzorů v chytrých telefonech. Tato řešení vyžadují vybudovat často nejen složitou, ale i nákladnou infrastrukturu v budově, a i přes tuto náročnost nejsou vždy dostatečně přesné (viz. Kapitola 2.1). Proto jsem se pokusil navrhnout přehledný systém navigace s, pro mnoho účelů, dostačující přesností.

Nejprve budou stručně probrány, již existující metody a případně jejich výhody a nevýhody. Dále podán návrh vlastního řešení využívající velmi levné technologie. A nakonec popsána implementace vytvořené demonstrační aplikace.

## Kapitola 2 Současné Navigační systémy v budovách

---

### 2.1 Metody určení polohy v budovách

V současné době existuje několik metod, principů a postupů využívaných pro navigaci lidí (nejčastěji návštěvníků) ve složitých a hlavně rozsáhlých budovách. Mezi takovéto budovy považujeme zejména různé úřady, nákupní centra a nemocnice. Někdy se jedná pouze o navigaci při pohybu, jindy jde až o skutečné určení aktuální polohy, tedy v podstatě lokalizaci. Proberme některé v současnosti využívané metody podrobněji. Nejprve příklady několika technologií, na kterých jsou tyto systémy založeny.

#### 2.1.1 Optické metody

Přesnost určení polohy pomocí optických senzorů se pohybuje, podle použité technologie, od jednotek metrů až po milimetry. Dnes nejrozšířenějším senzorem je CCD senzor (tedy kamera) přítomný ve všech chytrých telefonech tzv. smartphones.

Zjištění aktuální pozice pomocí zmíněného optického senzoru lze rozdělit na případ, kdy se senzor (v našem případě fotoaparát v telefonu) pohybuje a naopak na případ, kdy statický senzor (pevná kamera) detekuje pohybující se objekty v jeho okolí. Největším problémem u optických senzorů určujících polohu je, jak převést získaný 2D obraz světa do skutečné 3D reprezentace objektu (budovy). K vyřešení tohoto problému lze použít dva CCD senzory, což je ale u moderních telefonů ne vždy dostupné, jelikož toto řešení nemá v telefonu zatím žádné reálné využití.

##### 2.1.1.1 Reference z obrázků

Tato metoda využívá databázi předem pořízených obrázků (viz Obrázek 2.1) sloužících vhodně k následnému porovnání nově pořízeného obrázku, například fotoaparátem v mobilním zařízení (viz obrázek 2.2). Největším problémem této metody spočívá v dosažení odezvy v reálném čase, jelikož zpracování a porovnávání obrázku s celou databází je pro moderní mobilní telefon stále velmi náročný úkol. I zde vznikají další vcelku velké problémy a to jsou: měnící se světlost, různé úhly pohledu a další nepřijemnosti.

J. Ido [2] a jeho tým experimentálně implementoval tuto metodu. Pro získání prvotních (tedy referenčních) snímků umístí na hlavu robota kameru zaznamenávající v podstatě



**Obrázek 2.1** Databáze pořízených obrázků (Zdroj:[2])



**Obrázek 2.2** Aktuální obrázek (Zdroj:[2])

průchod budovou. Po analýze, kompresy a uložení získaných dat byl Ido schopen určit polohu robota s přesností na 30 cm.

### 2.1.1.2 Optické tagy jako reference

Předchozí metoda není příliš robustní, jelikož je značně závislá na okolním osvětlení a také předpokládá určitou neměnnost sledovaného okolí (prostředí budovy). U této metody snadno narazíme na problém, kdy pořídíme fotografii ve dne nebo v noci a rozdílné světelné podmínky mohou způsobit i velké chyby v lokalizaci.



Obrázek 2.3 Tři příklady optických tagů (Zdroj: [3])

Ke zvýšení robustnosti těchto systému se mohou používat tzv. optické značky např. QR kód, korekční kroužky nebo vzor složený z barevných teček (viz. Obrázek 2.3).

Firma Sky-Trax Inc. [3] vyvinula na tomto principu optický navigační systém pro vysokozdvizné vozíky používané ve skladišti. Na stropě byly rozmístěny značky (směrem dolů), a na střechu každého vozíku umístěna kamera (směrem nahoru) snímající značky rozmístěné na stropě. Získaný obrázek stropu se odešle na server, tam se zpracuje a tím se vyhodnotí aktuální pozice vysokozdvizného vozíku. Přesnost tohoto řešení se pohybuje mezi 2 cm a 30 cm.

### 2.1.2 Wi-Fi

Díky rozvoji bezdrátových sítí Wi-Fi a jejich velké hustotě pokrytí je dnes možné implementovat tímto stylem rovněž způsob navigace a to celkem levně. Komunikace po bezdrátové síti je určena standardy IEEE<sup>1</sup> 802.11. Využívají se pásma 2,4 GHz a 5 GHz, tudíž se za použité frekvenční pásmo nemusí platit licenční poplatky. Výhoda oproti optickým metodám lokalizace / navigace spočívá v nenutnosti přímé viditelnosti mezi vysílačem a přijímačem, jelikož elektromagnetické záření o výše zmíněných frekvencích může s útlumem proniknout i pevnými objekty (zdi, dveře). Útlum jednotlivými objekty je však kritický pro určení pozice pomocí Wi-Fi. Tedy na základě v jakém prostředí se systém používá, se poté volí model propagace signálu prostředím.

#### 2.1.2.1 Původ vysílače

Jde o nejjednodušší metodu využívající předpokladu, kdy si moderní zařízení připojující se k Wi-Fi síti, vybírají vysílač s největší intenzitou v daném místě. Následně se aktuální

---

<sup>1</sup> IEEE - Institute of Electrical and Electronics Engineers

poloha určí ze známé polohy vysílače. Kvůli dosahu Wi-Fi vysílačů v řádu i stovek metrů je takto získaná poloha velice nepřesná, a u více podlažních budov může dojít i ke špatnému určení podlaží, ve kterém se nacházíme. Pro zpřesnění lokalizace / navigace se využívá měření síly signálu z několika dostupných vysílačů.

#### 2.1.2.2 Otisk okolí

Pokud chceme tuto metodu přesto používat, musíme nejdříve zmapovat sílu signálu, ze všech dostupných vysílačů (vytvořit tzv. „otisk“ okolí), v různých místech budovy. Skutečnou pozici poté získáme porovnáním „otisku“ získaného uživatelem s již vytvořenou databází. Přesnost této metody se pohybuje v rámci jednotek metrů v závislosti na hustotě vysílačů a rovněž konstrukci budovy.

Nevýhodou je však stále značná závislost na aktuálním okolí, přesnost může nepříjemně ovlivnit např. přesunutí kovového nábytku, otevření / zavření silných dveří nebo dokonce i pohybující se člověk.

Tuto metodu využívá např. Google, Microsoft, Apple,... pro rychlejší lokalizaci uživatele v městských prostorech, avšak v kombinaci s GPS.

#### 2.1.3 RFID<sup>2</sup>

Takovýto systém se skládá z radaru s anténou naslouchajícím okolním aktivním vysílačům (sami poskytují identifikaci) nebo pasivním tagům (poskytují identifikaci na dotaz). Přesnost systému je závislá na hustotě pokrytí budovy vysílači. Nejpoužívanější metoda získávání pozice je pomocí měření intenzity signálu, kde se podle síly signálu určí pozice, na základě známého umístění vysílače/tagu. Samozřejmě lze použít metodu jak statických radarů a pohybujícího se tagu, nebo naopak pohybujícího se radaru a několika statických tagů.

#### 2.1.4 Bluetooth

Tuto technologii lze rovněž využít k lokalizaci a navigaci v budově (obdobně jako WI-FI), avšak její dosah je zřejmě nižší, běžně 5 – 10 m. Protože má technologie Bluetooth velkou latenci nedá se použít k navigaci, kde by zařízení s jednotlivými vysílači komunikovalo, ale dá se použít na navigaci, kde zařízení předem zná pozici vysílačů.

## 2.2 Existující systémy

Nyní příklady několika ucelených navigačních systémů z pohledu tématu této práce. Dnes nabízené systémy navigace v budovách jsou většinou vytvářeny tzv. na míru ve formě stojanů s interaktivními mapami. Tudiž má výrobce softwaru pod kontrolou veškerou údržbu tohoto systému. Tomuto jsem se snažil v mém (dále popsaném) návrhu maximálně vyhnout. Dále tedy uvedu některá již existující (komerční) řešení.

---

<sup>2</sup> RFID - Radio Frequency IDentification

### 2.2.1 Vytištěné statické mapy



Toto možnost představuje nejstarší a nejlevnější řešení. Většina návštěvníků budovy tomuto stylu rozumí, avšak neposkytuje se žádná interakce. Statická mapa však musí obsahovat velké množství informace o budově a v ní umístěných objektech, tedy o všech místnostech/obchodech. Kvůli absenci interaktivního vyhledávání může být orientace v takovéto mapě i značně nepřehledná.

### 2.2.2 Indoor navigace od firmy Symbion

Firma Symbion [4] nabízí kompletní řešení navigace v budovách zahrnující mobilní aplikaci i analytické nástroje. Navigační systém je založen na technologii Wi-Fi (viz kapitola 2.1.2).



Obrázek 2.4 Příklad grafického rozhraní aplikace pro navigaci v budovách od firmy Symbion (Zdroj: [6])

### 2.2.3 Navigační systém od Emam [5]

Navigační systém od firmy Emam [5] nenabízí aplikaci do mobilního telefonu. Poskytuje však pouze kiosky (informační portály) obsahující interaktivní mapu. K tomuto rovněž nabízí tzv. virtuální prohlídku obchodů.



Obrázek 2.5 Příklad nabízeného kiosku a uživatelského rozhraní interaktivní mapy (Zdroj: [7])

## Kapitola 3 Cíle

---

Pokud jste někdy v poslední době navštívili velké nákupní centrum nebo velkou nemocnici / úřad, tak jste určitě zjistili, že orientace v takovémto objektu není zdaleka snadná. Navíc v každé budově mají často poněkud odlišný systém (zobrazení mapy, umístění značek, rozcestníky). Proto jsem se rozhodl, pokusit se navrhnout a implementovat navigační systém, který bude co nejjednodušší (pro uživatele), nejlevnější (pořízení a údržba), snadno rozšiřitelný (doplnění informací), bude se snadno ovládat, a přesto bude dostatečně přesný.

