ČESKÉ VYSOKÉ
UČENÍ TECHNICKÉ
V PRAZE

F6

Fakulta dopravní
Ústav letecké dopravy

Diplomová práce

Tvoření optimálních letových tratí a prostorů s využitím interaktivního software

Bc. Tomáš Malich

Květen 2018

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
Fakulta dopravní
d ě k a n
Konviktská 20, 110 00 Praha 1

K621..... **Ústav letecké dopravy**

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení studenta (včetně titulů):

Bc. Tomáš Malich

Kód studijního programu a studijní obor studenta:

N 3710 – PL – Provoz a řízení letecké dopravy

Název tématu (česky): **Tvoření optimálních letových tratí a prostorů s využitím interaktivního software**

Název tématu (anglicky): Creating Optimal Air Routes and Air Spaces Using Interactive Software

Zásady pro vypracování

Při zpracování diplomové práce se řiďte osnovou uvedenou v následujících bodech:

- Rešerše týkající se softwarových nástrojů používaných v oblasti řízení letového provozu (vzhled, interakce, nejlepší praktiky)
- Rešerše v oblasti tvorby letových tratí a prostorů
- Analýza a návrh nového software pro tvorbu letových tratí a prostorů
- Implementace software
- Testování software
- Návrhy pro zlepšení a rozšíření software

Rozsah grafických prací: dle pokynů vedoucího diplomové práce

Rozsah průvodní zprávy: minimálně 55 stran textu (včetně obrázků, grafů a tabulek, které jsou součástí průvodní zprávy)

Seznam odborné literatury: European Air Traffic Management: Principles, Practice and Research (Andrew Cook)

The Future of Air Traffic Control:: Human Operators and Automation (National Academies Press)

Architektura softwaru (Peter Cripps, Peter Eeles)

Vedoucí diplomové práce: **Ing. Stanislav Pleninger, Ph.D.**

Ing. Jiří Frei, Ph.D.

Datum zadání diplomové práce: **28. července 2017**
(datum prvního zadání této práce, které musí být nejpozději 10 měsíců před datem prvního předpokládaného odevzdání této práce vyplývajícího ze standardní doby studia)

Datum odevzdání diplomové práce: **29. května 2018**
a) datum prvního předpokládaného odevzdání práce vyplývající ze standardní doby studia a z doporučeného časového plánu studia
b) v případě odkladu odevzdání práce následující datum odevzdání práce vyplývající z doporučeného časového plánu studia

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE Fakulta dopravní L. S.

Ing. Jakub Kraus, Ph.D.
vedoucí
Ústavu letecké dopravy

prof. Dr. Ing. Miroslav Svítek, dr. h. c.
děkan fakulty

Potvrzuji převzetí zadání diplomové práce.

Bc. Tomáš Malich
jméno a podpis studenta

V Praze dne 28. července 2017

Poděkování / Prohlášení

Děkuji všem lidem, kteří mi poskytli pomoc při tvorbě této diplomové práce. Ať to byly znalosti či morální podpora, oboje je velmi důležité. Zvláštní poděkování patří mým dvěma vedoucím práce, panu Ing. Stanislavu Pleningerovi, Ph.D. a panu Ing. Jiřímu Freiovi, Ph.D. Obrovské poděkování také patří mé rodině, která mě vždy podporovala a bez jejíž pomoci by tato práce nikdy nevznikla.

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje vsouladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací.

V Praze dne 29.5.2018

.....

Abstrakt / Abstract

Tato práce se zabývá návrhem a částečnou implementací software pro tvorbu a úpravu letových tratí, sektorů a vzdušných prostorů. Součástí práce je analýza možných technologií pro vývoj. Dále jsou analyzovány postupy a zařízení pro leteckou navigaci, podle kterých je tvořen návrh systému. Práce se také zaměřuje na využití virtuální a augmentované reality.

This diploma thesis deals with design and implementation of software for creation and editation of flight routes, sectors and airspaces. There is also analysis of possible technologies, which can be used for development. Next there are analysis of practices and devices which are used in aviation. Those analysis are than used for software design. Thesis also deals with possible usage of virtual and augmented reality.

Obsah /

/ Obrázky

Kapitola 1
Úvod

Tato práce vznikla ve spolupráci se společností Řízení letového provozu České republiky, s.p (dále jen ŘLP). Jelikož jsem já, jako autor práce, zaměstnáním programátor, chtěl jsem využít mých znalostí z oblasti softwarového návrhu a implementace k vytvoření aplikace, která bude zaměřena na řízení letového provozu a letectví obecně.

Představu o tom, k čemu by měla aplikace sloužit, poskytlo ŘLP. Vzhledem k neustálému narůstání hustoty letecké dopravy je nutné stále inovovat technologie a procesy pro řízení letového provozu.

Nyní se velká civilní letadla dostávají do cílových destinací pomocí letových tratí. Ty jsou tvořeny traťovými body. Během řízeného letu se letadlo může vyskytnout v několika různých sektorech, přičemž každý z těchto sektorů je ovládán jiným řídícím letového provozu.

Umístění a rozdělení těchto sektorů společně s umístěním traťových bodů tvoří velmi důležitou roli v efektivitě civilní letecké dopravy.

Existuje tedy poptávka po software, ve kterém bude možné stávající tratě, body a sektory upravovat a tvořit tak, aby efektivita letecké dopravy byla co nejvyšší.

Jeden z hlavních požadavků pro takový software je jednoduchost užívání. Základními kroky by mělo být zvolení destinace na Zemi, kde se budou provádět úpravy a dále už vytvářet změny pomocí intuitivního grafického rozhraní.

Aplikace by také měla umožňovat importování leteckých map, aby nebylo nutné pro každou destinaci zadávat tratě, traťové body a vzdušné prostory ručně.

Další funkcí by měla být editace vzdušných prostorů a sektorů řízení letového provozu.

Existuje také několik vizí budoucnosti, které by mohly poměrně značně změnit pravidla řízení letového provozu. Na to by měl být software připraven a programátor by měl při vývoji neustále pamatovat na to, aby změny v programu šly provést jednoduše, bez větších zásahů.

Cílem této práce není vytvoření takového software, ale jeho návrh a částečná implementace, která bude sloužit převážně k posouzení správnosti návrhu.

1.1 Struktura práce

Následující text je členěn do tří kapitol.

Rešerše a analýza se zabývá fungováním letecké dopravy a jejího řízení. Dále se dotýká nových vizí pro řízení letového provozu. V neposlední řadě se věnuje možnostem využití technologií pro virtuální či augmentovanou realitu a zdali je tato technologie vhodná pro navrhovaný software.

Návrh se zaměřuje na technologické aspekty software a slouží jako podklad k následné implementaci.

Implementace se zabývá hotovými algoritmy pro výpočty v systému.

Kapitola 2
Rešerše a analýza

2.1 Stručná historie navigace v letectví

Na začátku civilního letectví, které se datuje od dob po první světové válce, se pro navigaci využívalo pouze vizuálních prostředků. Jednalo se především o orientační nápisy, viditelné ze vzduchu nebo světelné majáky pro navigaci v noci. Navigační výbavou letadel byla mapa, kompas, hodiny a zrak pilota. Možnost provádět lety byla značně ovlivněna počasím a za špatných podmínek byla velmi ohrožena bezpečnost prováděných letů.

Prvním krokem k technologizaci letecké navigace bylo umístění radiotelegrafních zařízení na paluby letadel. Tím byla zajištěna možnost informovat posádky o změnách počasí. Zároveň byly položeny základy řízení letového provozu.

Velký zlom nastal po vynalezení radaru. Původně byl vyvinut pro vojenské účely a po skončení druhé světové války se začal užívat i v civilním letectví.

V průběhu dalších let docházelo k velkému nárůstu hustoty provozu ve vzdušném prostoru a také rychlostí letadel. To vedlo k čím dál větší potřebě využívat přesnější navigace a zvyšování dosahu navigačních zařízení. V průběhu 50. a 60. let byly vyvinuty systémy pro navigaci na dlouhé vzdálenosti, které využívaly hyperbolického určování polohy. Tyto systémy byly následně upozaděny při rozvoji satelitních navigačních systémů.

Pro navigaci na krátké vzdálenosti bylo a stále je využíváno principu radiového zaměřování.

Za zmínku stojí nesměrový radiomaják (NDB), který využívá všesměrovou anténu a vysílá na středních vlnových délkách. Přijímač, využívající NDB se nazývá automatický radiokompas (ARK, anglicky ADF - automatic direction finder). ARK umožňuje získat dva navigační údaje - kurzový úhel radiostanice a azimut radiostanice.

Další zařízení pro radiové navádění je všesměrový radiomaják (VOR) a měřiče vzdálenosti (DME). VOR umožňuje určit azimut letadla od vysílače. DME umožňuje určit šikmou vzdálenost mezi letadlem a vysílačem. Velmi často se VOR a DME slučuje v jedno zařízení VOR/DME, které pak poskytuje oba navigační údaje.

V 60. letech také došlo k použití INS. INS je v současné době jediný systém využívaný v civilním letectví, který není závislý na pozemních ani satelitních zařízeních. Princip systému spočívá v určování polohy letadla ve vztažné soustavě pomocí určování směru a rychlosti letu. K tomu se využívá gyroskopu a akcelerometru. Je tedy potřeba nejříve určit polohu letadla v soustavě a poté jí měnit v závislosti naměřených dat z přístrojů

INS.

Pro kritické fáze letu, jako je příblížení na přistání, se začala využívat přesná přístrojová příblížení. To jest přiblížení využívající systémy ILS, MLS, PAR a GLS. To jsou systémy, které poskytují vertikální vedení.

ILS je tvořeno kurzovým a sestupovým majákem. Kurzový maják obsahuje anténní systémy, které vyzařují signál ve směru osy dráhy v pásmu velmi krátkych vln. Jeden vysílač těchto vln je umístěn mírně nalevo a druhý mírně napravo (bráno z pohledu přistávajícího letadla). U jednoho z vysílačů je nosná vlna modulována kmitočtem 90 Hz a u druhého 150 Hz. Pokud se tedy letoun přibližuje přesně v ose dráhy, pak je úroveň signálu z obou vysílačů stejná. Naopak při vychýlení z osy se úroveň jednoho ze signálů zvyší, a tím je indikátor v přístroji na palubě vychýlen na příslušnou stranu. Sestupový maják pracuje na stejném principu, jen jsou osy vyzařování vychýleny nad a pod sestupovou rovinu.

Společně se zdokonalováním navigačních systému se rozvíjela i oblast sledování a komunikace.

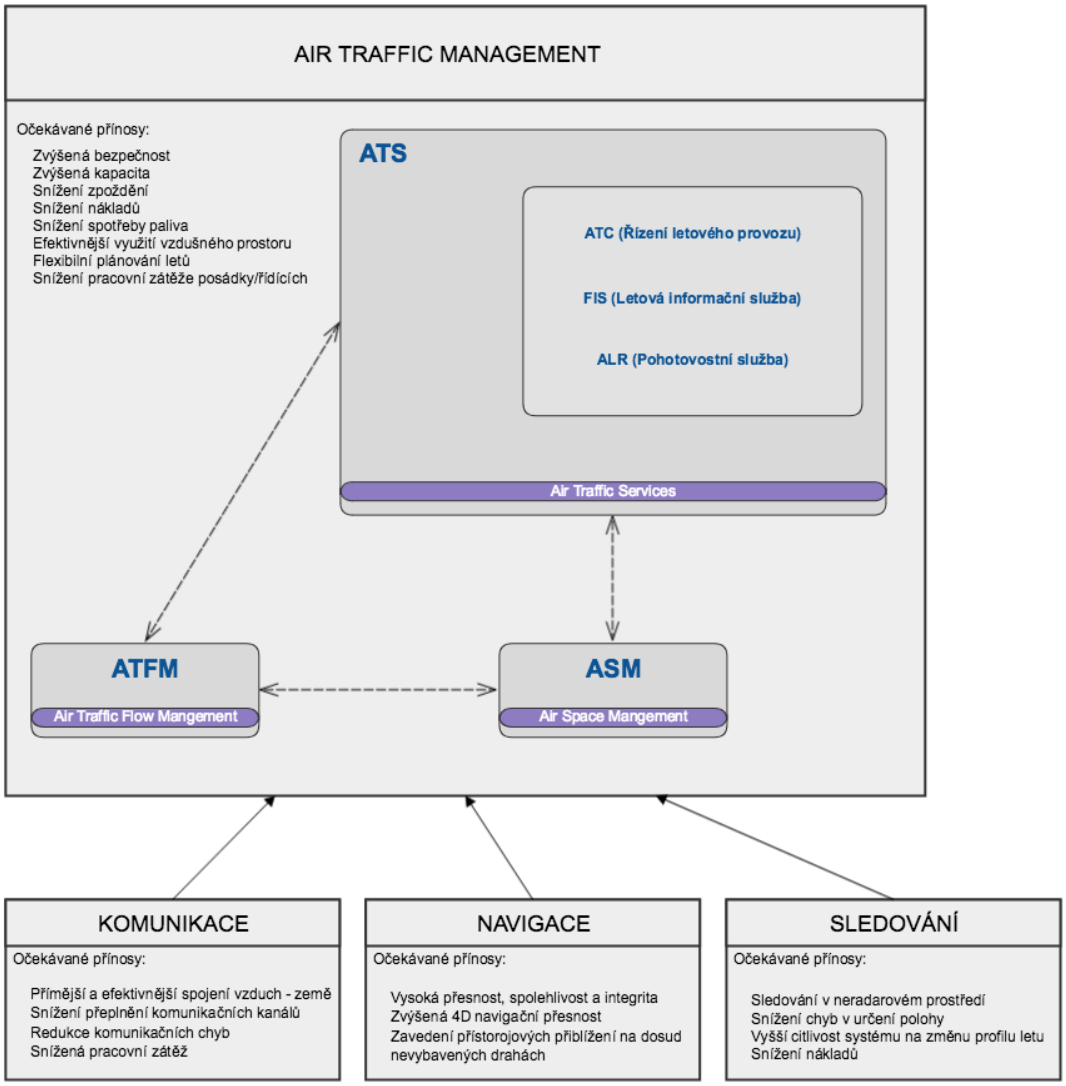
Prostor s radarovým pokrytím se stále zvětšoval, obvzlášť pak v okolí letišť. To mělo za následek rozdělení vzdušného prostoru jak vertikálně, tak horizontáně. Prostor byl také kategorizován podle zamýšleného provozu. Spolu s kategorizací prostorů došlo také ke specializaci a rozdělení služeb řízení letového provozu.

Zásadním milníkem v letecké navigaci bylo využití satelitní navigace. Prvním satelitním navigačním systémem byl TRANSIT, systém ze Spojených států amerických, který začal fungovat v roce 1964 a pro civilní užití byl uvolněn v roce 1967. Využíval dopplerovské metody určování polohy. V letectví se využíval převážně pro korekce jiných navigačních systémů, například INS. Od využívání TRANSITu se upustilo díky nástupu systému GPS.

Díky neustálé expanzi letecké dopravy a částečné neschopnosti zavádět nové technologie do praxe se neustále zvětšoval rozdíl mezi poptávkou a ze stran leteckých společností nad nabídkou kapacity vzdušného prostoru. To vedlo v roce 1983 Radu ICAO k ustanovení výboru FANS, jehož cílem bylo nalezení řešení pro nevyhovující stav letecké dopravy a také představení základních kamenů, na kterých bude v budoucnu stát zvyšování výkonnosti letecké dopravy.

Výbor FANS představil ATM, jehož definicí jsou procesy, postupy a prostředky, které v konečném důsledku zajišťují, aby letadlo bylo bezpečně vedeno nejen ve vzduchu, ale i na zemi. Skládá se z několika komponentů. ATS, ATFM a ASM. Výbor také představil koncept požadované navigační výkonnosti RNP.

Vztah CNS a ATM, tak jak byl navržen výborem je znázorněn na obrázku 2.1. [1][2]



Obrázek 2.1. Vztah mezi CNS a ATM.

2.2 Rozdělení vzdušného prostoru

2.2.1 Třídy vzdušného prostoru

Vzdušný prostor v České republice je rozdělen do několika tříd a to dle úrovně poskytovaných letových provozních služeb (ATS). Ty se označují písmeny. Konkrétně se jedná o třídy C, D, E a G.

Prostory klasifikované jako C, D nebo E jsou řízené vzdušné prostory.

Řízený vzdušný prostor definuje společnost Řízení letového provozu České republiky tako: „Řízený vzdušný prostor je vymezený vzdušný prostor, ve kterém se poskytuje služeb řízení letového provozu v rozsahu odpovídajícím jeho klasifikaci. Ve vzdušném prostoru třídy E však lety VFR nepotřebují letové povolení a nemají zde ani povinnost udržovat stálé obousměrné spojení se stanovištěm ATS.”

Prostor s klasifikací G je neřízený vzdušný prostor, v němž se poskytuje všem letům pouze letová informační a pohotovostní služba.

Obrázek 2.2 znázorňuje pravidla a rozsah poskytovaných služeb pro jednotlivé kategorie vzdušných prostorů. [3]

<table>
<tr><th>Třída</th><th>Od FL</th><th>Do FL</th><th>Způsob vedení letu</th><th>Rozestupy zajištené ŘLP</th><th>Poskytované ATS</th><th>Omezení rychlosti</th><th>VFR podléhá letovému povolení</th></tr>
<tr><td>C</td><td>195</td><td>660</td><td>IFR, VFR jen ve speciálních případech (ve vyhrazených prostorech).</td><td rowspan="4">Mezi IFR a IFR, IFR od VFR, VFR jsou podávány informace o provozu VFR (na žádost).</td><td rowspan="4">Služba ATC pro zajištění rozestupu od IFR letů a podání informace o provozu VFR (na žádost).</td><td rowspan="2">M1 IAS</td><td rowspan="4">Ano</td></tr>
<tr><td>C</td><td>100</td><td>195</td><td rowspan="2">IFR, VFR</td></tr>
<tr><td>C</td><td>95</td><td>100</td><td rowspan="2">250 KT IAS (Pouze lety VFR)</td></tr>
<tr><td>C</td><td colspan="2">CTR a TMA Praha</td><td>IFR, VFR</td></tr>
<tr><td>D</td><td colspan="2">CTR a TMA řízených letišť vyjma TMA Praha</td><td>IFR, VFR</td><td>Mezi IFR a IFR, IFR od VFR.</td><td>Informace o provozu mezi VFR a IFR lety (na zádost provozní informace vyhnout se provozu).</td><td>250 KT IAS pod FL 100</td><td>Ano</td></tr>
<tr><td>E</td><td>300 m AGL</td><td>95</td><td>IFR, VFR</td><td>Mezi IFR a IFR.</td><td>Informace o provozu, pokud je to možné.</td><td>250 KT IAS</td><td>Ne</td></tr>
<tr><td>G</td><td>0</td><td>300 m AGL</td><td>IFR, VFR</td><td>Žádné</td><td>Pokud je to vhodné jsou poskytovány informace o ostatním provozu</td><td>250 KT IAS</td><td>Ne</td></tr>
</table>

Obrázek 2.2. Pravidla pro třídy vzdušného prostoru.

Poznámka: „Na obrázku jsou uvedeny pouze údaje, které jsou relevantní pro další využití v této diplomové práci."

Prostory se dále člení dle jejich využití a dalších pravidel.

2.2.2 Zakázaný prostor

Prostor označován jako LKP + číslo.

Vymezený vzdušný prostor, v němž jsou lety letadel zakázány. Na žádost uživatelů je možné udělit povolení pro vstup do zakázaného prostoru. Toto povolení vydává Úřad pro civilní letectví ČR.

Při plnění úkolu a časové tísni je vstup také povolen pro:

- Lety policejní.
- Lety letecké záchranné služby, které bezprostředně souvisí se záchranou lidského života.
- Lety za účelem pátrání a záchrany.
- Lety provádějící leteckou hasičskou činnost.

2.2.3 Omezený prostor

Prostor označován jako LKR + číslo.

Vymezený vzdušný prostor, ve kterém jsou lety letadel omezeny v případě, že je prostor aktivován a není získáno povolení od příslušného stanoviště ATS.

2.2.4 Nebezpečný prostor

Prostor označován jako LKD + číslo.

Vymezený vzdušný prostor, ve kterém mohou v určitě době probíhat činnosti nebezpečné pro let. Je doporučeno se nezepečným prostorům vyhýbat.

2.2.5 Dočasně rezervovaný prostor

Prostor označován jako LKTRA + číslo.

Vymezený vzdušný prostor, v němž může probíhat pouze letecká činnost a přes který se v době jeho aktivace nemůže proletět, není-li k tomu zvlášť získáno letové povolení.

2.2.6 Dočasně vyhrazený prostor

Prostor označován jako LKTSA + číslo.

Vymezený vzdušný prostor, v němž může probíhat pouze letecká činnost a přes který nebude v době jeho aktivace povolen průlet.

2.2.7 Dočasně omezený prostor

Dočasná rezervace těch částí vzdušného prostoru, které nejsou publikovány v AIP ČR.

2.2.8 Oblast s povinným odpovídačem (TMZ)

Prostor, ve kterém má letadlo povinnost být vybaveno odpovídači hlásícími tlakovou nadmořskou výšku. Jedná se o odpovídače SSR schopnými provozu v módech A a C nebo v módu S. Zároveň platí povinnost tyto odpovídače používat, pokud poskytovatel letových navigačních služeb nestanoví pro daný vzdušný prostor jinak.

2.2.9 Letištní provozní zóna (ATZ)

Prostor, který je zřízen nad letištěm, kde není poskytována služba řízení letového provozu.

2.2.10 Řízený okrsek (CTR)

Prostor v těsném okolí řízeného letiště.

2.2.11 Koncová řízená oblast (TMA)

Prostor, který se zřizuje nad velkými, řízenými letišti. Popřípadě tam, kde se sbíhají letové cesty okolo jednoho nebo více hlavních letišť. Slouží k ochraně přibližujících se a odlétajících letadel v okolí letiště. [4] [5]

2.3 Rozdělení služeb ATC v ČR

Pro řízené vzdušné prostory (čili prostory klasifikované třídami C, D nebo E) je nutné zajistit služby řízení letového provozu.

Poskytování těchto služeb je rozděleno do několika kategorií viz. obrázek 2.3. [6]

Mezinárodní název	Zkratka	Český název	od [ft AGL]	do [ft AGL]	Místo působnosti
Delivery	DEL	Delivery	0	0	Slouží pro poskytování povolení před letem.
Ground	GND	Ground	0	0	Pojíždění a stojánky.
Tower	TWR	Věž	0	5000	Dráhy a prostor CTR
Approach	APP	Přiblížení	~ 3000	FL125	Prostor TMA
Departure	DEP	Odlety	~ 3000	FL125	Prostor TMA
Director	DIR	Direktor	~ FL70	3000	Prostor TMA
Control	ACC	Oblast	~ FL70	FL660	Prostory mimo CTR a TMA
Info	INFO	Info	1000	FL95	Prostory mimo CTR a TMA pro lety VFR

Obrázek 2.3. Rozdělení služeb ATC.

2.3.1 Delivery

Služba Delivery uděluje povolení letadlům. V ČR tuto službu poskytuje pouze Letiště Václava Havla Praha, Ruzyně a to pouze v dobách provozních špiček. Mimo špičku je zastoupena službou Tower.

2.3.2 Ground

Služba Ground má na starost řízení pohybů na plochách letiště, až na vzletové a přistávací dráhy. Také služba Ground bývá mimo špičku zastoupena službou Tower.

2.3.3 Tower

Služba Tower se stará o pohyby na dráze, přistání, vzlety a provoz v těsném okolí letiště (CTR).

2.3.4 Approach

Služba Approach řídí provoz v prostoru TMA. Zde dochází k navádění letadla na přistání nebo k opuštění prostoru TMA po vzletu.

V TMA se nachází síť příletových a odletových tratí. Ty jsou vymyšleny tak, aby nedocházelo ke konfliktům s letadly na ostatních tratích.

Při hustém provozu v oblasti TMA je nutné aktivovat služby Director a Departure. Služba Director se poté stará o letadla, která mají úmysl přistát. Služba Departure se naopak stará o letadla odlétající z letiště.

2.3.5 Control

Služba Control (Oblast) řídí ostatní řízený vzdušný prostor mimo letišť a jejich TMA a CTR. Control se stará především o zajištění plynulosti provozu.

Letadla zde létají po tratích a to velmi vysokými rychlostmi (500 až 1000 km/h GS).

Zátěž řídícího letového provozu může být extrémní. Na řešení konfliktu zbývá minimálně času a je třeba konflikty řešit desítky až stovky kilometrů předem.

Tento fakt a také velká hustota provozu na tratích vynutila dělit letové prostory na sektory. Sektory jsou vymezeny jak vertikálně, tak plošně. Každý sektor je řízen dedikovaným řídícím. Sektory lze aktivovat a deaktivovat dynamicky. V praxi to znamená, že více menších sektorů může být při menším provozu sloučeno v jeden velký sektor, který je následně řízen jedním řídícím letového provozu. Pokud by měl počet letadel v sektoru narůst nad povolenou hranici, opět se sektor rozdělí do menších celků.

Pokud letadlo opouští hranici sektoru a vstupuje do prostoru jiného, pak probíhá předání zodpovědnosti za řízení mezi řídícími letového provozu. Toto předání znamená pro pilota přeladění rádia na jinou frekvenci a předání letu v systému řízení.

Aktuální rozdělení sektorů (dle AIP) v ACC je popsáno na obrázku 2.4.
Sektory low a middle se používají pro přibližování, odlety a lety na blízké vzdálenosti. Popřípadě se využívají, pokud letadlo nemůže stoupat výše (například to nedovoluje jeho technický stav).

Běžné lety na delší vzdálenosti se provádějí v sektorech high a top. Letadla využívající sektory top jsou velmi často letadla soukromá, která cestují nad tratěmi, a tím pádem neomezují provoz na tranzitních tratích.

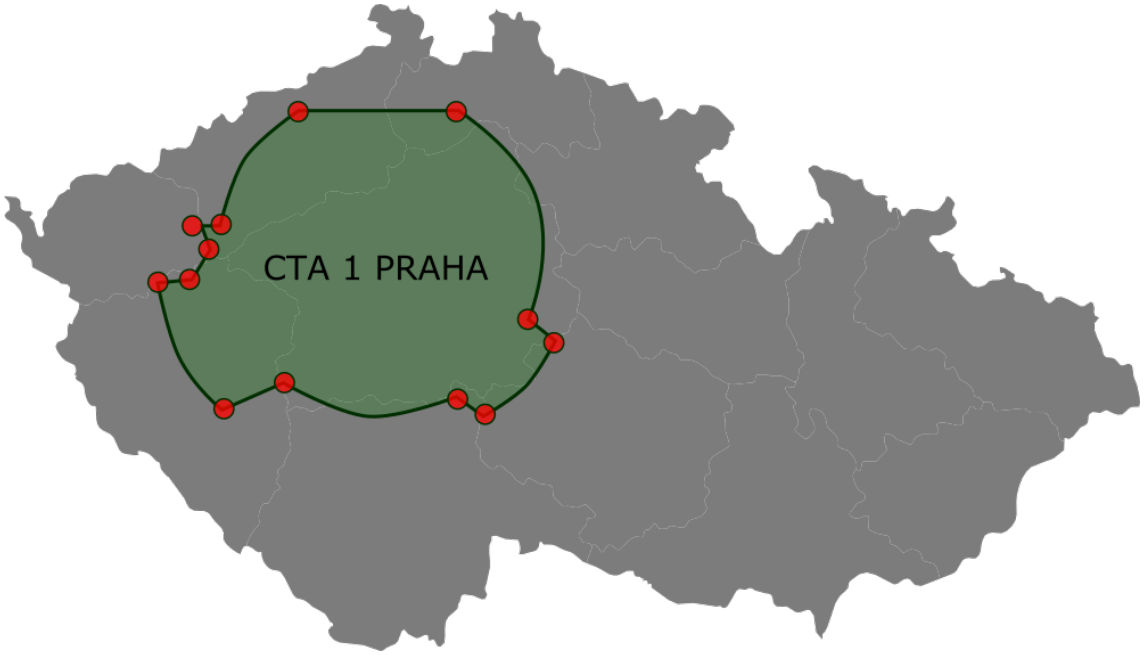
Letové hladiny se dělí na sudé a liché. Letadla, která letí kurzem 0 až 180 stupňů, letí vždy v lichých letových hladinách a letadla letící kurzem 181 až 359 stupňů využívají sudé hladiny. Tím je zajištěn výškový rozestup mezi proti sobě letícími letadly na stejné trati o 1000 stop.

Rozsahy sektorů jsou popsány v AIP. K jejich popisu se využívá geo souřadnic (zeměpisná šířka / délka), části státních hranic a kruhových oblouků s daným poloměrem.

Název sektoru	Část sektoru	Od FL	Do FL
WL - west low		125	305
WM - west middle		305	355
WH - west high		355	375
WT - west top		375	660
NL - north low	NL - PART 1	125	305
NL - north low	NL - PART 2 - AREA W OF OKX	125	305
NL - north low	NL - PART 3 - AREA S OF KLODZKO	245	305
NL - north low	NL - PART 4 - AREA S OF DESEN	125	245
NM - north middle	NM - PART 1	305	355
NM - north middle	NM - PART 2 - AREA W OF OKX	305	355
NM - north middle	NM - PART 3 - AREA S OF KLODZKO	305	355
NH - north high	NH - PART 1	355	375
NH - north high	NH - PART 2 - AREA W OF OKX	355	375
NH - north high	NH - PART 3 - AREA S OF KLODZKO	355	375
NT - north top	NT - PART 1	375	660
NT - north top	NT - PART 2 - AREA W OF OKX	375	460
NT - north top	NT - PART 3 - AREA S OF KLODZKO	375	460
SL - south low		125	305
SM - south middle		305	355
SH - south high		355	375
ST - south top		375	660
CTA 2 PRAHA		125	660

Obrázek 2.4. Sektory v ACC.

Na obrázku 2.5 je znázorněn sektor CTA 1 Praha. Červeně vyznačené body jsou pozice geo souřadnic. [7][8][9]



Obrázek 2.5. CTA 1 PRAHA.

2.4 Prostorová navigace - RNAV

RNAV je způsob letecké navigace při letu podle přístrojů (IFR), který vede letadlo po jakékoli trati pokryté signálem referenčních navigačních prostředků, nebo s využitím autonomních navigačních zařízení, případně kombinací obojího. Prostorová navigace umožňuje vést letadlo z jakéhokoli bodu či fixu na jiný. Tyto fixy jsou definovány libovolnými zeměpisnými souřadnicemi, čili zeměpisnou šířkou a délkou. Umožňuje tedy provedení letu bez nutnosti přeletu pozemních radionavigačních zařízení.

K předpokladům pro provedení takového letu patří nutnost vybavení letadla avionikou umožňující určit polohu letadla ve vytvořeném modulu, jeho odchylku od plánované tratě, vzdálenost do traťového bodu atd.

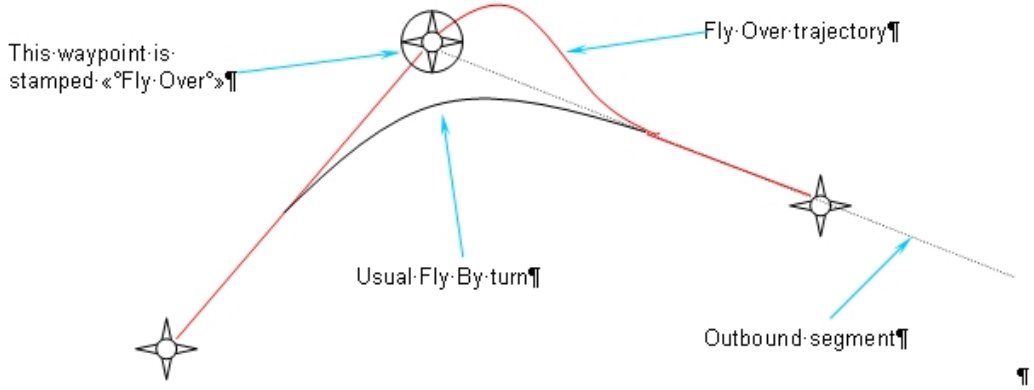
Komplexní RNAV systémy umožňují integraci vstupů z různých navigačních senzorů (jako jsou GNSS, VOR/DME, ANS, atd.) a jejich další zpracování.

Systémy také musí obsahovat navigační databázi s tratěmi LNS, traťovými body a postupy pro standartní přístrojové přílety a odlety.

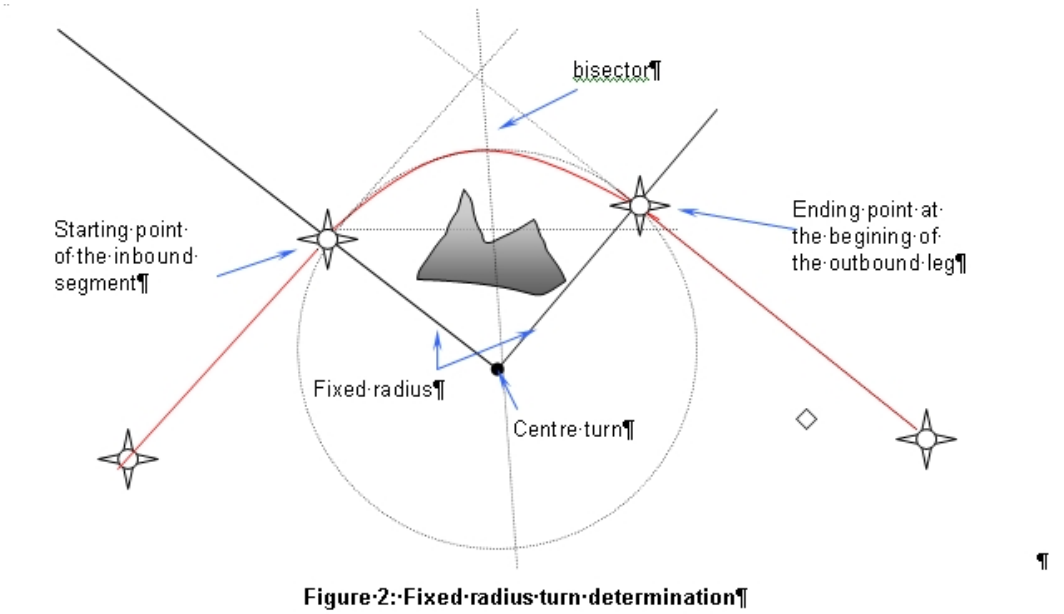
Navigace RNAV využívá waypointů (traťových bodů). Waypoint je bod tratě, ve kterém naváděný stroj mění svůj kurz, rychlost nebo výšku podle plánované tratě. Jsou definovány zeměpisnou šířkou a délkou. Výška je ignorována. Traťové body mají názvy nebo jsou spojeny s existujícími navigačními zařízeními či fixy. Využívají se dva typy waypointů - Fly-over a fly-by. Fly-over musí být naváděným strojem překročen na vertikální rovině. Fly-by určuje zatáčku s konstantním poloměrem, kterou je nutné provést mezi dvěma waypointy.