Celý navigační systém by měl být zejména velmi snadno rozšiřitelný. Tedy pokud chce správce / provozovatel systému aktualizovat mapu nebo přidat do budovy lokalizační NFC tag stačí pouze aktualizovat příslušné soubory (mapu, seznam tagů), které jsou v pro člověka co možná nejlépe přehledném formátu.

Grafické rozhraní aplikace by mělo být rovněž co nejjednodušší a obsahovat co nejméně vrstev, tak aby uživatel získal potřebnou informaci při vykonání co nejmenšího počtu úkonů / akcí.

Celková mapa by měla být navržena za účelem snadného přístupu k jednotlivým objektům mapy, podobně, jako je tomu v budovách, kdy mapa obsahuje objekty jako například chodby, místnosti atd., které mají vlastnosti jako jejich předobrazy ve skutečném světě.

Práce se v žádném případě nezabývá vytvořením komerčně odladěného systému, ani nikterak neřeší zabezpečení před nechtěným vniknutím do tohoto systému nebo tzv. naboháním. Práce se tedy věnuje pouze principiálnímu návrhu a ověření jeho základní činnosti.

## Kapitola 4 Návrh řešení

---

Celý systém by měl být pro uživatele co nejpřívětivější a co nejméně náročný na obsluhu. Proto by se systém měl skládat ze tří hlavních částí:

- Aplikace pro kiosky (informační panely).
- Aplikace na mobilní zařízení, pro step-by-step navigaci.
- Soubor NFC tagů viditelně rozmístěných po celé budově.

NFC tagy jsou v podstatě nálepky umístěné na zdech po celé budově, a případně doplněny další kresbou / znaky, tak aby byly co nejlépe viditelné. Tagy budou sloužit k lokalizaci návštěvníka prostřednictvím mobilní aplikace.

Kiosek by měl být hlavní částí navigačního systému, tedy centrální (vstupní) bod kde si návštěvník budovy (uživatel) vyhledá, jakou místnost chce navštívit a jak se do ní skutečně dostane. Informační panel by měl být stojan s dotykovou obrazovkou o úhlopříčce cca 100 cm. Na ní bude vyobrazena 3D interaktivní mapa. Na této mapě je zvýrazněna pozice kiosku a všech NFC tagů rozmístěných po budově.

Pokud uživatel neví, do jaké konkrétní místnosti směřuje, ale vyžaduje pouze specifickou službu / typ místnosti může použít postranní panel, ve kterém je uveden seznam typů místností (podle jejich významu). Při kliknutí na určitou položku se zobrazí seznam místností požadovaného typu, ze kterých si může uživatel vybrat. Po výběru místnosti se zobrazí informace o dané místnosti, v mapě se místnost rovněž zvýrazní a je uživateli nabídnuta možnost naplánování cesty do vybraného cíle.

Naopak pokud uživatel předem ví, do jaké místnosti se chce dostat, ale nezná k ní přesnou cestu, lze kliknutím na tuto místnost v mapě, zobrazit informace o ni a taktéž využít možnost navigace.

V aplikaci pro mobilní zařízení by neměla být, kvůli snížení náročnosti na výpočetní výkon zařízení, tudíž snížení spotřeby a zvýšení výdrže, mapa ve 3D formátu. Je tedy uvažováno pouze o klasické 2D mapě. Aplikace je zamýšlena pro step-by-step navigaci fungující díky známe pozici tagů. Uživatel si zvolí svůj cíl a aplikace vyhledá optimální trasu budovou. Poté když bude chtít uživatel znát svoji aktuální pozici, přiloží své mobilní zařízení k nejbližšímu tagu a aplikace aktualizuje trasu na základě této nově získané pozice.

### 4.1 Použité technologie

Pro tvorbu této práce musí být vybrány a tedy zvoleny některé technologie vhodné k jejímu úspěšnému zvládnutí. Tato kapitola se zabývá nejen výběrem vhodných technologií, ale rovněž i jejich zdůvodněním.

#### 4.1.1 NFC

NFC neboli Near Field Communication je technologie rádiové komunikace na velmi krátkou vzdálenost obvykle do 4 cm [6]. Tuto technologii definuje sada standardů ISO vydaných neziskovou organizací NFC Forum. Jde o technologii dnes velmi často využí-



vanou například pro bezkontaktní platby pomocí mobilního telefonu, kde je navázání komunikace znatelně rychlejší než pomocí technologie Bluetooth nebo Wi-Fi.

NFC je zpětně kompatibilní s RFID, právě proto NFC pracuje rovněž na frekvenci 13,56 MHz.

#### 4.1.1.1 NFC zařízení

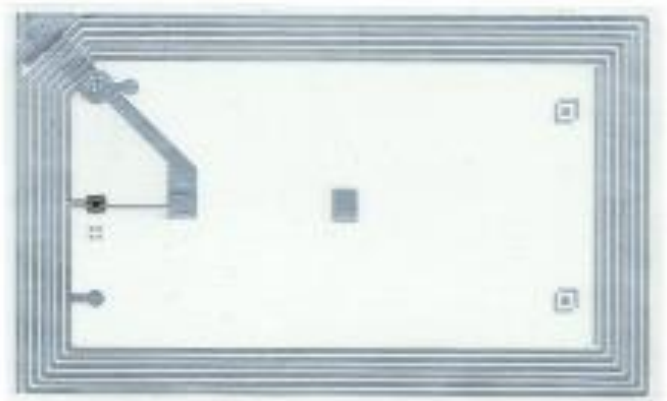
NFC zařízení se rozdělují do dvou skupin tzv. pasivní a tzv. aktivní. Hlavní rozdíl je především ve způsobu napájení (již poněkud méně ve schopnostech).

##### **Aktivní zařízení**

Aktivní zařízení např. mobilní telefon iniciuje spojení s druhým zařízením, které může být jak aktivní tak i pasivní. Aktivní se jmenuje, jelikož nejčastěji obsahuje vlastní zdroj napájení (a řídí komunikaci).

##### **Pasivní zařízení**

Pasivní NFC zařízení, většinou tzv. tag, nejčastěji nevyžaduje vlastní napájení, jelikož veškerá potřebná energie je dodána pomocí magnetické indukce při vybuzení tagu (v podstatě pouze odpovídají). Díky tomuto jsou pasivní tagy velice levné. Jak je vidět z obrázku 4.1, většina plochy NFC tagu je tvořena přijímací anténou sloužící nejen k přenosu dat, ale hlavně k dodání napájecí energie. Organizací NFC Forum jsou definovány 4 typy tagů lišící se přenosovou rychlostí, kapacitou interní paměti (i pro uživatelská data) nebo uzamykatelností této interní paměti, viz tabulka 4.1.



**Obrázek 4.1 : Pasivní NFC tag**

Typ	Možnost uzamčení	Maximální kapacita	Přenosová rychlost
1	Ano	2 kB	106 Kb/s
2	Ano	2 kB	106 Kb/s
3	Ne	1 MB	212 Kb/s
4	Ne	32 kB	106 Kb/s

**Tabulka 4.1 : Typy NFC tagů**

Pro zde navrhovaný navigační / lokalizační systém budou využity hlavně tzv. pasivní taky pro jejich velmi přijatelnou cenu.

#### 4.1.2 .NET Framework

.NET Framework je softwarová platforma vyvinutá společností Microsoft (kolem roku 2003) používána především v Microsoft Windows. Tento Framework obsahuje rozsáhnou knihovnu podpůrných objektů zvanou Framework Class Library (FCL) a poskytuje i celkem dobrou jazykovou interoperabilitu (v jakémkoliv programovacím jazyce může být použit kód z jiného programovacího jazyka). Program napsaný pro .NET je spuštěn ve virtualizovaném prostředí zvaném Common Language Runtime (CLR), představující v podstatě virtuální počítač a zajišťující služby jako zabezpečení, práce s pamětí a zpracovávání výjimek.

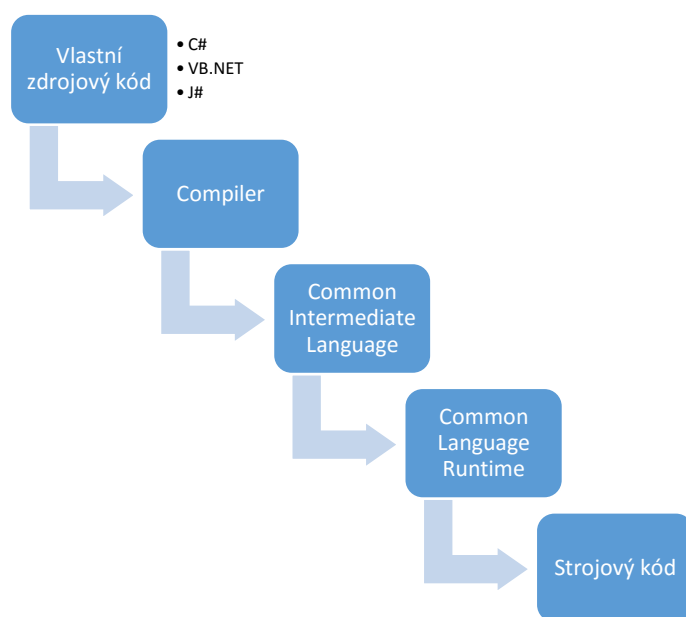
FCL poskytuje uživatelské rozhraní, práci se soubory, práci s databázemi, vývoj webových aplikací, numerické algoritmy a samozřejmě síťovou komunikaci. Výsledný program je kombinací FCL, vlastního zdrojového kódu a ostatních knihoven. .NET je používán především pro vývoj nových aplikací na operační systémy Windows.

##### 4.1.2.1 Architektura

#### **Common Language Infrastructure (CLI)**

CLI, poskytuje jazykově nezávislou platformu pro vývoj a běh programů zahrnující práci s pamětí, zpracovávání výjimek, bezpečnost a interoperabilitu. Tím že .NET Framework je implementován jako CLI, je tato funkcionalita dostupná ve všech programovacích jazycích podporovaných .NETem. Implementací CLI v .NET je CLR zaručující určité vlastnosti a chování jako již zmíněnou práci s pamětí, zpracovávání podmínek atd.

Aby mohl být program spuštěn na CLI, musí být nejdříve přeloženo do Common Intermediate Language (CIL), jedná se o kód určený pro CLR. Následně při skutečném spuštění aplikace je CIL kód zkompilován do strojového kódu cílového procesoru metodou just-in-time compiler (JIT). Kód je skutečně zkompilován až tehdy, když je potřeba, tedy až je opravdu spuštěn.



**Obrázek 4.2 Struktura Common Language Infrastructure**

## **Knihovny**

.NET Framework obsahuje sadu knihoven. Tyto knihovny implementují velké množství běžných funkcí jako čtení/zápis souborů, renderování grafických prvků, práce s databázemi, práce s XML soubory. Tyto knihovny jsou dostupné pro všechny jazyky podporované v .NET Framework. Knihovny v .NET se dělí do dvou skupin (bez jasného vymezení co do jaké skupiny patří) : Base Class Library (BCL) a Framework Class Library (FCL).

BCL je hlavním jádrem, které tvoří základní API (Application Programming Interface) pro CLR, a tvoří malou podmnožinu všech dostupných knihoven.

FLC obsahuje/odkazuje na celou knihovnu, která je součástí .NET Framework. Obsahuje rozšířený balík knihoven jako Windows Forms, Windows Presentation Foundation (WPF), atd.

### **4.1.2.2 C#**

C# nebo také „CSharp“ je vysokoúrovňový programovací jazyk, vyvinutý společně s .NET Frameworkem společností Microsoft a později standardizován. Vychází ze známých programovacích jazyků, jako jsou C++ a Java.

### **4.1.2.3 Windows Presentation Foundation (WPF)**

WPF je grafický subsystém .NETu umožňující snadnou tvorbu programů s grafickým rozhraním (GUI). Poprvé byl prezentován v .NETu V3.0, a je podporován operačním systémem Windows.

WPF vzniklo za účelem sjednocení práce s jednotlivými grafickými prvky jako 2D a 3D grafika, a pro ještě větší oddělení tvorby grafického rozhraní od logiky programu.

Výhody WPF:

- Oproti svému předchůdci s názvem Windows Forms, WPF využívá k vykreslování grafiky namísto GDI (Graphics Device Interface) technologii DirectX. To umožnilo využití složitějších grafických prvků a také přenést množství grafických úkonů přímo na GPU, čímž se snižuje zátěž procesoru (CPU)
- Zavedení DIP (Device Independent Pixel), tedy novou jednotku délky, zaručuje stejné zobrazení jak na obrazovce tabletu, tak i na obrazovce stolního počítače. Grafický formulář je popsán pomocí jazyka, založeném na XML, který se jmenuje XAML (Extensible Application Markup Language).
- Díky XAML je umožněn tzv. binding, tedy možnost napojit vlastnosti objektů na prvky formuláře.

## XAML

Díky úspěchu XML v oblasti návrhu webových stránek se tvůrci WPF rozhodli použít k popisu uživatelského rozhraní jazyk založený rovněž na něm. XAML umožňuje popsat vzhled a částečně i chování grafického uživatelského rozhraní s použitím minima procedurálního kódu. Dále použití XAML umožnilo oddělení modelu od zobrazovací vrstvy, což je považováno za dobrý návrhový princip.

### 4.1.2.4 Xamarin

Xamarin je platforma pro vývoj nativních aplikací na operační systémy Android, iOS a Windows Mobile. Aplikace jsou psány v jazyce C# (viz. kapitola 4.1.2.2). To umožňuje sdílení velkého množství kódu mezi aplikacemi.

Výše zmíněné platí zejména pro kód vytvářející logiku aplikace. Grafické rozhraní je již pro operační systém nutno poněkud upravit.

## 4.2 Návrh

Co si člověk představuje pod pojmem „lokalizace“ a „navigace“:

- Lokalizace – Schopnost určit pozici, na které se objekt aktuálně nachází.
- Navigace – Schopnost určit trasu z výchozí pozice do cílové pozice.
- Je tedy v celku jasné, že bez správné lokalizace nebude úspěšná navigace. Nejprve je tedy nutno určit aktuální pozici (samozřejmě libovolnou metodou) a poté lze plánovat cestu z této aktuální pozice do cílové pozice (rovněž libovolnou metodou).

Zde navrhovaný systém se tedy bude skládat ze dvou částí: Navigační a lokalizační. K lokalizaci budou použity NFC tagy rozmístěné na zřetelně vyznačených místech v budově, ze kterých se přečte unikátní identifikace, která je v mapě přiřazena k určité pozici. Díky těmto průběžným aktualizacím polohy lze uživatele navigovat po budově, kdy se ideálně s každou aktualizací polohy zobrazí aktualizovaná trasa.

Celé řešení by mělo být stručně rozděleno do tří částí:

- Aplikace pro kiosky (informační panely u vstupních bodů)
- Aplikaci pro mobilní zařízení sloužící k samotné navigaci.
- Soubor vhodně rozmístěných NFC tagů

## 4.2.1 Mapa

Mapa by měla být co nejjednodušší, mít jasnou a pevnou strukturu pro nejen snadné vytváření, ale i pozdější úpravu. Navrhovaná mapa se bude skládat z jednoho univerzálního základního prvku sloužícího pro tvorbu větších prvků.

### 4.2.1.1 Blok mapy

Tento základní blok mapy (dále blok), si lze představit jako čtvereček (3D krychle) v mapě. Blok by měl mít následující vlastnosti:

- Poloha v mapě (tři souřadnice X, Y, Z)
- Informace o zdech (zdali jsou na hranách bloku, nebo jestli to jsou dveře)
- Na které zdi je tag (pokud je v tomto bloku umístěn)
- Kdo je nadřazený objekt tomuto bloku
- Všechny okolní bloky

### 4.2.1.2 Objekt mapy

Dále by v mapě měly být objekty skládající se z dvou a více bloků. Takovýto objekt by mohl v budově reprezentovat jednotlivé místnosti, chodby, schodiště, výtahy atd. Rovněž by měl obsahovat informace o jeho účelu a dostatečný textový informační popis pro uživatele.

### 4.2.1.3 NFC tagy v mapě

NFC tagy by měli pracovat dvojitým způsobem

- V souborech mapy se nachází samostatný soubor obsahující identifikace vysílané z rozmístěných tagů a k nim příslušnou pozici v mapě, kde se tento tag nachází. Jde o tagy s pevně určeným obsahem.
- Anebo samotné tagy budou mít ve své identifikaci (zpráva z interní paměti) přímo uloženou pozici v mapě kde se nacházejí. Jde o tagy s možností vložení vlastní informace.

První řešení je dobré pro možnost zobrazovat pozici NFC tagů přímo v navigačním systému / aplikaci, ale není úplně snadné přidat nový tag nebo změnit jeho pozici. V tomto případě je nutno změnit konfigurační soubor systému. Kdežto u druhého řešení je velmi snadná rozšiřitelnost, ale je zapotřebí mít tzv. zapisovatelné NFC tagy do kterých se vloží informace o jejich pozici. Takto velmi snadno přidané tagy se však nezobrazují v mapě (mapa o nich samozřejmě neví).

Avšak vhodnou kombinací obou těchto způsobů lze dosáhnout velmi adaptivního a rozšiřitelného navigačního systému. Nejprve lze rozmístit určitý počet základních tagů a ty do systému, tedy mapy, zanést. Pokud bude později zjištěn tento počet jako nedostatečný, lze velmi snadno a rychle doplnit sledovaný prostor / budovu a další tagy obsahující přímo informace o svém umístění, ale nezobrazující se v mapě.

## 4.2.2 Grafické rozhraní

Grafické rozhraní by mělo být co nejjednodušší. Hlavní obrazovka aplikace „kiosky“ by se měla skládat z těchto částí:

- Hlavním prvkem rozhraní je interaktivní mapa zobrazená ve 3D.
- Na pravé straně obrazovky se bude nacházet postranní panel, ve kterém si uživatel bude moct vyhledat místnost podle jejího typu.
- Spodní lišta s tlačítkem pro návrat do předchozího stavu aplikace.

#### 4.2.2.1 Interaktivní mapa

Mapa by se měla přizpůsobovat kontextu, podle toho co uživatel potřebuje aktuálně vědět. Pokud uživatel klikne na nějaký objekt v mapě (místnost, chodba), tak se daný objekt zvýrazní a zobrazí se o něm dostupné informace. Tato stejná akce se vykoná i v případě, kdy uživatel vyhledá místnost, do které chce jít, přes postranní panel.

#### 4.2.2.2 Podpora více jazyků

U systému, který má navigovat po veřejných prostorech by měla být rovněž možnost přepínat mezi více jazyky, minimálně mezi úředním jazykem dané země a angličtinou (případně i další jazyky). Tato možnost musí být dostupná za běhu programu bez nutnosti jeho restartování.

## Kapitola 5 Implementace (Řešení)

---

Výsledkem implementace je spustitelný soubor typu EXE, tedy program pro kiosky. Dále knihovna přijímající a dekodující identifikace z NFC tagů. Součástí aplikace je rovněž mapa budovy.

Základní prvkem mapy je blok, který si lze představit jako kvádr mající svou pevnou pozici v mapě danou souřadnicemi X, Y a Z viz Obrázek 5.1. Z několika těchto základních objektů se skládá tzv. objekt mapy, ten si lze představit jako místnost nebo chodbu v budově.

### 5.1 Mapa

Jak se při implementaci mapy ukázalo, je zapotřebí veliké propojenosti jednotlivých objektů mapy, aby se v mapě snadno vyhledávalo.