Fly-over a fly-by waypointy jsou znázorněny na obrázku 2.6

Zatáčka s konstantním poloměrem je většinou uplatněna kvůli vyhnutí se překážce. Je definována pomocí počátečního bodu zatáčky, konečného bodu zatáčky a středu otáčení. Tyto parametry jednoznačně určují průběh zatačky. Znázornění je na obrázku 2.7



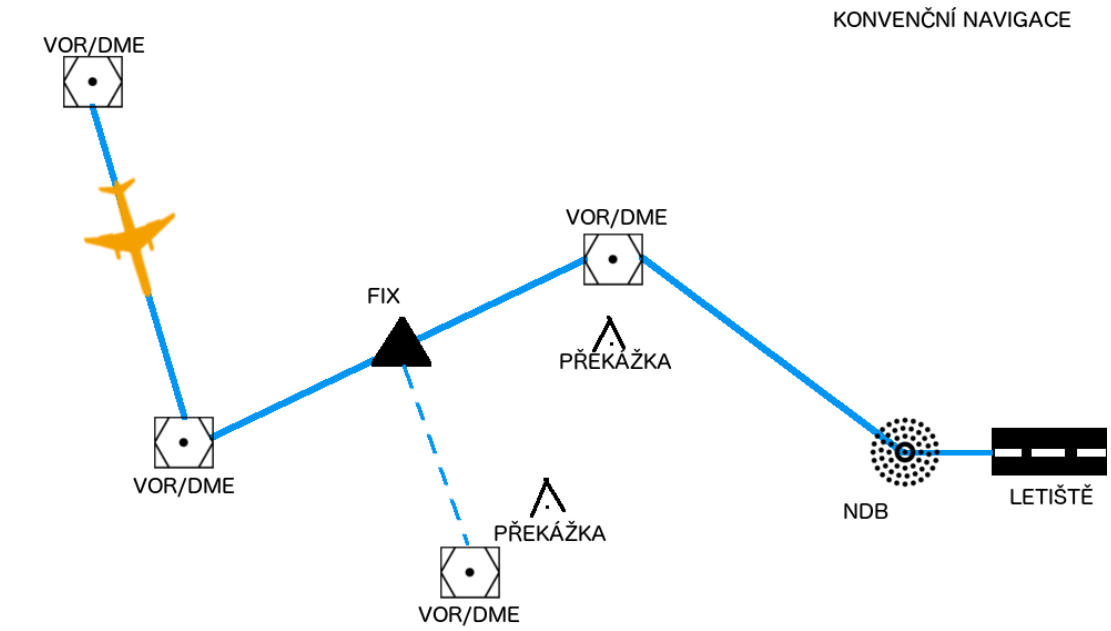
Obrázek 2.6. Fly-over a fly-by waypointy.



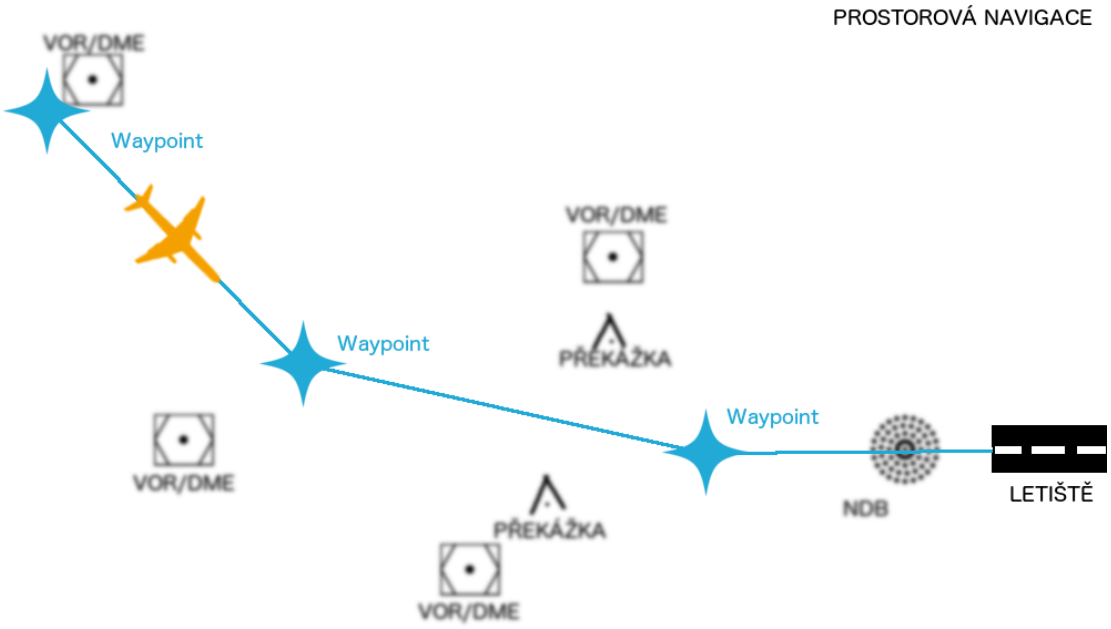
Obrázek 2.7. Zatáčka s konstantním poloměrem

Výhody RNAV navigace oproti konvenční (přístrojové) navigaci jsou kratší délky tratí a optimální vedení tratě. Díky funkcím systému je možné určit vzdálenost do traťového bodu, předpokládaný čas jeho přeletu nebo například příčnou odchylku od letěné tratě. Další velkou výhodou je flexibilita při reorganizaci a dílčích změnách ASM.

Na obrázcích 2.8 a 2.9 jsou znázorněny tratě při konvenční a prostorové navigaci. [10][11]



Obrázek 2.8. Konvenční navigace.



Obrázek 2.9. Prostorová navigace.

2.5 Performance-based navigation

Performance-based navigation rozšiřuje koncept RNP, Required navigation performance. RNP je provozní specifikace, která pracuje s pojmy přesnost, integrita, spojitost a dostupnost, což jsou požadavky na navigační výkonnost.

RNP specifikace se implementuje v místech, kde není dostatečné radarové pokrytí (například let nad oceány) a tam, kde je poloha letadla kritická vzhledem k překážkám (například přiblížení RNP).

Koncept PBN integruje původní požadavky na navigační výkonnost do takzvané navigační specifikace. Ta obsahuje navíc požadavky na výcvik pozemního personálu i posádek letadel, funkce palubního zařízení a jiné. Obsahuje navíc požadavek na "On board monitoring and alerting", což znamená, že palubní navigační systém kontroluje svou činnost a když usoudí, že neukazuje správně, tak vydá pilotovi výstrahu.

PBN se odklání od senzorové navigace k navigaci na požadavcích na výkonnost tak, jak jsou definovány pro konkrétní navigační specifikaci. Ta definuje také použitelné navigační senzory. Díky použití metod, jakými jsou například zatáčky s konstantním poloměrem, lze vést letadla s vysokou přesností a v zaručeném koridoru i po zakřivených tratích.

RNP (i jako součást PBN) je založena na výkonových požadavcích na letadlo, které se pohybuje po trati ATS. Požadavky na výkonnost jsou vyjadřeny již zmíněnými pojmy přesnost, integrita, spojitost a dostupnost.[12]

2.5.1 Přesnost

Přesnost v navigaci se skládá ze tří komponent.

- PDE - Chyba definice letové cesty (Path definition error).
- FTE - Letově technická chyba (Flight technical error).
- NSE - Chyba navigačního systému (Navigation system error).

Použití systému předpokládá, že trasa je definována v navigační databázi. PDE nastane, když definovaná trať v systému vykazuje odchylku od skutečné tratě. FTE souvisí se schopností pilota (autopilota) dodržet definovanou trať. NSE závisí na přesnosti aktuální pozice letadla podle navigačního systému.

Celková chyba se pak definuje jako TSE (Total system error), která je funkcí PDE, FTE a NSE. Požadavek na přesnost je vyjádřen jako maximální přípustná hodnota TSE v námořních mílích během 95% z celkové doby letu daného postupu.

2.5.2 Integrita

Schopnost poskytovat varování v případě, že se systém dostane do stavu, kdy není bezpečný k užívání. Chyba, která by mohla vést k nebezpečně zavádějícím informacím musí být detekována v daném intervalu na dané hladině pravděpodobnosti.

Klíčové kvantifikátory integrity jsou Alert limit (AL), Time to alert (TTA) a IR (Integrity risk). AL je hodnota parametru, při jejímž překročení vydá systém varování.

TTA je předepsaný čas, ve kterém musí bát vydáno varování. IR je pravděpodobnost, že varování nebude vydáno v kratším čase než je definován v TTA.

2.5.3 Spojitost

Schopnost systému provádět danou činnost bez neplánovaného přerušení v průběhu zamýšleného provozu. Vyjadřuje pravděpodobnost, že systém neselže v poskytování informace během zamýšlené doby provozu. Nároky na spojitost se zvyšují se zvyšující se hustotou provozu a složitostí vzdušného prostoru.

2.5.4 Dostupnost

Procento času (časový úsek) během kterého je služba využitelná vzhledem k času, na který byl provoz systému plánován. V úvahu se berou všechny výpadky bez ohledu na jejich příčinu. Služba je dostupná, pokud splňuje požadavky na přesnost, integritu i kontinuitu.

2.6 Free Flight Koncept

Současné řízení letového provozu je postaveno na systému, který je centralizovaný a využívá pevně navržených dopravních cest. Centralizovaný je proto, že rozhodnutí o trati letu je v rukách řídícího letového provozu. Řídící je aktivní součást systému a letadla jsou pasivní součást systému.

Letecká doprava pod radarovým pokrytím je uspořádána do letových cest. I přes to, že dnes už se letová cesta neskládá pouze z přímých tratí, které by byly vedeny od jednoho radiového majáku ke druhému, zůstávají letové cesty zachovány. Hlavním důvodem je, že tento způsob uspořádání letového provozu umožňuje řídícím sledovat velký prostor, který by vypadal chaoticky, pokud by letadla létala přímo z odletové do cílové destinace.

ATC zajišťuje jak vertikální, tak horizontální rozestupy mezi letadly. Vertikální rozestup je stanovený letovými hladinami. Horizontální rozestup je zaručen podélným rozestupem podle času nebo vzdáleností (na trati) a příčným rozestupem (letové cesty).

Problémy s rozesupy nastávají při křížení letových cest, změny výšky a při vzájemném přelétáváním na trati.

Letové cesty mají řadu nevýhod. Až na výjimky nekopírují optimální trať (vzhledem k délce letu, letěné vzdálenosti, spotřeby pohonných hmot atd.). Hustota provozu na tratích je uměle zvýšena, protože se nevyužívá celý vzdušný prostor.

Z pohledu dopravců současný systém ATM neumožňuje optimální let. Trajektorie letu jsou přizpůsobeny potřebám řídících tak, aby byl provoz na obrazovce zobrazen s určitým řádem. Jinak si řídící nedokáže dostatečně uvědomovat situaci ve vzdušném prostoru. Pokud to situace dovolí, je možné, aby řídící povolil posádce let podle jejich požadavků.

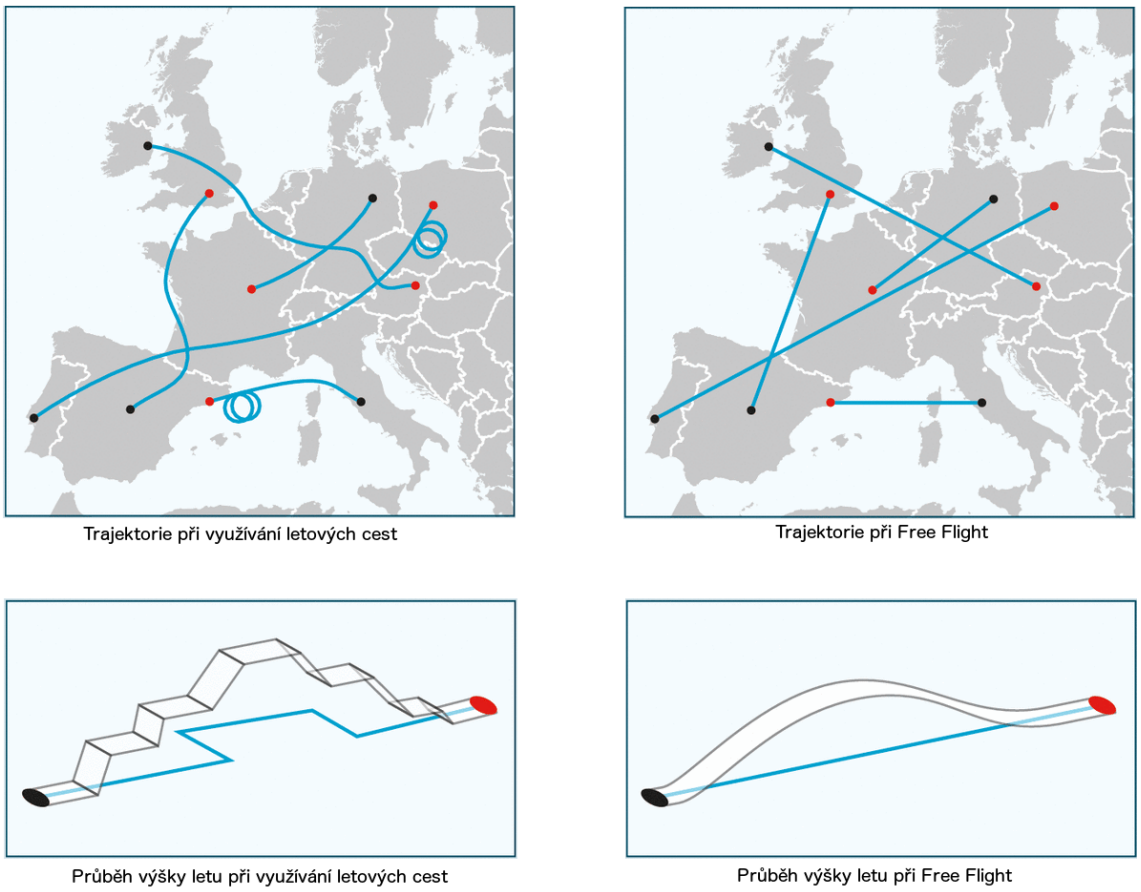
Letečtí dopravci by ovšem jistě uvítali systém, který by umožňoval vést letadla po optimálních trajektoriích, a tím pádem ušetřil na spotřebě paliva a času stráveném ve vzduchu. Podmínkou je ovšem zachování stejné míry bezpečnosti jako s využitím letových cest.

Tato myšlenka vedla ke vzniku konceptu Free Flight.

Zásadní změna, kterou přináší tento koncept, je přesunutí zodpovědnosti k zajištění rozestupů od řídících do kokpitů letadel. Letová posádka může s využitím přístrojů zajistit rozestup od okolního provozu za předpokladu, že je dostatečně informována o situaci v okolí letadla.

Každé letadlo, které vstoupí do prostoru s Free Flight režimem, musí být dostatečně vybaveno pro to, aby mohlo vysílat svou identifikaci, výšku, polohu, rychlost a informaci o zamýšlené trase. Tyto informace jsou okolními letadly využívány k výpočtům, které zajistí dostatečné rozestupy, aby nedošlo k nebezpečnému sblížení při zachování optimální letové trajektorii.

Znázornění rozdílu mezi konveční trajektorií při využití letových cest a při využití Free Flight je na obrázku 2.10.



Obrázek 2.10. Letové cesty versus Free Flight.

Tato změna má ovšem dost zásadní dopad na fungování ATM. Pro prostory s free flight režimem dojde k decentralizaci systému. To způsobí, že při zvýšení počtu letadel v prostoru se zvýší nároky na palubní výpočetní zařízení, protože letadla již nebudou pouze pasivními elementy.

K fungování konceptu Free Flight je zapotřebí integrovat systém ASAS, který podporuje režim Self-Separation na paluby letadel.

ASAS může být provozován ve čtyřech základních módech. Awareness, Spacing, Separation a Self-Separation. ASAS musí zajistit několik funkcí. Zpracování přijatých dat z okolních letadel zajišťuje Tracker. O zobrazení informací o okolním provozu se stará CDTI. Predikce konfliktů v delším časovém horizontu zajišťuje PASAS. Zjištění konfliktní situace a její vyřešení zajišťuje CD&R.

2.6.1 Řešení konfliktu v současném systému ATC

Pokud nastane konflikt, řídící vybere letadlo, které tento problém vyřeší manévrováním. Mějme případ, kdy systém vyšle řídícímu na jeho obrazovku signál, že hrozí konflikt mezi letadly A a B. Řídící si je tedy vědom konfliktu díky rozhraní člověk-stroj. Řídící vybere letadlo A nebo B a vyšle posádce instrukce, které po úspěšném provedení posádkou vyřeší konflikt. Existuje několik událostí, které mohou narušit proces identifikace a odvrácení konfliktu.

- Radarový systém ukazuje nepřesně polohu nebo výšku letadla A nebo B.
- Řídící letového provozu nezaznamenal konfliktní situaci a výstrahu.
- Selhání rádia v jednom z letadel nebo na zemi.
- Posádka jednoho z letadel nerozuměla nebo špatně provedla instrukce, které měla obdržet od řídícího.

Pokud vybrané letadlo z nějakého důvodu konflikt neřeší dle instrukcí řídícího, existuje alternativní řešení a to nechat manévrovat letadlo druhé.

2.6.2 Řešení konfliktu systémem ASAS při Free Flight

Konflikt letadel se vyhodnocuje systémy na palubách letadel tak, že se určí 3D polohy letadel v okolním prostoru a také jejich vektory rychlostí. Systém ASAS pak vyhodnocuje, zda-li jsou letadla na kolozní dráze a případné konflikty řeší pomocí manévrů jednoho z letadel, ale i obou letadel naráz.

2.6.3 ACAS

ACAS je antikolizní systém, který pracuje nezávisle na pozemních zařízeních. Slouží jako „poslední záchrana" při hrozící srážce letadel ve vzduchu. Tento systém bude i při konceptu Free Flight zachován a bude pracovat nezávisle na ASAS.

Princip fungování ACAS spočívá v definování ochrané zóny kolem letadla. Při narušení této zóny systém ACAS vydá TA (Traffic Advisory), čili varování před existencí provozu v blízkosti letadla. ACAS první generace vydával pouze TA, ovšem s nástupem druhé generace umožňuje také vydání RA (Resolution Advisory), čili návrh řešení konfliktu ve vertikální rovině, kdy jednomu letadlu je nařízeno stoupat a jednomu klesat.

V souladu s nařízením komise (EU) č. 1332/2011, systémem ACAS II s protisrážkovou logikou verze 7.1, musí být od 1. prosince 2015 vybaveny všechny civilní letouny s turbínovým motorem mající maximální vzletovou hmotnost přesahující 5700 kg, nebo s maximální schválenou sedadlovou konfigurací pro více než 19 cestujících.

ACAS třetí generace umí navrhovat řešení konfliktu jak ve vertikální, tak v horizontální rovině.

TA a RA systém vydává vzhledem k času zbývajícímu do bodu největšího sblížení (CPA - Closest Point of Approach). Bod CPA je lokální minimum fyzické vzdálenosti mezi dvěma letadly (šikmá vzálenost).

Časy do CPA, při kterých jsou vydávány TA a RA, jsou definovány citlivostí (SL - Sensitivity Level). Čas do CPA pro TA se pohybuje od 20 do 48 sekund a pro RA od 15 do 35 sekund. Ve výškách od 0 do 1000 stop nad zemí se vydává pouze TA. [13][14][15][16]

2.7 Výběr 3D enginu

Pro tvorbu software pro návrh a optimalizaci letových prostorů a tratí jsem se rozhodl použít hotový 3D engine. Bylo by možné vytvořit si engine a nástoroje vlastní, ale vzhledem k tomu, že je dnes dostupných poměrně dost řešení třetích stran, které jsou velmi kvalitní a často poskytují licence zdarma, nejeví se tata možnost jako vhodná.

Dále se zaměřím na enginy, které se v převážné většině případů využívají ke tvorbě videoher a to z několika důvodů:

- Mám s těmito enginy zkušenost.
- Mnoho z nich lze využít zdarma (za určitých podmínek).
- Obecně jsou velmi pokročilé, co se týče grafických možností.
- Mají mnoho dostupných nástrojů.
- Jsou dobře dokumentovány.
- Mají velkou zákaznickou základnu, čili mnoho řešení lze nalézt na fórech, obchodech s assety atd.
- Často exitují již hotová řešení pro VR/AG.

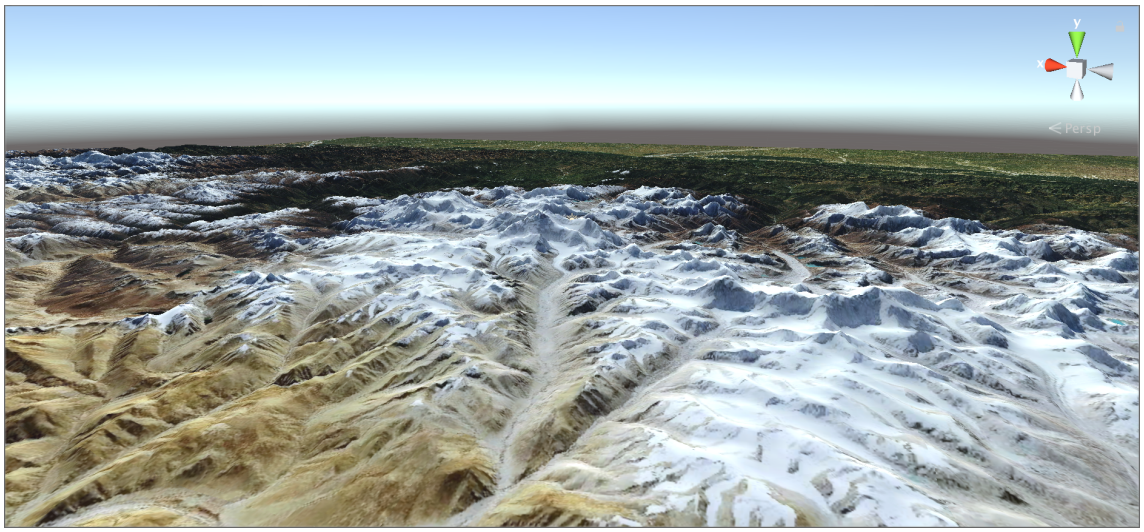
Také si kladu za cíl vybrat takové řešení, které bude splňovat mnou definované základní požadavky, které jsou:

- Podpora více platform (Windows, Mac, iOS, Android).
- Personal nebo education licence zdarma.
- Exitující framework pro integraci mapových podkladů.

Během rešerše jsem vyhodnotil jako tři nejvodnější kandidáty Unreal Engine 4, Unity3D (verze 5.x) a CryEngine V.

Všechny tři enginy splňují má kritéria. Po vyzkoušení všech kandidátů, kdy jsem se v každém ze zmíněných enginů pokusil vytvořit jednoduchý proof of concept, který spočíval v umístění mapy do scény a navádění objektu letadla po několika bodech, jsem se nakonec rozhodl zvolit engine Unity3D. Velkým plusem byla možnost využít framework Mapbox, který ideálně odpovídal mé představě o práci s mapovými podklady.

Ukázka scény Unity3D a Mapbox je na obrázku 2.11. [17][18]



Obrázek 2.11. Ukázka scény Unity3D a Mapbox.

2.8 Virtuální a augmentovaná realita

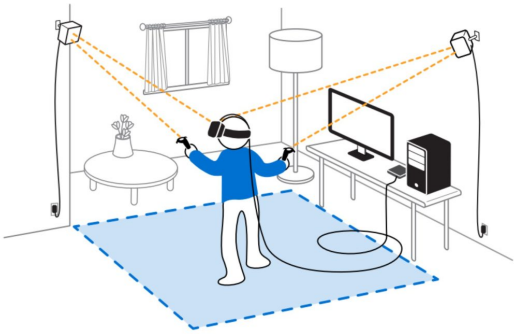
Pro navrhovaný software bude vhodné počítat s využitím virtuální a augmentované reality. Mělo by se jednat spíše o doplňkovou funkci programu, nikoli vyžadované součásti. Technologie VR a AG jsou poměrně nové (berme v potaz technologie použitelné do praxe). Výhodou by mělo být opravdové 3D zobrazení situace, která s využitím 2D obrazovek není možná.

Momentálně exituje několik zařízení pro virtuální realitu. Níže je popis těch, které jsou volně dostupné jak uživatelům, tak vývojářům.

2.8.1 HTC Vive

Vyvinut společnostmi HTC a Valve Corporation. Vizualizace HTC Vive je na obrázku 2.12 a 2.13. V současné době nejspíše nejpokročilejší a také nejdražší komerčně dostupné řešení pro VR. Při koupi zákazník obdrží helmu, dva ovladače do ruky a dva lighthouse vysílače pro snímání pohybu helmy a ovladačů v prostoru. S HTC Vive lze tedy snímát i pohyb uživatele v prostoru, který je s aktuální verzí zařízení omezen na čtverec 4,6 krát 4,6 metrů. Na helmě je také umístěna kamera, díky které lze do obrazu helmy přenášet i reálný obraz. Zařízení disponuje dvěma obrazovkami s rozlišením 1080x1200. Celkový obraz pro obě oči je tak tvořen 2160x1200 pixely a poměr stran je 9:5. Obnovovací frekvence zařízení je 90Hz a zorné pole činí 110 stupňů. Helma má v sobě také zabudované senzory pro snímání pohybu a to akcelerometr, gyroskop a laserové snímače polohy, které spolupracují s lighthouse vysílači a slouží k počátečnímu určení polohy v prostoru. Další pohyb po úvodní kalibraci je vypočítáván pomocí akcelerometru a gyroskopu. V prostoru vytyčeném vysílači se může pohybovat více zařízení Vive. Ovladače do ruky jsou bezdrátové, zatímco helma vyžaduje připojení kabelem. Ovladače disponují několika tlačítky, trackpady a také jsou schopny poskytovat zpětnou vazbu pomocí vibrací. V současnosti už je na trhu nová verze HTC Vice Pro, která má oproti první verzi větší rozlišení obrazovek a to 1440x1600 pixelů pro jednu obrazovku.

Obrázek 2.12. HTC Vive.



Obrázek 2.13. HTC Vive v prostoru.

2.8.2 Oculus Rift

Vyvinut společností Oculus Rift, kterou vlastní společnost Facebook. Vizualizace Oculus Rift je na obrázku 2.14. Jde o první VR zařízení, které lze považovat za moderní. Balení neobsahuje žádný ovladač. Pouze helmu a snímací kameru. Obrazovky v helmě mají rozlišení 1080x1200 a obnovovací frekvenci 90Hz. Zorné pole je 110 stupňů. Senzor pro snímaní pohybu je pouze magnetometr. Snímání v prostoru funguje narozdíl od HTC Vive na principu vysílání infračervených paprsků z helmy, které následně snímá kamera. Prostor pro pohyb je menší než v případě HTC Vive, ale lze jej rozšířit přidáním další kamery. Set lze rozšířit o další snímací kameru a ovladače Oculus Touch. Tyto ovladače jsou bezdrátové a jejich pohyb v prostoru se zjišťuje stejným způsobem jako pohyb helmy. Ovladač také podporuje jednoduchá gesta a detekuje sílu stisku. Podobně jako u Vive má několik tlačítek a poskytuje zpětnou vazbu pomocí vibrací. Nabízí také analogovou páčku.

Obrázek 2.14. Oculus Rift.

2.8.3 Playstation VR

Vyvinut společností Sony Interactive Entertainment. Vizualizace Playstation VR je na obrázku 2.15. Toto zařízení nelze propojit s počítačem, ale pouze s konzolí Sony Playstation 4. Tím pádem jeho využití není možné pro navrhovaný software. Helma disponuje dvěma obrazovkami s rozlišením 960x1080. Dohromady tedy poskutuje obraz o rozlišení 1920x1200 s poměrem stran 16:9. Obnovovací frekvence je 120Hz a zorné pole 100 stupňů. Na helmě je umístěno 9 LED diod pro sledování polohy za pomocí kamery (Playstation Camera, které je nutná pro fungování systému a připojuje se přímo do konzole PS4). Zároveň helma obsahuje akcelerometr, gyroskop a magnetometr. Do setu lze také přidat bezdrátové ovladače Playstation Move, který využívá stejného principu pro určení polohy v prostoru jako helma. Ovladače nabízí několik tlačítek, analogový trigger a poskytují zpětnou vazbu pomocí vibrací.

2.8.4 Google Daydream

Vyvinut společností Google. Vizualizace Google Daydream je na obrázku 2.16. Jedná se o řešení s využitím mobilního telefonu, který se vloží do helmy. Podporovány jsou mobilní telefony s operačním systémem Android verze 7.1 a vyšší. Výhodou je nižší pořizovací cena (zvlášť v situaci, kdy uživatel již vlastní mobilní telefon). Systém je tvořen helmou, do které se vkládá telefon, dvěma čočkami v helmě, které slouží k převodu obrazu z mobilního zařízení a jednoduchého bezdrátového ovladače. Určení polohy systému v prostoru lze díky senzorům v mobilním zařízení. Samotná helma žádnými senzory nedisponuje. Ovladač obsahuje dvě tlačítka a dotykovou plochu.

Obrázek 2.15. Playstation VR.

Obrázek 2.16. Google Daydream.

2.8.5 Ostatní VR systémy

Existuje ještě několik dalších zástupců. Ty se od výše popsaných většinou liší pouze detaily, jako je rozlišení obrazovek, ale podstata fungování je velmi podobná.

Nejdostupnější řešení pro vyzkoušení VR je Google Cardboard, což jsou brýle z kartonu s dvojicí čoček, které si uživatel poskládá sám. Do těchto brýlí se pak vloží mobilní zařízení, které podporuje režim VR. [19] [20] [21] [22]

2.8.6 Augmentovaná realita

Augmentovaná realita je založena na umístění virtuálních předmětu v reálném světě. Samozřejmě je reálný svět promítán do elektronického zařízení, které zobrazuje kopii reality, do které zasazuje virtuální předměty. K tomu je zapotřebí využívat senzory a algoritmy pro vnímání okolního světa a jeho následného využití ve virtuálním prostoru.

Pro konkrétní aplikaci v navrhovaném software počítám s využitím pohybu zařízení v reálném 3D prostoru a schopnosti rozpoznávat rovné plochy, jako jsou například stoly. Na tyto rovné plochy bude možné umístit virtuální svět, který bude zobrazovat vybranou oblast na zemi včetně terénu společně se simulovaným leteckým provozem, letišti, vzdušnými prostory atd.

Jsou dvě možnosti, jak uživateli poskytnout takový zážitek. Jeden ze způsobů je využít mobilního zařízení s obrazovkou, fotoaparátem a senzory, které jsou vyžadovány pro navigaci zařízení v prostoru. Virtuální svět je promítán na obrazovku takového zařízení a uživateli zprostředkovává virtuální svět. V tomto případě se jedná čistě o augmentovanou realitu. Vizualizace tohoto řešení je na obrázku 2.17

Obrázek 2.17. Augmentovaná realita bez helmy.

Druhá možnost je využít zařízení pro virtuální realitu (brýle), které si uživatel nasadí a vnímá už pouze „svět", který mu zobrazuje toto zařízení. Tato možnost uživatele více „zasazuje" do virtuálního světa, ale na druhou stranu s sebou může přinést i několik komplikací.

Zásadní problém je nevolnost (motion sickness), která se projevuje během používání VR. Míra nevolnosti a čas pro nástup je individuální, ale řádově se pohybuje v desítkách minut. Výrobci se snaží tento problém co nejvíce eliminovat. Obecně platí, že větší rozlišení a obnovovací frekvence může pomoci. Nejvíce ovšem záleží na VR zážitku jako takovém. Například ve hrách, kde se postava, do které se „vžije" uživatel, nehýbe, ale pouze se rozhlíží, nepřivozuje nevolnost tak rychle a intenzivně jako akční hry, kde se postava neustále hýbe v prostoru. Důvod, proč tomu tak je, ještě není dokázán, dá se

ale předpokládat, že je to vinou rozporu toho, co uživatel vidí a toho, co vníma jeho vnitřní ucho.

Spojení VR brýlí a AG, kdy je promítán jak zprostředkovaný reálný svět, který zaznamenává kamera na helmě, tak přidané virtuální objekty, by ovšem tyto problémy působit nemělo, protože uživatel uvidí reálný svět, který bude zároveň vnímat všemi ostatními smysly, nejenom zrakem. Vizualizace takového řešení je na obrázku 2.18

Obrázek 2.18. Augmentovaná realita s helmou.

2.9 Vize projektu

Cílem práce je navrhnout software, který bude sloužit k tvorbě a úpravě letových tratí a prostorů, potažmo sektorů ATC. Mělo by se jednat o intuitivní systém, který uživateli umožní optimalizovat podobu vzdušného prostoru s co možná nejmenším úsilím. Bude možné užívat jak systémy a metody, které jsou již zavedeny v praxi, tak nové postupy, které jsou zatím ve fázi připravy. Software potom bude sloužit jako pomůcka pro zavedení nových postupů a metod do praxe s co možná nejvyšší mírou optimalizace.

2.10 Funkční požadavky

Funkční požadavky jsou sestaveny na základě konzultací se zadavatelem.