#### 5.1.1 Blok mapy

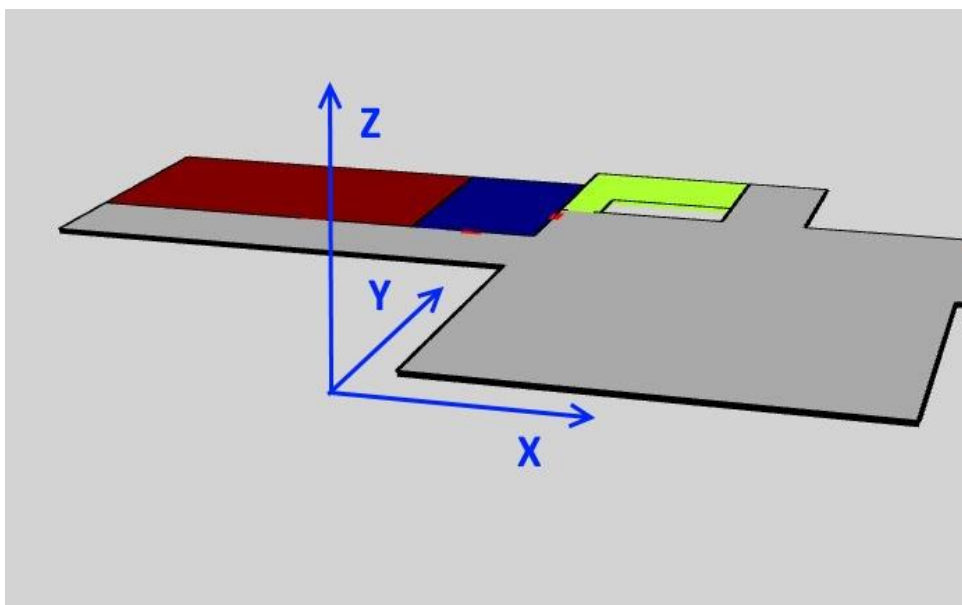
Jako základní stavební prvek mapy používám třídu „`MapItem`“ obsahující zejména informaci o všech šesti stěnách krychle. Tato informace je uložena jako enum s názvem „`EnumWall`“.

```
public enum EnumWall
{
    None, Door, Wall, Window //nic, dveře, zeď, okno
}
```

Každý „`MapItem`“ obsahuje stěny s tímto umístěním:

- A – stěna ve směru osy -X
- B – stěna ve směru osy +Y
- C – stěna ve směru osy +X
- D – stěna ve směru osy -Y
- E (spodní), F (horní) – zatím nejsou využity

Dále jsou obsaženy souřadnice pozice v podobě tří proměnných X, Y a Z typu integer (využívané ve významu: poloha v ose X, poloha v ose Y a patro; viz Obrázek 5.1). Tyto tři proměnné se dají zapsat i pomocí property `Position`, která je typu pole integeru, pro snadnější vkládání pozice, díky tomu můžu všechny tři souřadnice nastavit najednou.



Obrázek 5.1 Vyobrazení použitých souřadnic

```

public int[] Position
{
    get { return new int[] { X, Y, Z }; } //Vytváří novou instanci
    pole, které je naplněno proměnnými X, Y a Z
    set
    {
        if (value.Length == 3) // Pokud je vloženo pole o 3 prvcích,
            provede se tělo podmínky
        {
            this.X = value[0];
            this.Y = value[1];
            this.Z = value[2];
        }
    }
}

```

Blok mapy také obsahuje List se všemi sousedícími bloky. Toto pole je použito k vyhledání optimální cesty (viz kapitola 5.6). Dále zahrnuje odkaz na objekt mapy (kapitola 5.1.2) obsahující tento blok.

### 5.1.2 Objekt mapy

Objekt mapy tvoří třída „`MapObject`“ obsahující text reprezentující název objektu, List všech bloků mapy tvořící daný objekt mapy a popis objektu pro uživatele, tedy informace o daném objektu. Reprezentován Listem instancí třídy „`LangLines`“.

Popis objektu je tvořen výše zmíněnou třídou „`LangLines`“ a třídou „`Line`“. Kde třída „`LangLines`“ obsahuje List objektů typu „`Line`“ a vlastnost „`Language`“ typu string určující



jící v jakém jazyce je text uveden. To umožňuje snadno psát popis ve více jazycích současně.

Třída „Line“ reprezentuje jednu řádku pro popis objektu mapy a obsahuje samotný text jedné řádky, barvu pomocí tří složek R, G a B, které reprezentují intenzitu jednotlivých základních barev (červená, zelená a modrá), ze kterých je složena výsledná barva, a proměnou „Size“ určující velikost písma pro daný řádek textu (možnost některou informaci vyzdvihnout).

Objekt mapy dále obsahuje dvě funkce. První funkce s názvem „FromXML“ se volá po okamžitě načtení objektu ze souboru XML (viz kapitola 5.5) pro inicializaci prvků, jež nejsou přímo uloženy v souboru, ale dají se zjistit z kontextu.

```
public virtual void FromXML(Building b)
{
    foreach (MapItem item in Items)
    {
        item.owner = this; //Bloku mapy se nastaví jeho nadřazený objekt
        b.Map[item.X, item.Y, item.Z] = item; // Objekt se přidá na svou
        //pozici do trojrozměrného pole budovy
        isBorder(item);
    }
}
```

Druhou funkcí je „isBorder“ volanou na všechny bloky tvořící objekt mapy. Funkce zjistí za pomoci znalosti okolí, zdali je na hraně bloku zeď či nikoliv.

### 5.1.2.1 Potomci objektu mapy

Objekt mapy má tři potomky tvořící základní objekty v budově.

- Třída „Room“ představuje místnost v budově. Oproti objektu má navíc vstup, přepisuje funkci „FromXML“ kterou rozšíří o načtení odkazu na blok, ze kterého se do místnosti vchází, a metodu „isDoor“, která podle polohy vstupu označí příslušnou stěnu jako dveře.
- Třída „Hall“ reprezentuje chodu. Navíc má funkci „doEntrances(Building b)“ sloužící k odstranění zdí mezi chodbami kvůli jejich pozdějšímu zobrazení.
- Třída „Stairway“ popisuje schodiště / výtahy, tedy objekty sloužící k přemísťování mezi patry.

### 5.1.3 Základní třída mapy

Základní třída mapy s názvem „Building“ („Budova“) obsahuje zejména tyto položky:

- název budovy – textový popis
- rozměr mapy – počet bloků tvořící budovu v každé ose (X,Y,Z)
- trojrozměrné pole bloků mapy (viz kapitola 4.2.1.1 a 5.1.1), které reprezentuje celou budovu
- Dictionary (slovník) poskytující snadný přístup k jednotlivým objektům mapy za podle typu (např. obchod s potravinami, s oblečením, ...) objektu a jeho názvu.

## 5.2 Práce s NFC

Pro čtení identifikace z NFC tagů bylo vyzkoušeno několik doporučených postupů získaných nejčastěji jako ukázky z WWW projektů. Některé však pracovali pouze na desktopových Windows a jiné naopak pouze na mobilních Windows. Po několika poloúspěšných pokusech, byl vytvořen a odladěn kód, za částečného využití tzv. Universal Windows Platform, což jsou aplikace vytvořené cíleně pro WIN10, avšak pracující jak na desktopových, tak i mobilních Windows. Tato výsledná verze byla použita ve vytvářené aplikaci a je tedy popsána dále.

Mnou vytvořené řešení se (v současné době) neskládá ze dvou různých aplikací pro kiosky (informační panely) a pro mobilní telefon. Namísto toho je vytvořena pouze jedna aplikace spustitelná jak na kiosku (lokálním počítači), tak na mobilním tabletu (mající klasický desktopový procesor s architekturou x86/x64). Na tabletu je navíc použita knihovna s kódem pro čtení z NFC tagů a předávání jejich přečtené identifikace do hlavní aplikace. NFC senzor (sloužící pro komunikaci s NFC tagem) je inicializován následovně:

```
// získání defaultního NFC senzoru v zařízení
_proximityDevice = ProximityDevice.GetDefault();
// test zda je NFC sensor dostupný
if (_proximityDevice != null)
{
    //Nastavení typu požadované zpravy a nastavení handler pro příjem
    _MessageType = _proximityDevice.SubscribeForMessage("WindowsUri",
        MessageReceivedHandler);
}
}
```

Následné vyzvednutí identifikace z tagu:

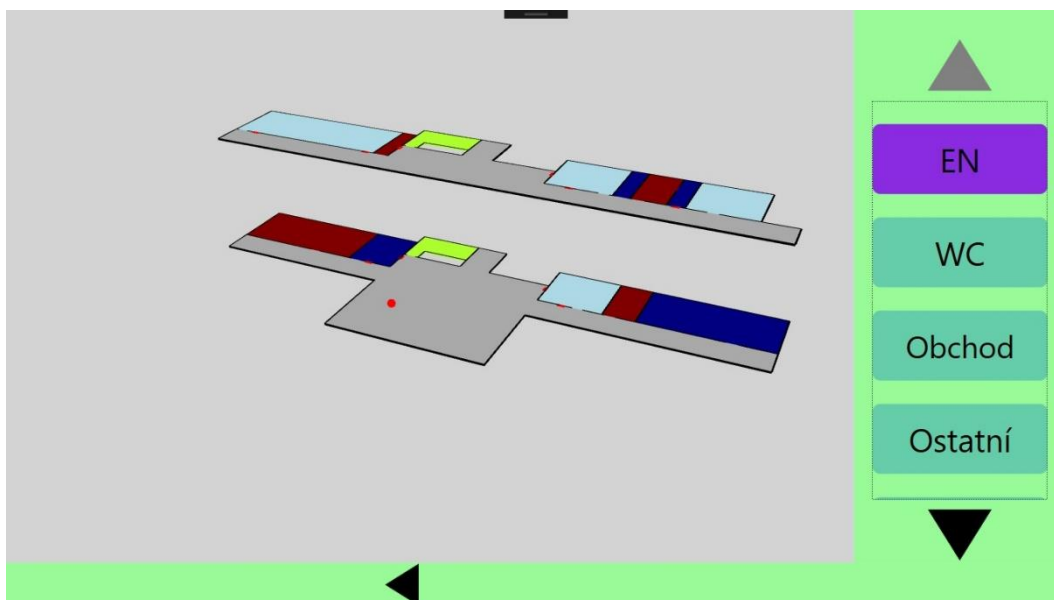
```
try
{
    // přečtení surových dat, obdržených z NFC tagu
    using (var reader = DataReader.FromBuffer(message.Data))
    {
        // nastavení potřebného kodování
        reader.UnicodeEncoding =
            Windows.Storage.Streams.UnicodeEncoding.Utf16LE;
        // vyzvednutí zpravy jako čitelný text
        string receivedString = reader.ReadString(
            reader.UnconsumedBufferLength / 2 - 1);
        //Pokud existuje zavola se obsluha, která zprávu dále zpracuje
        if (OnMessageNFC != null) { OnMessageNFC(receivedString); }
    }
}
catch (Exception e)
{
    Debug.WriteLine(e.StackTrace);
}
```

Hlavní aplikace přijímá událost „OnMessageNFC“ a při její aktivaci, zpracuje ident vy-zvednutý z NFC tagu.

### 5.3 Grafické rozhraní (GUI)

Základní rozložení grafického rozhraní aplikace jsem vytvořil za pomoci jazyka XAML (použitého v .NET Framework) (kapitola 4.1.2.3).