- Systém bude umožňovat tvorbu a úpravu vzdušných prostorů.
- Systém bude umožňovat tvorbu a úpravu sektorů.
- Systém bude umožňovat tvorbu a úpravu traťových bodů.
- Systém bude umožňovat tvorbu a úpravu letištních drah.
- Systém bude umožňovat simulaci leteckého provozu v reálném čase.
- Systém bude umožňovat tvorbu a úpravu scénářů leteckého provozu.
- Systém bude umožňovat vstup uživatele do simulace provozu v reálném čase.
- Systém bude umožňovat ukládání a načítání již vytvořených vzdušných prostorů, sektorů, traťových bodů a scénářů leteckého provozu.
- Systém bude umožňovat importování dat z leteckých map.
- Systém bude umožňovat kooperaci více uživatelů v jedné instanci simulace.

- Systém bude umožňovat zobrazení simulace na VR/AG zařízení.
- Systém bude umožňovat přepínání mezi režimy simulace a editace.
- Systém bude umožňovat zobrazení situace ve 2D i 3D.
- Systém bude umožňovat použití virtuálního řídícího, který bude navádět letadla dle definovaného scénáře.
- Systém bude umožňovat import letecké mapy AIXM ve formátu xml.

Jednotlivé funkční požadavky analyzuji blíže a určuji dílčí požadavky. Při analýze dále detekuji další požadavky, které vyplývají z podmínek pro návrh funkčních požadavků. [23]

2.10.1 Tvorba a úprava vzdušných prostorů

Vzdušný prostor je definován jako 3D objekt v prostoru. Má tedy jednoznačně určený tvar. Tvar vzdušného prostoru bude popsán 3D souřadnicemi. Pro určení souřadnic bude nutné definovat prostor s kartézskou soustavou souřadnic a osami x, y a z. Vzdušný prostor bude potřeba umisťovat na mapu dle jeho geo souřadnic.

Z toho vyplývá, že bude potřeba připravit systém, který bude tvořen mapou, na kterou bude možné umisťovat tvořené vzdušné prostory.

Tvorba prostoru by měla probíhat následovně:

1. Uživatel vybere body na mapě
2. Mezi těmito body vzniknou vazby, které budou tvořit 2D tvar prostoru (při pohledu zvrchu).
3. Tvar spojnic může být upraven tak, aby spojnice nebyly úsečky, ale tvořily křivku.
4. Uživatel označí první bod, a tím dokončí tvorbu tvaru prostoru.
5. Uživatel vybere kategorii vzdušného prostoru.
6. Uživatel nastaví vzdušnému prostoru spodní a horní hranici (výšky nad zemí).
7. Uživatel potvrdí tvorbu sektoru, a tím je sektor uložen.

Upravovat již vytvořený prostor půjde po vybrání (označení) daného prostoru. Objeví se nabídka pro editaci, kde bude zároveň možnost prostor odstranit.

Vzdušný prostor může být tvořen více podprostory. Toho se docílí pomocí seskupování sektorů.

2.10.2 Tvorba a úprava sektorů

Sektor je definován jako 3D objekt prostoru. Stejně jako u vzdušných prostorů bude tvar popsán součadnicemi v kartézské soustavě. Souřadnice sektorů budou také vytvářeny pomocí geo souřadnic. Spodní a horní hranice bude zadána pomocí vyšek nad zemí.

Tvorba sektorů bude mít stejný průběh jako tvorba vzdušných prostorů až na několik drobných odlišností. U sektorů se nezadává kategorie, nýbrž zodpovědné stanoviště ATC.

Sektory lze seskupovat do větších sektorů, které pak řídí pouze jeden řídící letového provozu.

Upravovat a mazat sektory lze stejným způsobem jako vzdušné prostory.

2.10.3 Tvorba a úprava traťových bodů

Traťový bod bude určen zeměpisnou šířkou a délkou. Bude platný pro jakoukoli výšku nad zemí.

Vytvořený bod půjde vybrat a následně upravovat nebo odstranit v nabídce pro editaci.

Při tvorbě a editaci bude možné zadat jméno, které se bude skládat z pěti znaků latinské abecedy.

2.10.4 Tvorba a úprava letištních drah

Letištní dráha bude určena dvěma geo souřadnicemi a názvy drah. Systém nebude vyžadovat další údaje, jako nastavení procedur přesného či nepřesného příblížení. Letadlo bude moci na dráze přistát, pokud se bude přibližovat ve směru dráhy, popřípadě s menší odchylkou. Rychlost letadla na přistání bude řesit systém automaticky a bude průběžně měnit rychlost letadla tak, aby bylo letadlo schopné přistát. To bude ovšem podmíněno tím, zda-li bude taková úprava rychloti možná vzhledem k parametrům letadla a potřebné trajektorie. Pokud dojde ke konfliktu a systém nenalezne takové řešení, aby mohlo letadlo přistát na dráze, bude uživatel varován a zahájí se postup nezdařeného přiblížení. Postup nezdařeného přiblížení bude v rámci simulace zjednodušen a na všech dráhách bude stejný.

2.10.5 Simulace leteckého provozu v reálném čase

Software bude obsahovat režim simulace v reálném čase. Při přepnutí do tohoto režimu se začnou hýbat objekty, které jsou definovány jako pohyblivé. Letadla v systému budou měnit polohu dle definovaných scénářů, popřípadě po trajektorii určené kurzem a rychlostí letadla. V tomto režimu bude možné letadla ovládat pomocí příkazů:

- Změna kurzu.
- Změna rychlosti.
- Změna výšky letu.
- Navigovat na traťový bod.
- Přistát na vybrané letiště.
- Zahájit vyčkávání dle vybraného holding patternu.
- Předat letadlo jinému řídícímu.

Na aktivních letištních drahách bude v režimu simulace možné přidat do systému nové letadlo, které následně vzlétne z vybrané dráhy.

V režimu simulace nebude možné upravovat vzdušné prostory, sektory, traťové body a letištní dráhy.

2.10.6 Tvorba a úprava scénářů leteckého provozu

Software bude umožňovat tvorbu scénáru podle kterých bude následně simulovaný letecký provoz. Spuštění simulačního bodu definuje čas T0. Plánované události ve scénářích se budou spouštět buď dosažením času T0 + čas spuštění události, nebo sekvenční návazností událostí.

Každá entita v systému bude obsahovat manager scénáře a bude ji možné definovat události vzhledem možnostem dané entity.

V průběhu jednoho aktivního scénáře bude také možné tvořit a aktivovat další scénáře.

Příklady možných scénářů:

- Na traťovém bodě se vytvoří nové letadlo s volacím znakem OK123 v čase 3 minuty a 10 sekund, které po vytvoření začne užívat scénář „scenar1".
- Na letišti Ruzyně se vytvoří nové letadlo s volacím znakem OK321 v čase 1 minuta a 2 sekundy, které vzlétne v čase 2 minuty a 30 sekund. Po vzletu se na letadle OK321 aktivuje scénář „scenar2".
- Letadlo po dosažení traťového bodu MEGAN dále pokračuje na traťový bod OSKAR.

Každý scénář musí mít definované odkazy na entity v systému, kterých bude využívat po dobu svého běhu.

2.10.7 Vstup uživatele do simulace provozu v reálném čase

Software bude umožňovat zásahy uživatele do režimu simulace. Bude se jednat o zásahy do scénářů provozu, zadávání příkazů aktivním letadlům a letištím. V tomto režimu nebude možné měnit vzdušné prostory, sektory, umístění letištních drah atd. Tyto vstupy budou zadávány pomocí editačních nabídek pro jednotlivé entity.

2.10.8 Ukládání a načítání již vytvořených vzdušných prostorů, sektorů, traťových bodů a scénářů leteckého provozu

Software bude umožňovat zachování vytvořených prostorů, sektorů, traťových bodů a scénářů pomocí ukádání do externích souborů či databáze.

Prostory, sektory a traťové body budou ukládány do projektů. Projekty budou soubory dat, které po načtení do systému vytvoří prostory, sektory a traťové body tak, jak byly uloženy uživatelem. Projekt bude definován unikátním názvem a součástí uložených dat v projektu bude také informace o umístění centra a rozsahu lokality pro editaci a simulaci. Tato lokalita se objeví po načtení projektu, posléze ale bude možné lokalitu změnit.

2.10.9 Importování dat z leteckých map

Software bude umožňovat načtení a následné zobrazení vzdušných prostorů a traťových bodů za pomoci importu dat z leteckých map. Předpokládaný formát leteckých map je AIXM jako soubor xml. Importovaná data budou přetrasformována do struktury projektu a následně je půjde spolu s otatními daty projektu ukládat, měnit a načítat.

2.10.10 Kooperaci více uživatelů v jedné instanci simulace

Software bude umožňovat zapnutí režimu serveru. V tomto režimu bude možné se k instanci, která poběží na jednom uživatelském stroji, připojit z jiného stroje (tato instance bude nazývána klient), který po připojení uvidí stav simulace a editace na serveru. Klient bude standartně v režimu pozorovatele a nebude moci do simulace zasahovat. Následně bude na straně serveru možné zvolit klientovi jinou roli.

Možné role budou:

- Řidící sektoru.
- Editor.
- Pilot.
- Observer.

Řidícímu sektoru bude také přidělen sektor a bude moci ovládat letadla, která do jeho sektoru spadají. Také mu bude umožněno předat jím ovládaná letadla jinému řídícímu ve chvíli, kdy se letadlo bude blížit opuštění sektoru.

Editor bude moci v režimu editace tvořit, upravovat a mazat vzdušné prostory, sektory, trať ové body, scénaře leteckého provozu a dráhy letišť. V režimu simulace mu bude umožněno ovládat všechny entity v simulaci stejně jako uživateli na straně serveru.

Pilotovi bude umožněno ovládání přiděleného letadla podle pokynů, které nebudou předány pomocí systému, ale verbálně (popřípadě jinými prostředky jako rádiem, telefonem atd.).

Observerovi bude umožněno pouze sledování simulace.

2.10.11 Zobrazení simulace na VR/AG zařízení

Software bude umožňovat spuštění v módu VR, AG a VR+AG. V tomto módu bude možné se připojit jako klient k již spuštěné instanci v režimu server a bude mu přidělena role Observer.

V módu VR bude uživatel sledovat simulaci v kompletně virtuálním prostředí a to pomocí zařízení poskytujícího funkci virtuální reality.

V módu AG bude uživatel sledovat simulaci zprostředkovaně pomocí zařízení poskytujícího funkci augmentované reality. Zde se bude jednat o mobilní zařízení. Uživatel po zapnutí aplikace v módu AG nejříve detekuje rovnou plochu vhodnou pro umístění simulovaného „světa” (například prázdný stůl) a následně se mu na vybranou plochu začne promítat simulace.

V módu VR+AG dojde ke spojení obou výše zmíněných módů. Postup bude stejný jako při módu AG s tím rozdílem, že bude uživatel zároveň pro sledování virtuálního světa využívat zařízení poskytujícího funkci virtuální reality.

2.10.12 Přepínání mezi režimy simulace a editace

Systém bude umožňovat přepínání mezi režimem simulace a editace.

Při přepnutí z režimu simulace do režimu editace se „zastaví” čas a bude možné tvořit, upravovat a mazat vzdušné prostory, sektory, trať ové body, letištní dráhy a výchozí scénáře leteckého provozu.

Při přepnutí z režimu editace do režimu simulace se „spustí” čas a bude možné upravovat scénaře leteckého provozu, zadávání příkazů letadlům a předávání letadel mezi řídícími.

2.10.13 Zobrazení situace ve 2D i 3D

Systém bude umožňovat zobrazení ve 2D pohledu „shora", kdy bude vidět mapa a všechny aktivní entity budou vykresleny pomocí 2D obrazců. Také bude umožňovat přepnutí do 3D zobrazení, kde bude mapa vykreslena jako 3D objekt pomocí výškové mapy a aktivní entity budou umístěny v prostoru i vzhledem k jejich výšce nad zemí.

2.10.14 Použití virtuálního řídícího, který bude navádět letadla dle definovaného scénáře

Systém bude umožňovat pro každy sektor aktivaci virtuálního řídícho letového provozu. Díky tomu budou letadla v sektoru naváděna systémem automaticky bez zásahu uživatele vzhledem k jejich scénářům. V budoucnu se počítá také s využitím umělé inteligence (pravděpodobně neuronových sítí), která bude sloužit k výuce virtuálních řídících za pomocí dat ze simulací, které budou řízeny uživatelem.

2.10.15 Import letecké mapy AIXM ve formátu xml

Systém bude umožňovat importování leteckých map AIXM pomocí souborů s příponou xml. Tato data budou transformavána tak, aby byla využita v sytému k vytvoření letových prostorů a traťových bodů.

2.11 Nefunkční požadavky

Nefunkční požadavky jsou sestaveny autorem této práce s využitím zažitých praktik, které jsou popsány ve zdrojích [24] [25] [26].

Motivace k vytvoření nefunkčních požadavků je vyvinout kvalitní a stabliní systém a nastavení kriterií, podle kterých se bude kvalita měřit. [23]

Pro navrhovaný software definuji tyto nefunkční požadavky:

- Výkon.
- Udržitelnost.
- Spolehlivost.
- Dostupnost.
- Rozšiřitelnost.
- Bezpečnost.

2.11.1 Výkon

Specifikovat přesné požadavky pro výkon je obtížné. Výkon se často měří jako doba reakce na uživatelský požadavek, jako počet zpracovaných transakcí za časový interval, počet rendrovaných snímků za sekundu atd. Tyto parametry jsou závislé na použitém hardware a schopnosti vývojaře optimalizovat zdrojový kód aplikace tak, aby byl hardware optimálně využitý.

Požadavek na výkon systému definuji tak, že je nezbytné, aby systém reagoval na vstup uživatele s takovým prodlením, aby uživatel nenabyl dojmu, že aplikace, nevykonává jeho příkazy okamžitě.

Ve vývoji se bude počítat s tím, aby systém takto fungoval na stroji střední třídy dostupným v době vývoje.

Pro určení výkonnosti aplikace tedy bude zapotřebí testování reálných uživatelů a následného vyhodnocení jejich zpětné vazby.

2.11.2 Udržitelnost

Udržitelnost je závislá na schopnosti opravení nedostatků systému tak, aby oprava neovlivnila jinou část systému. K udržitelnosti přispívá vhodné dělení funkčních celků do komponent, které jsou na sobě minimálně (ideálně vůbec) závislé.

Po celou dobu návrhu bude nutné brát udržitelnost jako důležitý parametr pro rozhodování o architektuře systému.

2.11.3 Spolehlivost

Systém musí zaručit integritu vytvořených dat. Data systémem vytvořená budou uložena buď v souborech nebo v databázi. V obou případech je nutné udržovat data duplicitně až do chvíle, kdy i přes to, že data nejsou duplicitní, nedojde k jejich ztrátě při neočekávané události.

2.11.4 Dostupnost

Dostupnost bude mít v navrhovaném systému nižší prioritu. Nebude se jednat o komplexní systém, který by vyžadoval neustálý provoz, ale o aplikaci, která bude spuštěna pouze pro účely dočasné práce s ní. Nebude tedy třeba komponenty systému replikovat, aby byla zajištěna stálá dostupnost. V případě nečekané události, jako je selhání hardware, software, sítě atd., se aplikace stane nedostupnou.

2.11.5 Rozšiřitelnost

Rozšiřitelnost bude jedním z klíčových nefunkčních požadavků. Přidávání nové funkcionality do systému nebo modifikace stávající funkcionality se bude považovat za zcela běžný zásah.

Pro dosažení dobré rozšiřitelnosti bude zapotřebí, co nejmenší provázání komponent systému, častému využívání rozhraní, vysoká úroveň zapouzdření jednotlivých komponent a kvalitní objektový model.

Systém bude mít také oddělené části pro výpočetní část simulace a zobrazovací části, aby bylo možné nahradit aktuální 3D engine s co nejnižším počtem zásahů do systému.

2.11.6 Bezpečnost

Systém není navrhován pro fungování v síti, která je volně dostupná ze sítě internet. Přesto bude nutné dodržet standardy bezpečnosti, aby nedošlo k únikům dat při síťové komunikaci, případně k podvržení dat falešných.

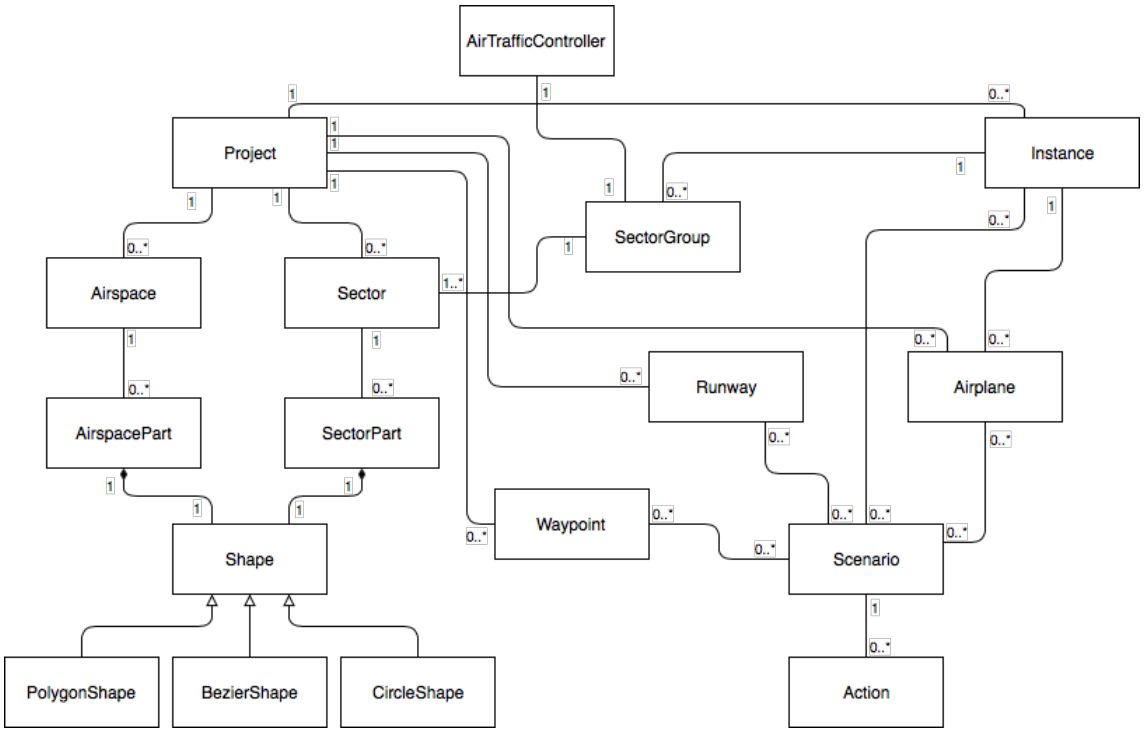
Systém bude pro veškerou síťovou komunikaci využívat pouze zabezpečeného protokolu SSL. Soubory a databáze nebudou šifrovány. Systém bude vyžadovat stroj, který

je zabezpečený uživatelským heslem k tomu, aby se data na něm uložená, považovala za bezpečná. Jinými slovy bezpečnost uložených dat bude delegována na operační systém a uživatele.

Kapitola 3
Návrh

3.1 Diagram tříd

Na základě funkčních požadavků je vytvořen UML diagram tříd, který je na obrázku 3.1. Tyto třídy budou v systému tvořit datový model. Pro přehlednost jsou na obrázku vynechány atributy a funkce tříd, ty jsou pro každou třídu detailně popsány dále v textu.



Obrázek 3.1. Diagram tříd.

Další dvě třídy, které nejsou obsaženy v diagramu jsou třída Node, AirplaneType, AirpsaceClass, AirspaceType. Třída Node je vynechána z důvodu přehlednosti diagramu, navíc se jedná o pomocnou třídu, která v sobě pro zjednodušení zdrojového kódu a znovupoužitelnosti nese informaci o třech atributech, které by jinak musely být vždy definovány jednotlivě. Třídy AirplaneType, AirspaceClass a AirspaceType jsou vynechány, protože souvisí s tématem verzování aplikace.

3.2 Popis atributů a funkcí tříd

V této podkapitole detailně popisuji jednotlivé třídy.

Popis jednotlivých atributů je dle šablony Název atributu - popis, typ, úroveň přístupu, GET — SET (get nebo set značí implicitní vytvoření funkce get nebo set pro daný atribut).

Popis jednotlivých funkcí je dle šablony Název funkce - popis, (parametry funkce), návratový typ, úroveň přístupu. Parametry funkce jsou zapsány za sebou oddělené čárkou a ve formátu název parametru - typ parametru.

Poznámka: „Pokud u atributu není uvedeno GET, SET nebo, počítá se s vytvořením funkcí GET i SET."

Poznámka: „Pokud u atibutu nebo funkce není uvedena úroveň přístupu, počítá se s úrovní privátní."

3.2.1 Node

Node slouží k uchování atributů určujících polohu v soustavě.

Atributy:

- latitude - zeměpisná šírka, float
- longitude - zeměpisná délka, float
- altitude - výška nad zemí (v metrech), float

Funkce:

- setAltitudeInFt - přenastavý altitude pomocí přepočtu ft na metr, (altitudeInFt - float), void, veřejný
- setAltitudeInFl - přenastavý altitude pomocí přepočtu Fl na metr, (altitudeInFl - float), void, veřejný

Pokud není výška definována, pak je ignorována a počítá se s tím, že uzel je platný ve všech výškách nad zemí.

3.2.2 Project

Project zapouzdřuje informace o entitách a jejich rozmístění v čase T0.

Atributy:

- name - název projektu, string
- mapOrigin - počáteční bod na mapě při otevření projektu, Node
- airspaceList - list referencí na vzdušné prostory v projektu, List¡Airspace¿
- sectorList - list referencí na sektory v projektu, List¡Sector¿
- runwayList - list referencí na přistávací dráhy v projektu, List¡Runway¿
- waypointList - list referencí na trať ové body v projektu, List¡Waypoint¿
- airplaneList - list referencí na letadla v projektu, List¡Airplane¿

3.2.3 Instance

Instance zapouzdřuje informace o rozmístění entit v jiném čase než T0.

Atributy:

- name - název instance, string
- currentTime - uběhnutý čas od času T0 (v milisekundách), float
- airplaneList - list referencí na letadla v instanci, List¡Airplane¿
- scenarioList - list referencí na scénáře v instanci, List¡Scenario¿
- sectorGroupList - list referencí na skupiny sektorů v instanci, List¡SectorGroup¿
- projectName - unikátní jméno projektu ke kterému je scénář přiřazen, string

3.2.4 Airspace

Airspace uchovává informace o vzdušném prostoru a obsahuje odkazy na jeho části.

Atributy:

- airspaceClass - třída vzdušného prostoru, enum¡AirspaceClass¿
- airspaceType - typ vzdušného prostoru, enum¡AirspaceType¿
- airspacePartList - list referencí na části vzušného prostoru, List¡AirspacePart¿

3.2.5 AirspacePart

AirspacePart obsahuje informace o jedné z částí tvořicí vzdušný prostor.

Atributy:

- airspace - reference na vzušný prostor, do kterého spadá daná část, Airspace
- lowerAltitude - výška nad zemí, kde začíná část vzdušného prostoru (v metrech), float
- upperAltitude - výška nad zemí, kde končí část vzdušného prostoru (v metrech), float
- shape - 2D tvar části vzdušného prostoru (při pohledu zhora), Shape

Funkce:

- setLowerAltitudeInFt - přenastavý lowerAltitude pomocí přepočtu ft na metr, (altitudeInFt - float), void, veřejný
- setUpperAltitudeInFt - přenastavý upperAltitude pomocí přepočtu ft na metr, (altitudeInFt - float), void, veřejný
- setLowerAltitudeInFl - přenastavý lowerAltitude pomocí přepočtu flight levelu na metry, (altitudeInFl - float), void, veřejný
- setUpperAltitudeInFl - přenastavý upperAltitude pomocí přepočtu flight levelu na metry, (altitudeInFl - float), void, veřejný

3.2.6 Shape

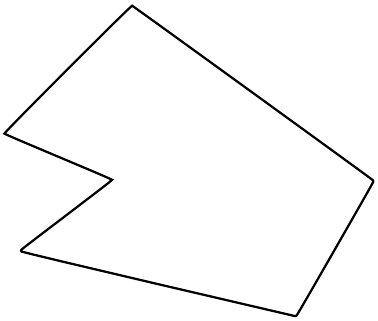
Shape je abstraktní třída od které jsou odděděny třídy PolygonShape, BezierShape a CircleShape.

Atributy:

- nodeList - list referencí na jednotlivé uzly polygonu, List¡Node¿

Systém bude pracovat pouze s polygony, u kterých se žádné dvě usečky nebudou křížit.

Příklad takového polygonu je na obrázku 3.2.



Obrázek 3.2. Polygon.

3.2.7 BezierShape

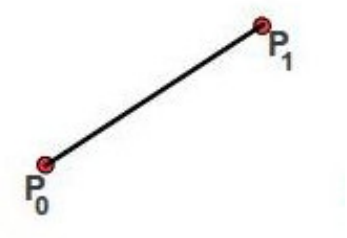
BezierShape slouží k uchování informace o tvaru objektu, který je popsán pomocí Bézierovy křivky.

Atributy:

- bezierCurveType - typ bézierovi křivky, enum¡BezierCurveType¿
- nodeList - list referencí na body, které udávají tvar bézierovi křivky, List¡Node¿

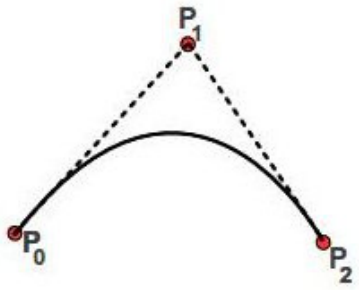
Systém bude podporovat lineární, kvadratické a kubické Bézierovo křivky. Pro určení typu křivky bude sloužit enumerátor s označením třídy BezierCurveType. Každý typ křivky vyžaduje jiný sled bodů v atributu nodeList. Tyto body slouží k následnému vykreslení křivky ve 2D, či 3D prostoru.

Lineární Bézierovo křivka je znázorněna na obrázku 3.3.

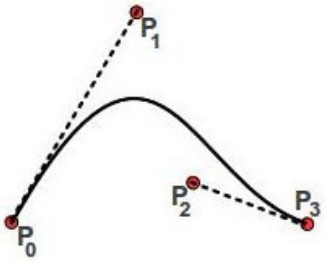


Obrázek 3.3. Lineární Bézierova křivka.

Kvadratická Bézierova křivka je znázorněna na obrázku 3.4.
Kubická Bézierova křivka je znázorněna na obrázku 3.5.



Obrázek 3.4. Kvadratická Bézierova křivka.



Obrázek 3.5. Kubická Bézierova křivka.

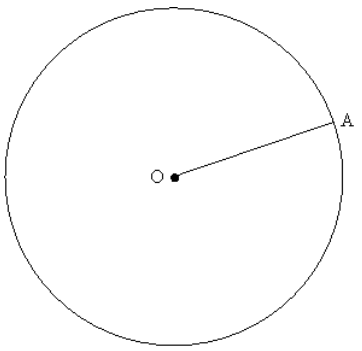
3.2.8 CircleShape

CircleShape slouží k uchování informace o tvaru objektu, který je popsán pomocí kruhu.

Atributy:

- centerNode - reference na střed kruhu, Node
- radius - poloměr kruhu, float

Kruhový tvar popsaný středem a poloměrem je znázorněn na obrázku 3.6.



Obrázek 3.6. Kruhový tvar.

3.2.9 Sector

Sector uchovává informace o sektoru ACC a obsahuje odkazy na jednotlivé části, které ho tvoří.

Také odkazuje na skupinu sektorů, do které patří a kterou řídí právě jeden řídící.

Atributy:

- name - název sektoru, string
- sectorGroup - reference na skupinu sektorů, do které daný sektor spadá, SectorGroup
- airplaneList - list referencí na letadla, která aktuálně spadají do dané instance sektoru, List¡Airplane¿
- sectorPartList - list referencí na části sektoru, které tvoří danou instanci sektoru

3.2.10 SectorPart

SectorPart obsahuje informace o jedné z částí tvořících sektor.

Atributy:

- sector - reference na sektor, do kterého spadá daná část, Sector
- name - název části sektoru, string
- lowerAltitude - výška nad zemí, kde začíná část sektoru (v metrech), float
- upperAltitude - výška nad zemí, kde končí část sektoru (v metrech), float
- atcType - služba ATC, která má kontrolu nad částí sektoru, AtcType
- shape - 2D tvar části sektoru (při pohledu zhora), Shape

Funkce:

- setLowerAltitudeInFt - přenastavý lowerAltitude pomocí přepočtu ft na metr, (altitudeInFt - float), void, veřejný
- setUpperAltitudeInFt - přenastavý upperAltitude pomocí přepočtu ft na metr, (altitudeInFt - float), void, veřejný
- setLowerAltitudeInFl - přenastavý lowerAltitude pomocí přepočtu flight levelu na metry, (altitudeInFl - float), void, veřejný
- setUpperAltitudeInFl - přenastavý upperAltitude pomocí přepočtu flight levelu na metry, (altitudeInFl - float), void, veřejný

3.2.11 AtcType

Enumerátor pro určení typu služby ATC.

Enumy:

- Tower
- Approach
- Departure
- Director
- Control
- Info

3.2.12 AirTrafficController

AirTrafficController obsahuje informace o řídícím letového provozu.

Atributy:

- type - typ řídícího letového provozu, ControllerType

3.2.13 ControllerType

Enumerátor pro určení typu řídícího letového provozu.

Enumy:

- Local
- Remote
- Virtual

3.2.14 Waypoint

Waypoint obsahuje informace o traťovém bodu.

Atributy:

node - geolokace bodu, Node name - jméno traťového bodu, string

3.2.15 Runway

Runway obsahuje informace o přistávací dráze letiště.

Atributy:

- node1 - jeden z krajních bodů dráhy, Node
- node2 - druhý krajní bod dráhy, Node
- name1 - název dráhy ze směru node1, string
- name2 - název dráhy ze směru node2, string
- scenario - scénář akcí, Scenario

3.2.16 Airplane

Airplane obsahuje informace o letadle v systému.

Atributy:

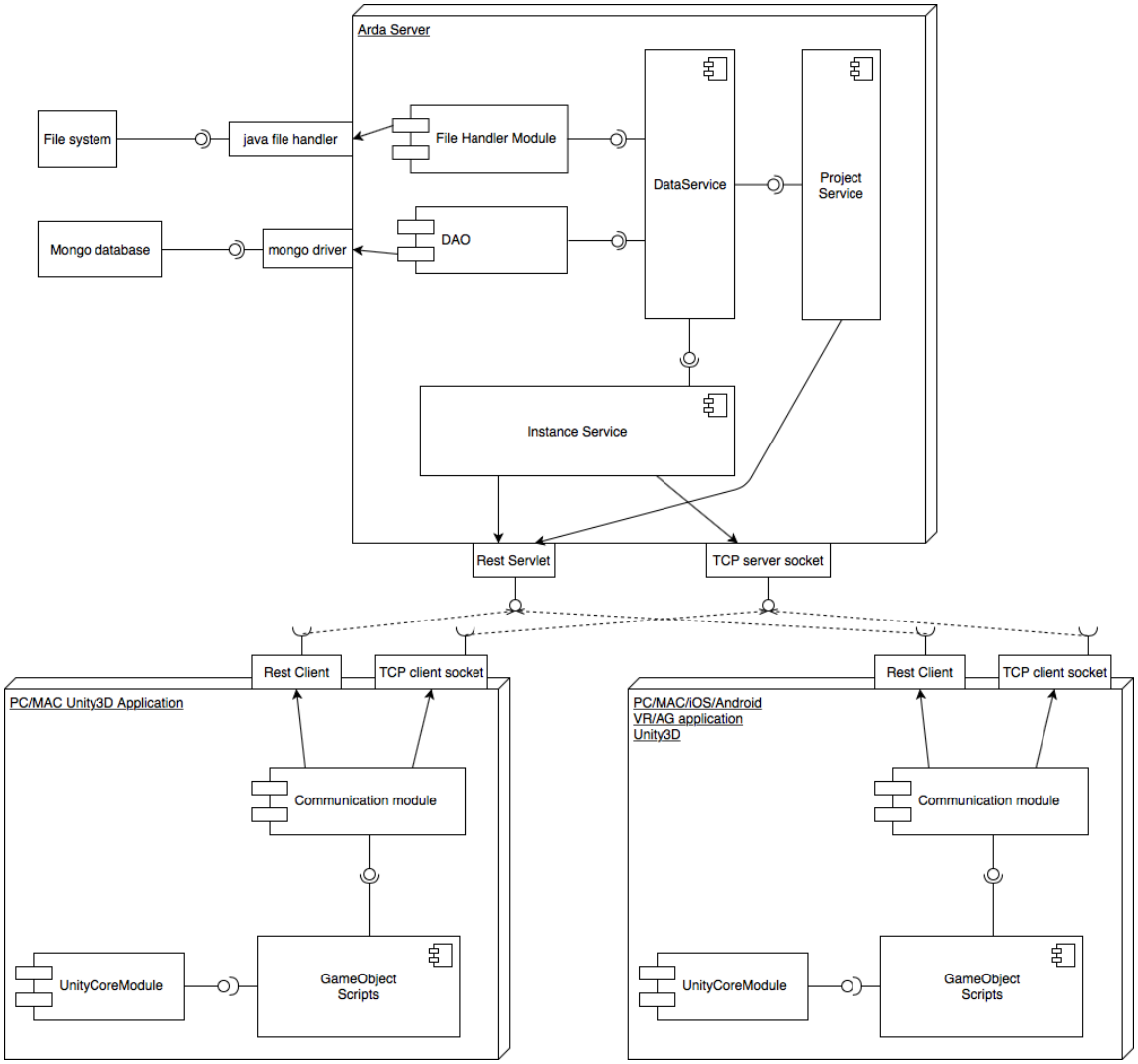
- airSpeed - pravá vzdušná rychlost letadla v metrech za sekundu, float
- heading - kurz letu ve stupních, float
- node - pozice letadla, Node
- airplaneType - obsahuje technické parametry letadla, AirplaneType
- scenario - scénář akcí, Scenario

3.3 Architektura komponent systému

Na základě nefunkčních a funkčních požadavků je vytvořen diagram komponent a navržena architektura systému.