GUI se skládá ze dvou hlavních prvků: mapy a postranního panelu. Mapa je pro lepší interakci zobrazena ve 3D a lze s ní, pomocí dotyků prstů po obrazovce, hýbat, natáčet a i vybírat různé místnosti, o který se zobrazí informační panel s možností vyhledání optimální cesty. Místnosti lze ale i vyhledat pomocí postranního panelu, kde jsou zobrazeny jejich jednotlivé typy. Podle zvoleného typu místnosti lze zobrazit všechny místnosti požadovaného typu a uživatel si může cílovou místnost vybrat.



Obrázek 5.2 Náhled na celé grafické prostředí

#### 5.3.1 Postranní panel

Postranní panel je tvořen prvkem ScrollView umožňující snadno rolovat jeho obsahem nahoru / dolů. Jeho obsah je prvek Grid naplněný potřebnými tlačítky. Toto naplnění probíhá již v kódu aplikace obsahujícím popisovanou logiku programu. Tlačítka se tedy generují dynamicky podle obsahu mapy.

##### 5.3.1.1 Tlačítka

GUI bylo navrženo s ohledem na ovládání pouze dotykem prstu po ploše obrazovky. Jeho grafické / zobrazené prvky musí být tedy dostatečně velké pro snadné ovládání i bez přehmatů. Kvůli přehmatům jsem implementoval určité časové zpoždění, se kterým tlačítka

reagují na stisk, aby nedocházelo k nechtěným stiskům tlačítek. Tlačítko je tedy nutno podržet nějaký minimální čas. Tohoto jsem dosáhl za použití „DispatcherTimer“ (časovač vhodný pro práci s GUI) a skutečnosti, že mají všechny grafické prvky vlastnost nazvanou „Tag“ typu „object“. Do lze uložit odkaz na libovolný objekt, jelikož všechny třídy jsou automaticky potomci třídy/rozhraní „object“.

Tedy při stisku jakéhokoliv tlačítka potřebuji spustit „DispatcherTimer“, a poté zavolat vždy tu samou obsluhu. Rozhodl jsem se využít výše zmiňované vlastnosti „Tag“ (od „DispatcherTimer“) k uložení základních informací vztahujících se k příslušnému tlačítku a rovněž k uložení odkazu na obsluhu tlačítka, zvanou po detekci úspěšného stisku. Pro tento účel jsem vytvořil třídu s názvem „MyTag“.

„MyTag“ obsahuje proměnou „pushed“ typu `bool` k indikaci, zdali bylo tlačítko stisknuto minimálně požadovanou dobu a odkaz na obslužnou funkci tlačítka.

Detekce stisku tlačítka probíhá takto: při spuštění programu se nastaví perioda „DispatcherTimeru“ a nastaví se obsluha jeho přetečení. Při stisku tlačítka (TouchDown) se zavolá následující obsluha:

```
private void TouchDownHandler(object sender, MouseButtonEventArgs e)
{
    buttonTimer.Tag = sender; //Do tagu timeru se ulozi odkaz na tlacitko
    if (!buttonTimer.IsEnabled) //Pokud je timer zastaven
    {
        buttonTimer.Start(); //spusteni timeru
    }
}
```

Kde `buttonTimer` je instance „DispatcherTimer“. Pokud je tlačítko stisknuté déle než perioda časovače, zavolá se obsluha přetečení časovače mající následující kód:

```
private void TickHandler(object sender, EventArgs e)
{
    // vyzvednutí volajícího objektu
    DispatcherTimer timer = (sender as DispatcherTimer);
    timer.Stop(); //Timer je zastaven
    // vyzvednutí tlačítka z tagu timeru
    Border border = timer.Tag as Border;
    //Zaznamená se, že tlačítko bylo stisknuto
    //po dostatečně dlouhou dobu
    (border.Tag as MyTag).Pushed = true;
    //Pozadí tlačítka se obarví na černou a text na bílou barvu,
    //k detekci stisknutého tlačítka
    ((border.Child as Viewbox).Child as TextBlock).Foreground =
        Brushes.White;
    border.Background = Brushes.Black; }
}
```

Když je tlačítko uvolněno (TouchUp), zavolá se příslušná obsluha. Pokud bylo detekováno dostatečně dlouhé stisknutí tlačítka (přetečení časovače), zavolá se rovněž jeho obsluha, na kterou je odkazováno v tagu tlačítka viz:

```

private void TouchUpHandler(object sender, MouseButtonEventArgs e)
{
    buttonTimer.Stop(); //zastaveni timeru
    Border border = sender as Border; // pretypovani tlacitka
    MyTag tag = border.Tag as MyTag; // pretypovani tagu
    // Vrati barvy tlacitka do puvodniho stavu
    ((border.Child as Viewbox).Child as TextBlock).Foreground =
        Brushes.Black;

    border.Background = tag.OriginalColor;
    //pokud neni tag tlacitka null a tag timeru je stejný jako sender
    //a bylo detekováno stisknutí a hendler není null
    if (buttonTimer.Tag != null && buttonTimer.Tag.Equals(sender)
        && tag.Pushed && tag.Handler != null)
    {
        buttonTimer.Tag = null;
        tag.Handler(sender, e); //Zavolani obsluhy tlacitka
    }
    tag.Pushed = false;
}

```

To co tlačítko skutečně vykoná je uloženo ve vlastnosti `Tag` tohoto tlačítka. To se samozřejmě vykoná pouze tehdy, je-li tlačítko stisknuto minimálně po dobu periody přetečení časovače.

### 5.3.1.2 ScrollViewer

Jelikož v budově může být velké množství typů místností (obchody, chodby, toalety, ...), zobrazující se na postranním panelu jako tlačítka, použil jsem na vkládání těchto tlačítek prvek „`ScrollViewer`“ umožňující vytvořit libovolně dlouhý sloupec těchto tlačítek a posouvat s nimi ve vertikálním směru.

Jelikož se program bude ovládat zejména dotykem, tak aby toto bylo zcela zřetelné, že tlačítka na postranním panelu se mohou posouvat ve vertikálním směru, přidal jsem ještě dvě názorná tlačítka umožňující posun tlačítek ve „`ScrollViewer`“ i pomocí těchto dvou zmíněných dodatečných tlačítek.

### 5.3.2 3D zobrazení mapy

Pro vytvoření a zobrazení mapy ve 3D jsem použil prvek „`Viewport3D`“ (využívající `DirectX`). Jednotlivé „`MapItem`“ jsou reprezentovány jako kvádr ve 3D světě. Všechny bloky jednoho objektu mapy (například místnost) jsou uloženy v instanci třídy „`Model3DGroup`“. Ten se poté vloží do objektu „`ModelUIElement3D`“, zajišťující potřebnou interaktivitu mapy viz. kapitola 5.3.2.1. Dále je ještě vložen do objektu „`ModelVisual3D`“ pro skutečné zobrazení ve 3D scéně a nakonec rovněž vložen do „`Viewport3D`“.

Funkce pracující s 3D objekty jsou uloženy v souboru `Utils.Graphics3DUtils01`. Zde jsou vytvořeny základní statické třídy a metody vytvářející grafické 3D prostorové objekty.

#### 5.3.2.1 Interaktivita mapy

Interaktivitu bylo dosaženo za pomoci prvku „`ModelUIElement3D`“. Ten vhodně poskytuje událost při dotyku / kliku na libovolný zobrazený 3D objekt.

Další možností, jak zlepšit ovládání mapy je její posouvání / rotování pomocí pohybu prstu po ploše obrazovky. Toho jsem dosáhl pomocí třídy „`Graphics3DCamera`“ obsahující doplňující schopnosti pro pohyb kamerou.

## 5.4 Podpora více národnostních jazyků

Všechny prvky „`TextBlock`“ mají své pojmenování vytvořené jako unikátní řetězec. Ten je v každém jazyce svázán s lokalizovaným řetězcem. Pokud v právě použité lokalizaci chybí nějaký text, zobrazí se varovná zpráva, a místo lokalizovaného řetězce se použije tento unikátní řetězec v hranatých závorkách, např. pokud je unikátní řetězec „`auto`“ použije se „`[auto]`“. Tento stav velmi snadno signalizuje chybějící text v daném jazyce.

Pro snadnou manipulaci s jazyky jsem vytvořil třídu reprezentující jazyk. Tato třída obsahuje název jazyka a List objektů typu „`Message`“, reprezentující dvojici složenou z unikátního názvu / klíče a k němu přiřazeného lokalizačního řetězce. K získání lokalizovaného řetězce třída obsahuje funkci „`GetText`“ vracející požadovaný text podle zadaného unikátního řetězce, viz. příklad:

```
public string GetText(string tag)
{
    try
    {
        //Pokud vyhledavani pomoci unikatniho retezce
        return Messages.Find(Item => Item.Tag == tag).Text;
    }
    catch
    {
        //Pokud není unikatni retezec nalezen v danem jazyce a odchyti
        //se vyjimka vrati se unikatni retezec s v hranatych zavorkach
        return "[" + tag + "];"
    }
}
```

Jednotlivé jazyky jsou spravovány zastřešující třídou „`LangManager`“ starající se o načtený jazyk. Ta rovněž umí přepínat jazyky. Dále umožňuje poskytnout pro několik různých prvků typu `TextBlock`ů stejný text. Tohoto je dosaženo odstraním čísla z konce jejich unikátních názvů (ve WPF nelze pojmenovat dva prvky zcela stejným názvem). Jak můžeme vidět na příkladu:

```
public string getString(string Name)
{
    if (Name != null)
    {
        // odstrani z konce tagu cisla
        Name = Name.TrimEnd('0', '1', '2', '3', '4', '5', '6', '7', '8', '9');
        //Kontroluje se pokud se podarilo nacist jazyk
        if (current != null)
        {
            //Vrati se pozadovany retezec
            return current.GetText(Name);
        }
    }
}
```

```

    }
    else
    {
        //Pokud se nepodarilo nacist pozadovany jazyk
        //pouzije se anglictina
        english.GetText(Name);
    }
}
return null;
}

```

I když projekty vytvářené pomocí MS Visual Studio poskytují podporu pro vícejazyčné aplikace, jde o textové soubory začleněné do projektu nebo o soubory vcelku složité pro externí editaci. Z tohoto důvodu bylo přistoupeno k výše uvedenému řešení.

## 5.5 Ukládání do XML

K ukládání objektů používám statickou třídu „XUtils.Xml.XmlSerialization“ využívající „XmlSerializer“. Příklady uložení tříd do XML:

### Objekt mapy (odstavce 4.2.1.2 a 5.1.2)

```

<MapObject xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xsi:type="Hall" Type="Hall"
Name="C3">
  <Items>
    <Item Main="None" WallA="Wall" WallB="None" WallC="None"
WallD="Wall" Ceiling="None" floor="None" X="0" Y="6" Z="0" />
  </Items>
<LangLines>
  <LangLines Langlanguage="cs">
    <Lines>
      <Line R="0" G="0" B="255" Size="30" Text="Text cesky" />
    </Lines>
  </LangLines>
  <LangLines Langlanguage="en">
    <Lines>
      <Line R="0" G="0" B="255" Size="30" Text="Text anglicky" />
    </Lines>
  </LangLines>
</LangLines>
</MapObject>

```

### Jazyk (odstavec 5.4)

```

<Lang xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <Messages>
    <Messages Tag="TEST" Text="Test_CZ" />
  </Messages>
  <Name>cs</Name>
</Lang>

```

## 5.6 Algoritmus pro nalezení optimální cesty

K nalezení cesty do cíle používám algoritmus uspořádaného prohledávání stavového prostoru A\*(A Star). Když se algoritmus rozhoduje, do kterého prvku půjde dál, nejprve vyhodnotí všechny dostupné možnosti funkcí

$$f(n) = g(n) + h(n) \quad (1)$$

Kde  $g(n)$  představuje cenu již uplatněnou pro dosažení bodu  $n$ .  $h(n)$  naopak představuje odhad ceny potřebné pro dosažení z bodu  $n$  do cíle. Cenou je v mém případě vzdálenost v podobě bloků mapy. Aby algoritmus pracoval správně musí být odhad ceny do cíle  $h(n)$  co neblíží přesné hodnotě. Pro  $h(n)$  musí platit

$$\forall n : 0 \leq h(n) \leq h^*(n) \quad (2)$$

Kde  $h^*(n)$  je přesná hodnota ceny (vzdálenosti) do cíle. Všechny takto hodnocené prvky se ukládají do prioritního listu seřazeného vždy tak, aby prvek s nejmenší hodnotící funkcí byl navrchu.

Po zhodnocení všech okolních prvků a jejich přidání do listu, se vezme prvek na vrcholu listu, tedy ten s nejmenší hodnotící funkcí. Takto se pokračuje, dokud algoritmus nedosáhne cíle.

Abych algoritmus mohl implementovat, potřebuji, aby každý blok mapy obsahoval odkazy na všechny jeho okolní bloky. Poté algoritmus bude vybírat a dostávat samotné bloky mapy, ze kterých vytvoří cestu



## Kapitola 6 Používání aplikace

Výsledný systém byl navržen pro nejjednodušeji využívání a to jak uživatelem návštěvníkem, tak i správcem systému.

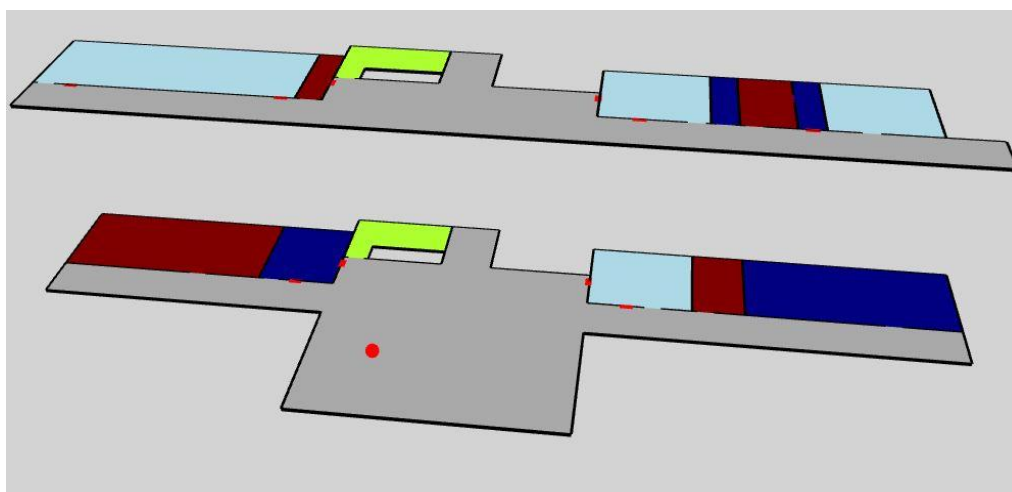
### 6.1 Ovládání kiosku (informačního panelu)

Na obrazovce se nachází dva ovládací prvky aplikace:

- Mapa - zobrazena ve 3D
- Postranní panel – lze vybrat místnost podle jejího typu

#### 6.1.1 Mapa

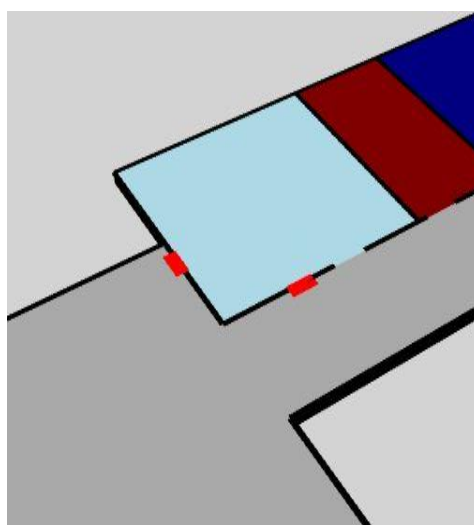
Aktuální pozice, nebo pozice informačního panelu je vyobrazen pomocí koule červené barvy, viz. Obrázek 6.1.



Obrázek 6.1 3D mapa s vyznačenou pozicí kiosku

NFC tagy rozmístěné po budově jsou, v mapě, vyobrazeny červenou značkou viz Obrázek 6.2.

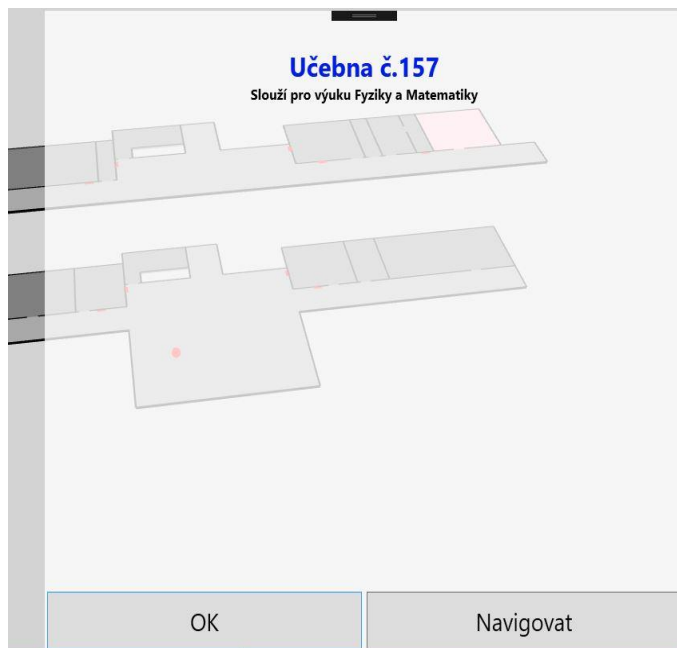
K lepší orientaci v mapě, lze s mapou posouvat intuitivním přetažením prstu po mapě (položít prst na mapu a táhnout požadovaným směrem). Pro získání informací o místnosti lze kliknout na každou místnost, tím ostatní místnosti poněkud zešednou a vybraná místnost je zvýrazněna, také se zobrazí dialog s informacemi o místnosti společně (viz Obrázek 6.3) s tlačítkem pro vyhledání optimální cesty.



Obrázek 6.2 Zvýraznění tagu v mapě

### 6.1.2 Postranní panel

V horní části postranního panelu se nachází tlačítko na přepínání národnostního jazyka. Je možné přepínat mezi češtinou a angličtinou, tedy pokud je nastaven český jazyk, na tlačítku je zobrazen nápis EN, a pokud je nastavena angličtina, na tlačítku je CZ. V případě použití více jazyků se na tomto tlačítku postupně zobrazují zkratky dostupných jazyků. Tento styl nemusí být z hlediska rychlosti přepínání jazyků zcela ideální, avšak do pěti jazyků je dostačující. V případě použití více jazyků bude vhodnější nějaké zobrazení formou seznamu.



**Obrázek 6.3 Okno s informacemi o místnosti**

Dále postranní panel obsahuje sadu tlačítek reprezentující jednotlivé typy místností. Pokud je typů místností více než je kapacita postranního panelu, je možné s těmito prvky „rolovat“, a to intuitivně přetažením prstu po tlačítcích, nebo pomocí tlačítek se šipkami nahoru respektive dolů. Po vybrání typu místnosti se v mapě zvýrazní všechny místnosti požadovaného typu. Když si uživatel vybere konkrétní místnost zobrazí se dialog s informacemi o místnosti stejně jako v kapitole 6.1.1.

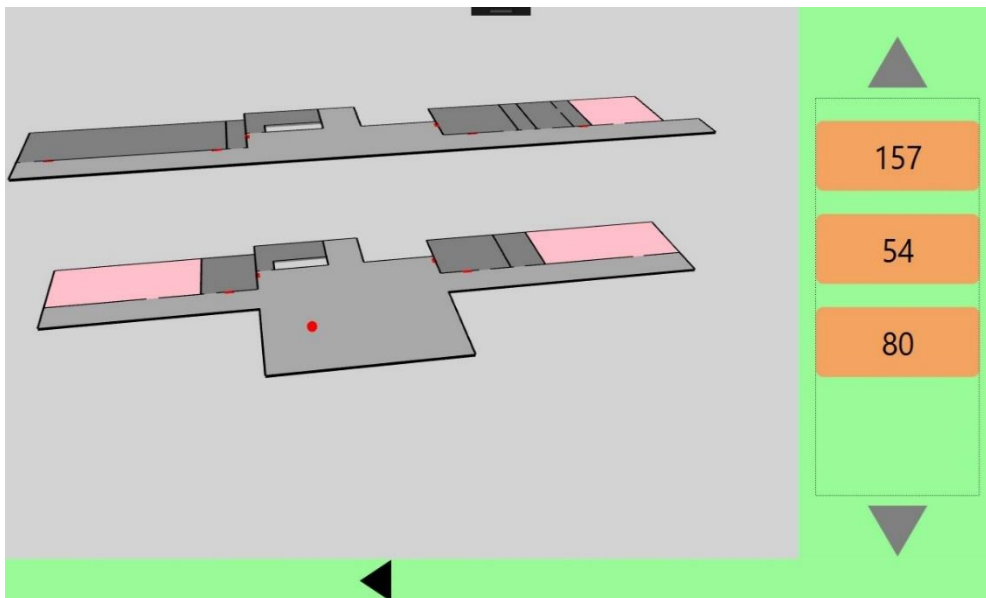
### 6.1.3 Navigace a lokalizace po budově

Po vybrání cílové lokality (místnosti) algoritmus najde optimální cestu (viz. kapitola 5.6) a zobrazí ji v mapě viz. Obrázek 6.5.