Na obrázku 3.7 je návrh modulů a komponent systému.

Systém bude rozdělen na serverovou a klientskou část. Komunikace s nekritickými požadavky na odezvu bude probíhat přes rozhraní REST. Komunikace s kritickými požadavky na odezvu (například změna pozice letadla) bude probíhat přes protokol TCP.



Obrázek 3.7. Architektura komponent a modulů.

Zprávy mezi serverem a klienty budou ve formátu JSON.

Databáze pro ukládání dat bude MongoDB, což je objektová databáze.

3.3.1 Arda Server

Aplikace Arda Server bude vytvořena na frameworku Spring, který je nadstavbou JavaEE. K jejímu spuštění tedy bude vyžadována přítomnost Java virtual machine. Důvod pro volbu Javy je převážně snadné docílení multiplatformosti.

Arda server bude klientům poskytovat rozhraní pro získání:

- Stavu aktuálně běžící instance.
- Seznamu uložených projektů.
- Seznamu uložených instancí k vybranému projektu.
- Data vybraného projektu.
- Data vybrané instance.

Také bude klientům poskytovat rozhraní pro odesílání příkazů pro:

- Vytvoření nového projektu.
- Smazaní projektu.
- Vytvoření nové instance projektu.
- Smazání instance projektu.
- Tvorbu a změny vzdušných prostorů.
- Tvorba a změny sektorů.
- Změnu scénáře letadla.

Výpočty pro pohyb letadel budou prováděny na serverové straně. Vstupy, které mohou ovlivnit trajektorii letu, budou přicházet pouze z letových scénářů. Pokud tedy například uživatel klientské části odešle na server příkaz ke změně kurzu vybraného letadla, je tento příkaz přidán do scénáře letadla a server začne dle získaného příkazu do scénáře měnit kurz letadla podle pravidel, která jsou definovaná na serverové straně.

3.4 Maps SDK for Unity

I přes to, že bude v systému zeměkoule projektována na 2D plochu, je potřeba dělat výpočty pro pohyb letadel na 3D modelu zeměkoule.

Jako vstup jsou dostupná data o aktuální poloze letadla v geografických souřadnicích (zeměpisná šířka a délka), výška letadla nad zemí, rychlost letadla a směr letu.

Po systému bude vyžadováno, aby pro letadlo na pozici A, která je definovaná pomocí zeměpisné šířky a délky, našel nejlepší možný kurz pro navedení letadla tak, aby se v co nejkratším časovém intervalu dostalo na pozici B.

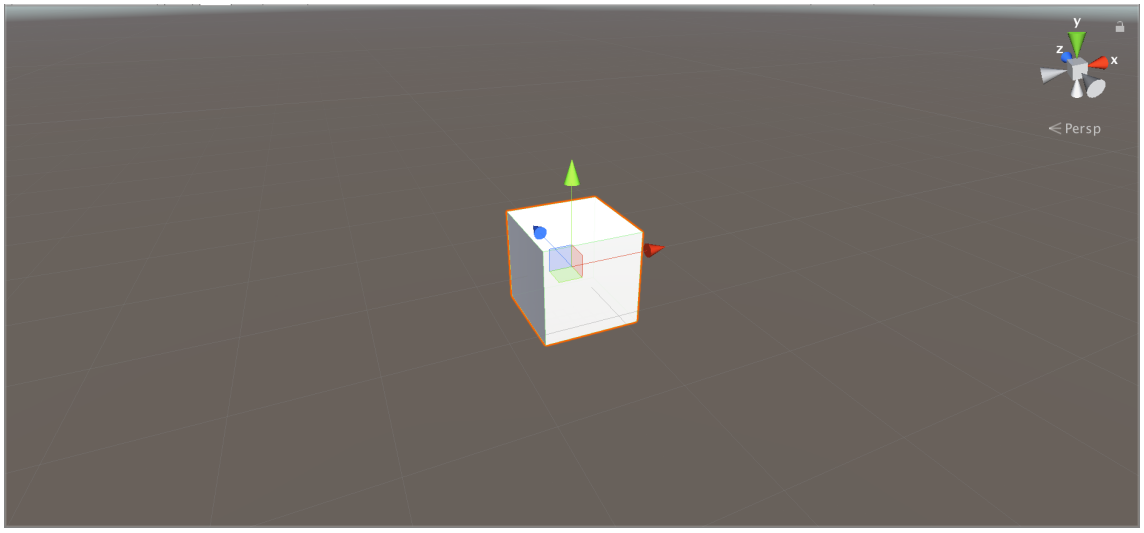
Pro mapové zobrazení bude využit Maps SDK for Unity od společnosti Mapbox. Tento framework umožňuje mnoho užitečných věcí. Například je schopen na vygenerovanou mapu úmistit objekt podle definované zeměpisné šířky a délky. Je potřeba brát na vědomí, že vygenerovaná Mapbox mapa v Unity je zasazena do 3D scény a její střed je ve scéně umístěn na součadnicích x, y a z.

Na obrázku 3.8 je znázorněno umístění krychle o velikosti jednoho čtverečního metru na souřadnice [0, 0, 0].

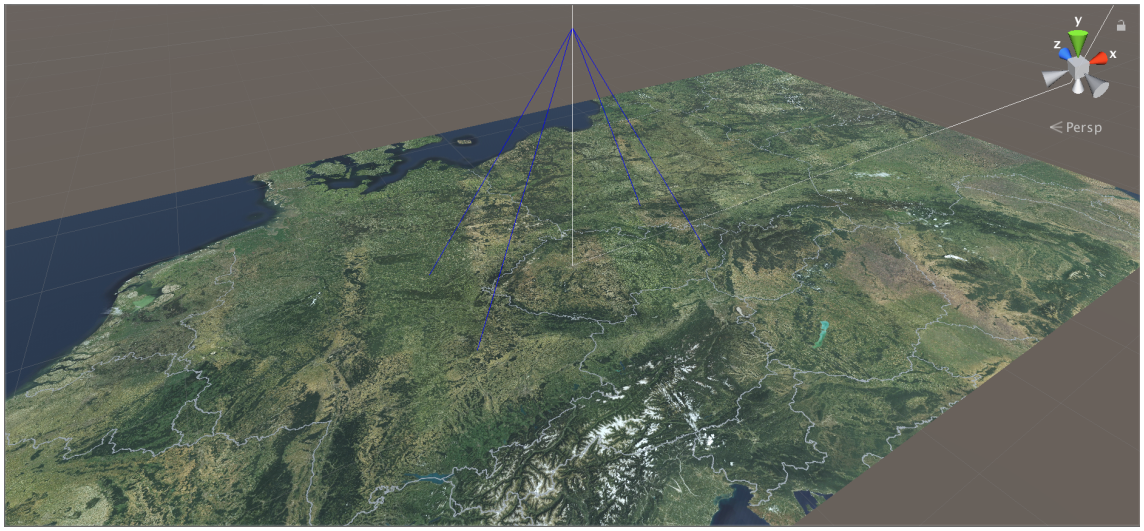
Na obrázku 3.8 je možné si povšimnout čtvercované sítě, která značí rovinu x,z. Tato rovina se bude považovat za vodorovnou a mapa bude umístěna rovnoběžně s touto rovinou a posunutá na ose y podle toho, v jaké výšce nad zemí se bude nacházet střed mapy.

Příklad takového umístění mapy je znázorněn na obrázku 3.9.

Mapbox umožňuje mnoho možností zobrazení mapy. Pro účely navrhovaného software bude vytvořen vlastní styl mapy, který bude využívat satelitního zobrazení, bude vykreslovat hranice států a také bude využívat výškové mapy pro deformaci roviny podle reálného terénu.



Obrázek 3.8. Souřadnicový systém Unity3D.



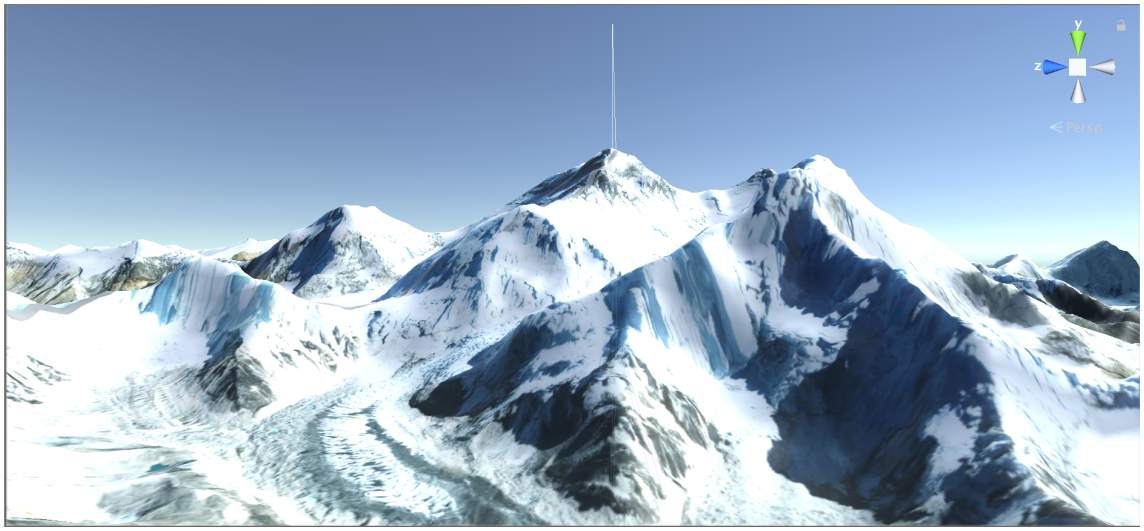
Obrázek 3.9. Umístění mapy v Unity3D.

Na obrázku 3.9 je plocha deformována dle reálného terénu, ale není to příliš patrné. Naproti tomu na obrázku 3.10 je znázorněn Mount Everest, který je také vykreslen pomocí Unity a Mapbox. Zde už je terén velmi patrný.

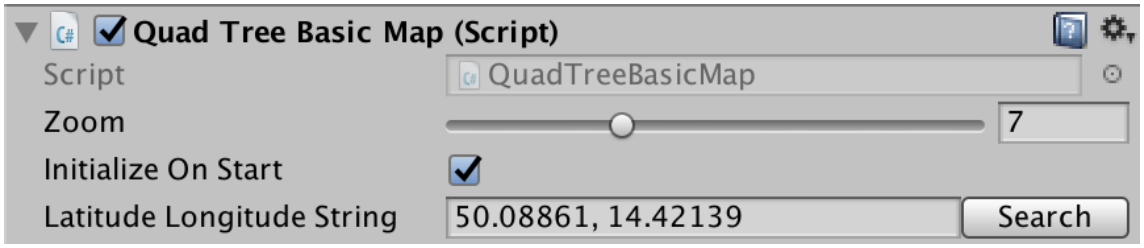
Vykreslování mapy probíhá pouze v části mapy zeměkoule, která je definována pomocí parametrů. Specificky pomocí středového bodu mapy v zeměpisných souřadnicích a hodnoty přiblížení mapy, která se v Map SDK označuje jako zoom a může nabývat hodnot 0-22. Příklad takového nastavení v editoru Unity je znázorněn na obrázku 3.11.

3.5 Navigace letadel v systému

Země není dokonalá koule, a na to je potřeba brát ohledy při výpočtech. Navrhovaný software bude používat WGS84 (World Geodetic System 1984) model. Tento model je světově uznávaný standard pro definování souřadnicového systému a referenčního



Obrázek 3.10. Terén na mapě v Unity3D.



Obrázek 3.11. Nastavení pozice mapy.

elipsoidu pro navigaci.

Souřadnice na modelu WGS84 vychází ze zeměpisných souřadnic. Poloha na modelu se tedy určí pomocí zeměpisné šířky a délky.

Zeměpisná šířka nabývá hodnot 0 až 90 stupňů na sever i na jih od rovníku. Zeměpisná délka pak nabývá hodnot 0 až 180 stupňů na západ i na východ od nultého poledníku.

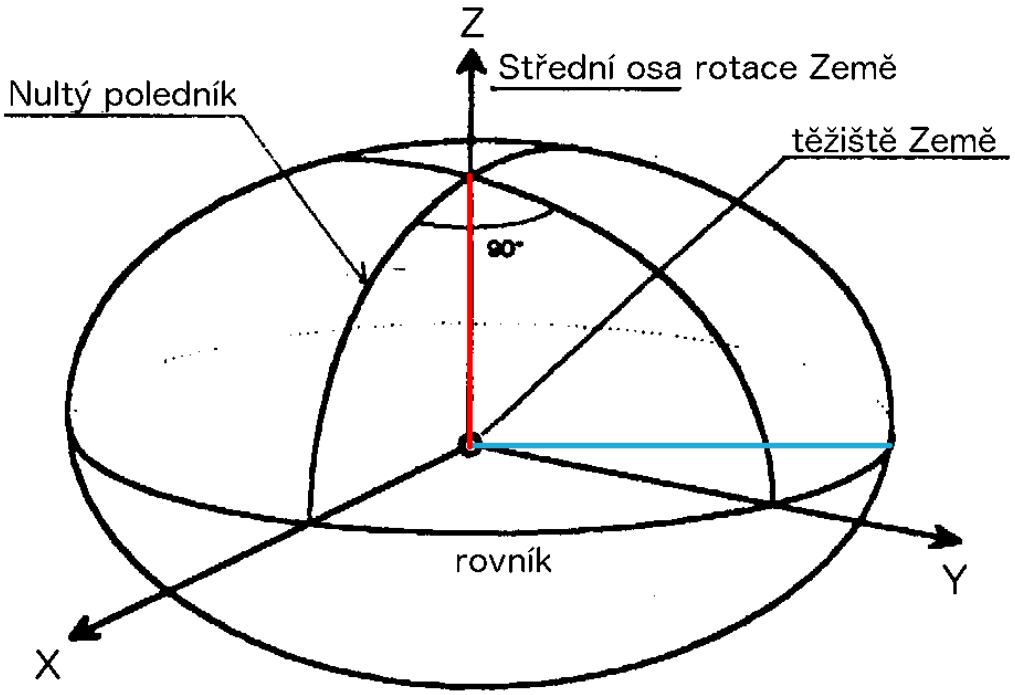
Souřadnicový systém WGS84 je pravotočivá kartézská soustava souřadnic se středem v těžišti Země. Kladná osa x směřuje k průsečíku nultého poledníku a rovníku. Kladná osa z k severnímu pólu a kladná osa y je na obě předchozí osy kolmá ve směru 90 stupňů východní délky a 0 stupňů šířky. Jedná se tedy o pravotočivou soustavu souřadnic.

Parametry pro definování referenčního elipsoidu jsou:

- délka hlavní polosy a = 6 378 137m
- délka vedlejší poloosy b = 6 356 752,3142m

Na obrázku 3.12 je schéma systému WGS84. Modře je znázorněna hlavní poloosa, červeně vedlejší poloosa.

Pro pohyb letadla v modelu WGS84 je tedy nutné postavit algoritmus, který bude v krátkých časových intervalech měnit polohu letadla v zeměpisných souřadnicích



Obrázek 3.12. WGS84.

vzhledem ke kurzu, rychlosti, výšce a aktuální poloze letadla.

Výsledkem tohoto algoritmu bude vždy zeměpisná šířka a délka. Ta se následně využije ke správnému umístění letadla na mapu pomocí funkce definované v API Mapbox SDK s názvem GeoToWorldPosition, která si jako parametr bere Vector2, což je třída definovaná v Unity3D a slouží k uložení dvou parametrů x a y do jedné instance třídy.

Také bude potřeba objektem letadla v souřadnicích scény správně rotovat, aby model letadla směřoval ke kurzu letu.

3.5.1 Algoritumus pro pohyb letadla

Je potřeba řešit změnu polohy vzhledem k letěné vzdálenosti za čas, který uběhl mezi poslední iterací a novou iterací algoritmu.

Jako jeden z parametrů algoritmu tedy bude rozdíl času mezi dvěma iteracemi v milisekundách. Ten bude značen jako Δt.

Dále je potřeba znát rychlost letadla, které se pohybuje v součadnicovém systému. Ta bude značena jako v.

Výpočet dráhy s v daném časovém intervalu bude tedy dán rovnicí:

$$s = \Delta t * v \tag{1}$$

Jelikož se letadlo nepohybuje po přímce, ale po křivce, a také proto, že jako výsledek algoritmu jsou vyžadovány výsledné úhly (zeměpisná šířka a délka), je potřeba vypočítat změnu úhlu v daném časovém intervalu a poměr, ve kterém je nutné změnu úhlu aplikovat mezi zeměpisnou šířku a délku.

Pro výpočet změny úhlu je zapotřebí znát poloměr kružnice, která má střed ve stejném bodě jako střed systému WGS84, a výšku letadla nad zemí.