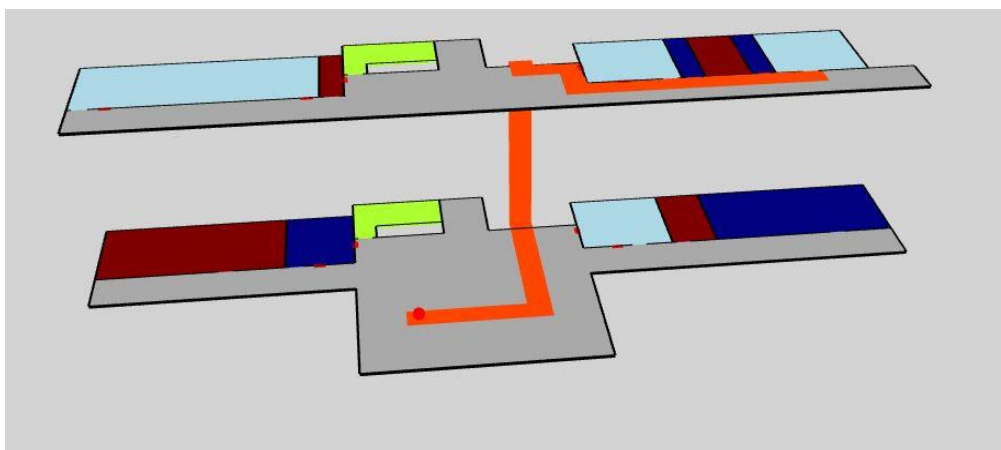
K určení aktuální polohy jsou využity NFC tagy rozmístěné po budově na zvýrazněných místech s nápisem informujícím o přítomnosti lokalizačního tagu. Po přiložení mobilního zařízení k některému tagu se aktualizuje pozice, a navigace pokračuje od této aktuální pozice (viz. Obrázek 6.6).

Navigaci lze ukončit dvěma způsoby:

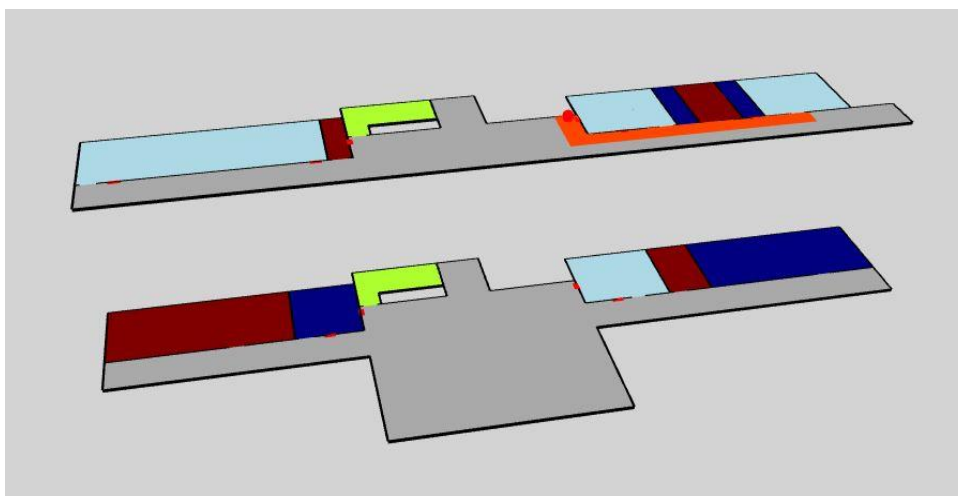
- Ukončení uživatelem – stisknutí tlačítka pro ukončení navigace
- Přečtení ke konečnému tagu – uživatel přiloží zařízení k tagu u cílové místnosti (pokud je tam umístěn).



Obrázek 6.4 Při vybrání určitého typu místnosti se všechny místnosti tohoto typu



Obrázek 6.5 Zobrazení nově vytvořené cesty

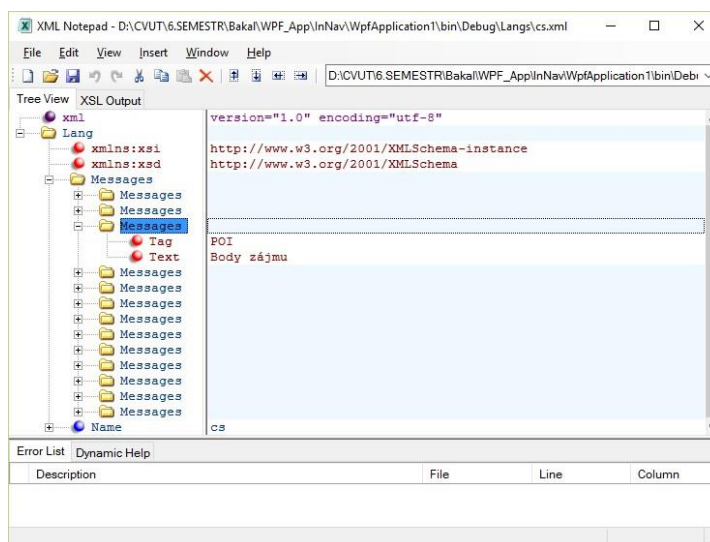


Obrázek 6.6 Aktualizovaná pozice pomocí tagu

## 6.2 Správa systému

System je navržen tak, aby jeho náročnost na HW výkon byla co nejmenší a mohl běžet na libovolném mini-PC<sup>3</sup> o malých rozměrech a možnosti připojení zařízení k HDMI portu.

K údržbě a rozšíření není zapotřebí znalosti žádného programovacího jazyku, stačí pouze znalost XML (v současné době není vytvořen speciální XML editor pro tuto potřeby). K tvorbě a editaci XML souboru lze celkem vhodně využít program od firmy Microsoft s názvem „XML Notepad 2007“<sup>4</sup> poskytující jednoduché a přehledné grafické prostředí pro editaci těchto typů souborů.



Obrázek 6.7 Ukázka programu XML Notepad

<sup>3</sup> Například: [Intel Compute Stick Sterling City Atom 32GB Windows](http://www.intel.com/products/computestick/sterlingcity/atom32gb/windows/)

<sup>4</sup> Lze stáhnout: <https://www.microsoft.com/en-us/download/details.aspx?id=7973>

## 6.2.1 Editace souborů mapy

Cesta ke kořenovému adresáři mapy je „umístění\_spustitelného\_souboru\Building\Název\_budovy“. V tomto adresáři se nachází soubor se základními informacemi o budově:

- Její velikost – v počtu základních bloků v jednotlivých osách budovy
- Seznam tagů – obsahující jejich pozice v mapě a na jaké zdi se nachází

Krom tohoto souboru složka obsahuje podsložky se soubory s uloženými objekty. Pro každý typ objektu jedna podsložka a pro každý objekt daného typu jeden soubor s názvem tohoto objektu. K tomuto způsobu se přistoupilo z bezpečnostních důvodů, aby při poruše jednoho souboru nedošlo ke ztrátě všech dat. Kořenový adresář může (v současné době) obsahovat až tyto čtyři podsložky:

- Hall – obsahující soubory s uloženými chodbami
- Room – obsahuje uložené místnosti
- Stairway – obsahuje uložené schodiště
- WC – obsahuje uložené toalety

Například pokud chcete upravit místnost s názvem „Místnost\_C3“, musíte otevřít a upravit soubor „umístění\_spustitelného\_souboru\Building\Název\_budovy\Room\Místnost\_C3.xml“.

### Jak přidat místnost

Každá podsložka obsahuje vzorový soubor s uvedením jeho doporučeného obsahu. Např. jak přidat místnost:

```
<?xml version="1.0" encoding="utf-8"?>
<MapObject xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xsi:type="Room" Type="Room"
<!-- Opsat u všech stejne -->
Name="80" EntrX="32" EntrY="7" EntrZ="0">
  <!-- Nazev budovy a souradnice vstupu -->
  <Items>
    <Item Main="None" WallA="Wall" WallB="None" WallC="None" WallD="Wall"
Ceiling="None" flor="None" X="27" Y="8" Z="0" />
    <!--Popis jendoho prvku mapy, u zdi muze byt vsude "None"
                                     Souradnice pozice prvku -->
  </Items>
  <LangLines>
    <LangLines Langlanguage="cs">
      <Lines>
        <Line R="0" G="0" B="255" Size="30" Text="Místnost C3" />
        <!-- Nastavení barvy textu| velikost textu a text samotny -->
      </Lines>
    </LangLines>
    <!-- Konec textu pro dany jazyk -->
  </LangLines>
</MapObject>
```

### 6.2.2 Editace národnostních jazyků

Všechny soubory s uloženými jazykovými texty jsou ve složce „umístění\_spustitelného\_souboru\Langs\“. K vytvoření nové lokalizace je doporučeno použít jako referenční soubor anglickou lokalizaci. V takovémto souboru pak lze pouze přeložit texty, a uložit jako nový soubor (s názvem podle zkratky jazyka).

## Kapitola 7 Testování

---

Jelikož se jedná o experimentální a demonstrační řešení, je jeho ověření nebo dokonce testování poněkud omezené. Celkový systém se skládá z několika dílčích částí, a tudíž se testování zaměřilo zejména na tyto jednotlivé části.

### 7.1 Aplikace pro kiosek

Vhodnost aplikace určené pro tzv. kiosek. Aplikace pro kiosek byla vytvořena se záměrem minimálních požadavků na výkon použitého počítače. Tato aplikace byla ověřena na komponentech:

- Běžný stolní PC (Core 2 Duo 2.5GHz) s připojeným 24 palcovým dotykovým monitorem. Kancelářské vybavení.
- Mini-PC ve velikosti USB FLASHky (MeeGo Pad T01 s Windows 10) zasunutý do HDMI vstupu zobrazovacího zařízení s připojenou 32 palcovou dotykovou TV. Vybavení v CAT (Centrum Asistivních Technologii) v Dejvicích.

V obou případech se jedná o vcelku levné a méně výkonné počítače. Aplikace na obou těchto zařízeních pracovala spolehlivě. Ovladatelnost aplikace poněkud závisí na použitém dotykovém monitoru. Ne všechny dotykové monitory používají stejnou technologii. Nejčastěji je použita kapacitní technologie (použitý dotykový monitor v prvním případě), která je však náchylná na mastnot a některým lidem způsobuje občasné problémy (například u nižšího prokrvení prstů). Při použití v praxi bude zřejmě nejčastěji použit princip s optickým věncem okolo zobrazovací plochy (použitá TV ve druhém případě). Tato technologie se vyznačuje poněkud delší odezvou, však zcela nezávislou na různých mastnotách a jiných rušivých elementech. I při použití této technologie bylo ovládání aplikace velmi přijatelné.



Obrázek 7.1 MeeGo Pad T01



Obrázek 7.2 Dotyková obrazovka

## 7.2 NFC tagy

Jak již bylo napsáno dříve, nebylo zcela snadné najít univerzální programový kód pro čtení obsahu / identifikace NGF tagů pro desktopové i mobilní Windows. Pro ověření tedy dobrého čtení identifikace z NFC tagů jsem vybral několik dostupných NFC tagů. Ze všech těchto získaných NFC tagů bylo možno vytvořeným programovým kódem přečíst spolehlivě jejich identifikaci.

U tagů neobsahujících tzv. formátování NDEF bylo možno vyčíst pouze jejich základní textový řetězec (obdoba WWW adresy), což zcela postačuje. Aplikace stejně využívá pouze nespecifického unikátního řetězce, který se podle předem vytvořeného konfiguračního souboru převede na pozici v mapě. U tagů poskytujících NDEF tato schopnost stejně není využita a jejich obsah se čte jako libovolný text / řetězec. Ve všech případech se jednalo o pasivní tagy.

Pro ověření využití zapisovacích tagů byl použit vývojový kit „STMicroelectronics X-NUCLEO-NFC01A1“.



Jedná se rovněž o pasivní tag, avšak s možností zápisu libovolného řetězce. Pomocí vhodné aplikace (popsané dále) byla na tento tag zapsána informace „POS=12;25;2;A“. Což znamená jeho umístění do bloku mapy s polohou X=12, Y=25, Z=2 a „A“ znamená na jaké zdi je tag umístěn.

## 7.3 Demonstrační mapa

Za účelem ověření aplikace pro vhodné zobrazení mapy byla vytvořena mapa části přízemí a prvního patra budovy Fakulty elektrotechnické v Dejvicích. Mapa je zobrazena správně, ale nelze v ní (v současné době) korektně zobrazit šikmé chodby. Což je největším nedostatkem mnou navrženého formátu mapy.

## 7.4 Lokalizace mobilním zařízením

Jelikož se jedná o demonstrační aplikaci, tak pro jednoduchost byl při testování použit tablet se systémem MS Windows 10 (stejně WIN jako na desktopovém PC) a to typ ASUS ME400CL s podporou NFC. Bohužel velmi málo tabletu obsahuje podporu NFC. Zde vytvořená aplikace byla na tomto tabletu schopna přečíst všechny testované tagy (pevné i zapisovací). Aplikace „NFC Tag Writer“ (dostupná ve Windows Store) byla použita k zápisu vlastní pozice na zapisovací tag („STMicroelectronics X-NUCLEO-NFC01A1“). Tento tablet poskytuje celkem malý procesorový výkon, přesto na něm aplikace běžela velice uspokojivě.

Při přiložení NFC tagu se okamžitě zobrazovala aktuální pozice uživatele podle konfiguračního souboru pro převod identifikace NFC tagu na pozici v mapě. Rovněž bylo ověřeno tzv. „add-hoc“ přidání NFC tagu do systému (zapisovacích), které nebyly



v konfiguračním souboru uvedeny. Informace z těchto zapisovacích NFC byla úspěšně rozpoznána a získaná aktuální pozice zobrazena v mapě.

## 7.5 Plánování cesty

Nedílnou součástí aplikace je plánování cesty. Pro testování této schopnosti byly stanoveny dva typy úloh (několikrát opakované):

- Snadný úkol – Plánování cesty od kiosku / vstupu do cílové místnosti.
- Zabloudění – Nutnost přeplánování (naplánování nové) cesty při dosažení / načtení NFC tagu, který nebyl na dříve naplánované cestě.

Při plánování se hodnotily dva ukazatele:

- Správnost plánovacího algoritmu, tedy reálnost naplánované trasy (zejména začátek, konec, přímost).
- Časová náročnost plánování, tedy čas naplánování trasy.

Plánovací algoritmus vytvářel skutečně dobrou trasu cesty. I když v některých případech by člověk volil cestu možná přímější, než ta skutečně naplánovaná. Lepší plánování by bylo zřejmě dosaženo zavedením nějakého hodnocení (penalizace) podle typů místností, například rovná, i když delší, chodba je mnohdy přijatelnější než, kratší, ale klikatá.

Čas pro naplánování trasy byl rovněž velmi dostačující. Prodleva v trvání jedné až dvou vteřin na skutečně pomalém PC je velmi přijatelná. V algoritmu v současné době není vytvořena žádná optimalizace.

## 7.6 Netestované části

Některé části aplikace však nebyly testovány. Jedná se zejména o obsluhu GUI na kiosku (tedy uživatelskou dovednost / snadnost). GUI je navrženo pouze po technické stránce a pro jeho skutečné grafické a ovládací odladění by bylo vhodné použít některé metody využívané v návrhu uživatelských rozhraní a vhodně připravit testovací postup. Toto by značně překročilo rozsah této práce.

## Kapitola 8 Závěr

---

Cílem práce bylo seznámit se s dostupnými technologiemi využívanými pro lokalizaci a navigaci v rozsáhlých budovách. Dále navrhnout a částečně otestovat vytvořené řešení vlastního lokalizačního a navigačního systému. Toto vše s ohledem na snadné ovládání, jednoduchou rozšiřitelnost a nízké výkonové nároky.

V rámci práce jsem zvolit použité technologie, navrhnul celkové řešení a rovněž vytvořil demonstrační aplikaci pro operační systém Windows umožňující lokalizaci a navigaci za použití algoritmu „A star“, pro vyhledávání optimální cesty do cílového místa v budově.

Mezi hlavní výhody mého řešení považuji:

- Jednoduché a intuitivní uživatelské prostředí.
- Nenáročnost na výkon, tedy možnost použít levné mini-PC.
- Jednoduchá rozšiřitelnost mapových i lokalizačních podkladů.
- Levná instalace a údržba – NFC tagy jsou velice levné, systém nemusí spravovat specializovaná firma.

Mnou vytvořený systém je v současné době vhodný zejména pro pravoúhlé budovy, není tedy snadno aplikovatelný na budovy s kulovými místnostmi nebo s úhlopříčnými chodbami. Cílem práce bylo navrhnout a ověřit princip tohoto řešení nikoli vytvořit konečný a odladěný systém. K vytvořenému systému není dodáván žádný software pro intuitivní úpravu mapových podkladů v uživatelském prostředí.

### 8.1 Budoucí rozvoj

I když je vytvořený systém dostatečně funkční a tedy v podstatě použitelný, je několik předpokladů jak jej vhodně rozšířit a tedy dále vylepšit.

Jeden z důvodů proč jsme se rozhodl použít platformu .NET byla existence projektu Xamarin (kapitola 4.1.2.4) umožňující s minimem změn přenést již existující programový kód, a použít jej k vytvoření aplikací na tři největší mobilní platformy Android, iOS a Windows Mobile.

Dále by bylo dobré vytvořit program pro konfiguraci celého systému, ve kterém by bylo možné upravovat mapu budovy rovněž pomocí 3D rozhraní (nikoli editací souborů).

## Literatura

---

- [1] N. Farina, „Using Wi-Fi for Navigating the Great Indoors,“ radar.oreilly.com, 11. 10. 2011. [Online]. Available: <http://radar.oreilly.com/2011/10/indoor-navigation.html>.
- [2] J. Ido, Y. Shimizu, Y. Matsumoto a T. Ogasawara, „Indoor Navigation for a Humanoid Robot Using a View Sequence,“ *The International Journal of Robotics Research*, sv. 28, č. 2, p. 315–325, 2009.
- [3] Sky-Trax Inc., 2011. [Online]. Available: <http://www.sky-trax.com/>.
- [4] „symbion.cz,“ Symbion CZ s.r.o., [Online]. Available: [http://www.symbion.cz/indoor\\_navigace.html](http://www.symbion.cz/indoor_navigace.html).
- [5] Emam s.r.o., [Online]. Available: <http://emam.cz/navigacni-systemy>.
- [6] Wikipedia, „Near Field Communication,“ Wikimedia Foundation, 4. 5. 2016. [Online]. Available: [https://cs.wikipedia.org/wiki/Near\\_Field\\_Communication](https://cs.wikipedia.org/wiki/Near_Field_Communication).
- [7] J. Kosek, XML pro každého : podrobný průvodce, Praha: Grada, 2000.
- [8] R. Mautz, Indoor Positioning Technologies, Zurich: ETH, 2012.
- [9] M. Reynolds, Xamarin mobile application development for Android, Birmingham: Packt Publishing, 2014.
- [10] M. Pěchouček a M. Rollo, *Informované metody prohledávání stavového prostoru*, Praha: Katedra Kybernetiky, 2015.

## **Přílohy A – Použité zkratky**

---

GPS	Global Positioning System – systém pro lokalizaci po celém světě
Wi-Fi	Bezdrátová forma lokální sítě
CCD	charge-coupled device, senzor pro snímání obrazové informace
QR kód	„Quick Response“ kód, slouží k uložení informace
IEEE	Institute of Electrical and Electronics Engineers
RFID	Radio Frequency IDentification – rádiový
NFC	Near Field Communication
WPF	Windows Presentation Foundation
XML	Extensible Markup Language
XAML	Extensible Application Markup Language
GUI	Graphical user interface – grafické uživatelské prostředí

## **Příloha B – Obsah CD**

---

CD obsahuje následující adresáře, a v nich uvedené položky:

- Text
  - Navigace\_v\_budovach.pdf (text práce)
- Program
  - Vytvořená demonstrační navigace pro navigaci v budovách (aplikaci lze spustit pomocí souboru „Navigace\_v\_budovach“)

Pro další informace se obraťte na vedoucího práce (Ing. Petr Novák Ph.D.).