V každé iteraci tedy bude vypočítávána hodnota r dle rovnice:

$$r = r_e + h_a \tag{2}$$

kde r_e je vzdálenost od středu k povrchu Země v modelu WGS84 a h_a je výška letadla nad zemí.
Vzdálenost mezi středem Země a povrchem Země v dané zeměpisné šířce a délce bude prováděno výpočtem:

$$r_e = \sqrt{\frac{(a^2 * cosx)^2 + (b^2 * sinx)^2}{(a * cosx)^2 + (b * sinx)^2}} \tag{3}$$

Kde a je délka hlavní poloosy, b je délka vedlejší poloosy a x je zeměpisná šířka.

Změna úhlu $\Delta\alpha$ lze díky rovnosti poměrů zapsat rovnicí:

$$\Delta\alpha = \frac{s}{r} \tag{4}$$

Výsledná zeměpisná šírka se následně vypočítá, díky poměrnému rozdělení celkové změny úhlu vzhledem ke směru letu rovnicí:

$$x = x_0 + sin(h) * \Delta\alpha \tag{5}$$

Kde h je směr letu v radiánech a x_0 je počáteční zeměpisná šířka.

Pro výpočet výsledné zeměpisné délky se použije rovnice:

$$y = y_0 + cos(h) * \Delta\alpha \tag{6}$$

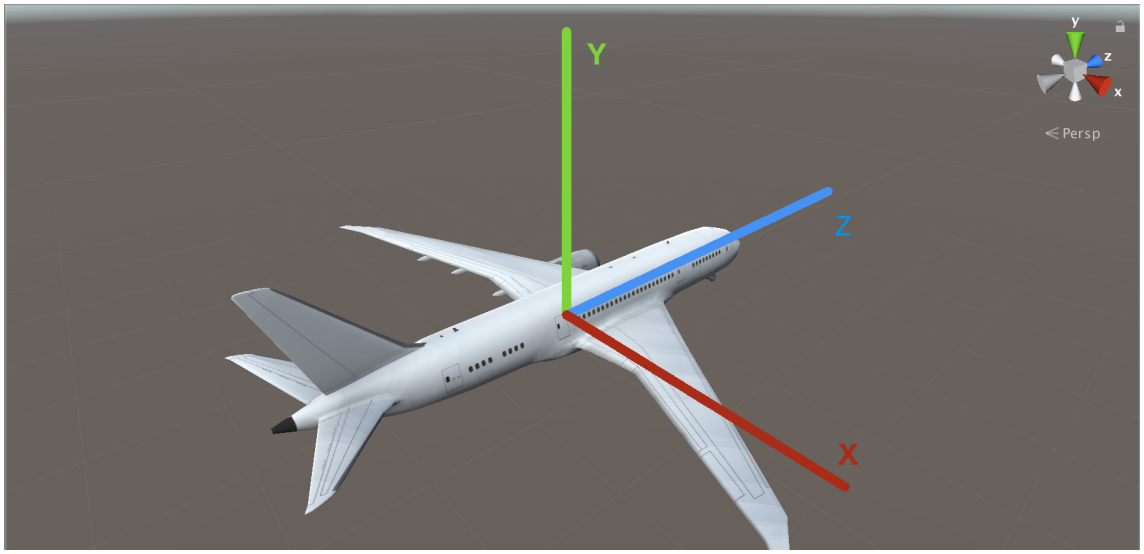
Kde y je výsledná zeměpisná délka a y_0 je počáteční zeměpisná délka.

3.5.2 Algoritmus pro rotaci letadla

V systému bude sever vždy umístěn na souřadnicích $[x, y, z] = [0, 0, \infty]$

Letadla jako 3D objekty v Unity budou v souřadnicovém systému umístěna tak, aby jejich kurz směřoval přímo v ose z. To je znázorněno na obrázku 3.13
K rotaci 3D objektu se bude využívat kvaternionů. K rotacím lze také využívat Eulerovy úhly. Ty mají ovšem několik nevýhod:

- Jsou závislé na souřadném systému.
- Souřadný systém je lokální vůči tělesu.
- Komplexní rotace jsou závislé na pořadí jednotlivých rotací.



Obrázek 3.13. Letadlo v Unity.

- Mají složitý inverzní problém.
- Existence gimbal locku, kde i přes změny hodnot v jednom z úhlu se nemění rotace celého tělesa.

Kvaterniony byly objeveny 16. října 1843 Williamem Rowanem Hamiltonem. Když ho napadla myšlenka kvaternionů, byl právě na vycházce a přecházel most. Protože byl zvyklý si každou myšlenku zapisovat, a neměl při sobě papír a tužku, vyškrábal tedy svou myšlenku do kameného mostu po kterém zrovna přecházel. Vyškrábal tam rovnici: $i^2 = j^2 = k^2 = -1$.

Matematika kvaternionů není triviální, a proto se s ní v této práci nebudu zabývat do hloubky. Unity3D navíc poskytuje veškerá rozhraní pro práci s kvaterniony.

V Unity3D je rotace objektu popsána v instanci třídy Vector3, která v sobě obsahuje tři parametry x, y a z. Každý objekt ve scéně má na sobě takový vektor, který popisuje jeho rotaci.

Pro správnou rotaci letadla vzhledem k jeho kurzu tedy bude využita standartní funkce Unity3D Quaternion.Euler, která vyžaduje tři parametry. Otáčet 3D objektem letadla vzhledem k jeho kurzu je potřeba pouze v ose y (čili osa Yaw). Funkce tedy bude volána v každem Update cyklu na GameObjektu letadla a takto: Quaternion.Euler(airplane.transform.rotation.x, airplane.flyDirection, airplane.transform.rotation.z), kde airplane je reference na GameObjekt letadla, transform je reference na objekt, který Unity3D využívá pro popis umístění předmětu ve scéně, rotation je reference na instanci Vector3, která má v sobě údaje o aktuální rotaci GameObjektu a flyDirection je float hodnota, ve které je uložen aktuální kurz letadla.

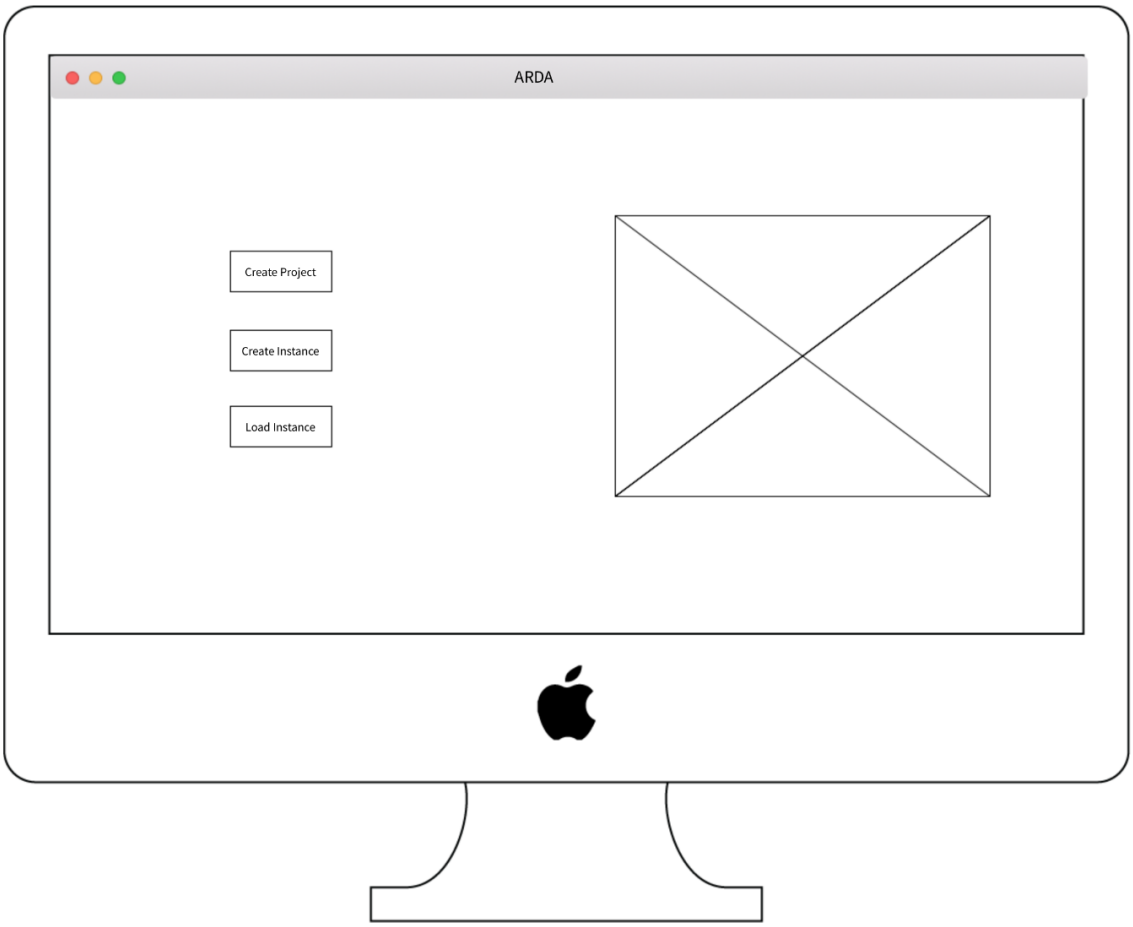
Poznámka: GameObjekt je v Unity3D každá entita, která je přidána do scény. Každý takový objekt obsahuje funkci Start() a Update(). Funkce Start() se volá při vytvoření objektu. Funkce Update() se volá v každém cyklu aktualizace stavu engine, což může být řádově několik milisekund po sobě.

3.6 Klientská aplikace

Aplikace bude zabalena tak, aby se všechny její komponenty nastartovaly automaticky po spuštění jednoho souboru. Serverová část i klientská část v Unity3D startují souběžně. Pokud klientská část nastartuje dříve než serverová, pak se objeví dialog s vyčkáváním na připojení. Po připojení k serverové části se objeví hlavní menu.

3.6.1 Hlavní menu

Návrh hlavního menu je vyobrazen na obrázku 3.14.



Obrázek 3.14. Mockový návrh hlavního menu.

Hlavní menu bude obsahovat tři tlačítka s možností vytvoření nového projektu, vytvoření nové instance již existujícího projektu a načtení instance projektu.

Po výběru vytvoření nového projektu bude uživatel vyzván k zadání základních údajů o projektu, jako je název projektu, způsob jeho uložení (databáze, soubor), výchozí střed mapy a popřípadě další parametry, které se v průběhu implementace ukážou být nutné k inicializaci projektu. Po potvrzení je zkontrolována unikátnost názvu projektu a pokud nedojde k chybě, ojeví se hlavní obrazovka aplikace.

Po výběru vytvoření instance již existujícího projektu bude uživatel vyzván k výběru existujícího projektu ze seznamu. Následně zadá nové jméno instance. Po potvrzení

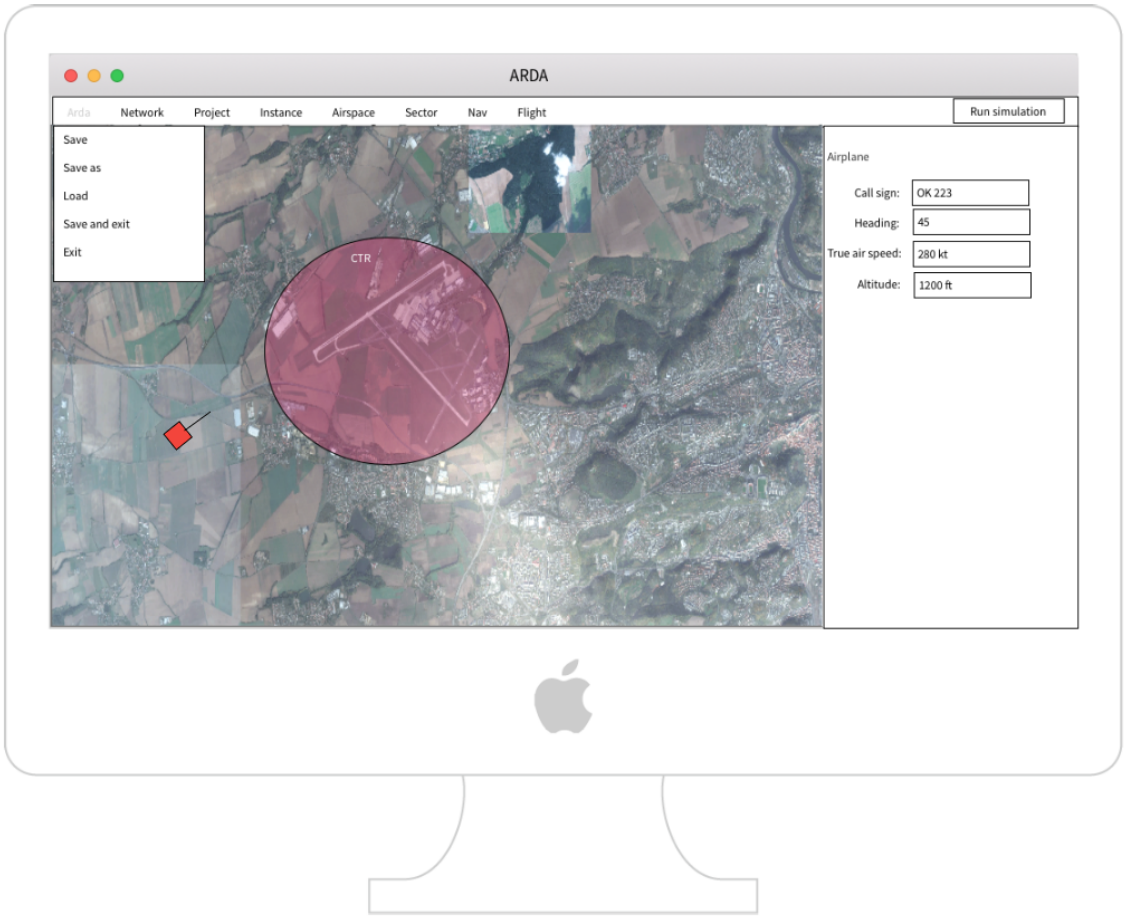
systém zkontroluje unikátnost názvu instance, a pokud nedojde k chybě, objeví se hlavní obrazovka s načteným projektem.

Po výběru načtení instance projektu je uživatel vyzván k výběru existujícího projektu a následně k výběru existující instance. Po potvrzení se objeví hlavní obrazovka s načteným projektem a jeho instancí.

Poznámka: Projekt v sobě uchovává informace o rozmístění vzdušných prostorů, sektorů, traťových bodů, letištních drah, výchozích poloh letadel a výchozích scénářích letového provozu. Instance je vázána na projekt a uchovává v sobě informace o aktuálních polohách letadel, aktuálního stavu scénáře letového provozu, aktuálního času od času t0 a aktuálního seskupení sektorů do skupin.

3.6.2 Main obrazovka

Návrh rozložení hlavní obrazovky je vyobrazen na obrázku 3.15



Obrázek 3.15. Mockový návrh hlavní obrazovky.

V horní části obrazovky se nachází obslužné menu se záložkami Arda, Network, Instance, Airspace, Sector, Nav a Flight. Těmto záložkám bude věnováno více dále v

textu.

Vpravo nahoře se bude nacházet tlačitko, které bude přepínat mezi režimy simulace a editace.

V pravé části obrazovky se bude nacházet dynamický panel, který bude zobrazovat informace o právě vybrané entitě (letadlu, přistávací dráhy atd.). Tento panel se bude nazývat editační.

Záložka Arda

Záložka Arda rozbaluje menu s tlačítky:

- Save - ukládá aktuální projekt a instanci
- Save as - ukládá aktuální projekt a instanci jako nový projekt a instanci
- Load - načítá projekt a instanci
- Save and exit - ukládá projekt a instanci, následně přesměruje do hlavního menu
- Exit - přesměruje do hlavního menu bez uložení

Záložka Network

Záložka Network rozbaluje menu s tlačítky:

- Start server - spustí režim serveru, ve kterém je možné se připojit k instanci i z jiných strojů pomocí sítě
- Stop server - vypne režim serveru
- List of clients - zobrazí seznam připojených klientů v editačním panelu. Pro každého klienta bude možné v editačnim panelu měnit jeho roli a přiřazovat skupinu sektorů

Záložka Project

Záložka Project rozbaluje menu s tlačítky:

- New - vytvoří nový projekt
- Load - načte projekt a vytvoří prázdnou instanci
- Save - uloží projekt
- Save as - uloží projekt jako nový
- Info - zobrazí informace o projektu v editačnim panelu

Záložka Instance

Záložka Instance rozbaluje menu s tlačítky:

- New - vytvoří novou instanci projektu
- Load - načte instanci projektu
- Save - uloží instanci projektu
- Save as - uloží instanci projektu jako novou
- Info - zobrazí informace o instanci v editačním panelu

Záložka Airspace

Záložka Airspace rozbaluje menu s tlačítky:

- Add - Aktivuje v editačním panelu nástroj pro tvorbu vzdušného prostoru.
- Add part - Aktivuje v editačním panelu nástroj pro tvorbu části vzdušného prostoru (včetně jeho tvaru). Aktivní pouze v případě, že je právě vybrán nějaký vzdušný prostor.
- List - Aktivuje v editačním panelu seznam všech vzdušných prostorů.

Záložka Sector

Záložka Sector rozbaluje menu s tlačítky:

- Add - aktivuje v editačním panelu nástroj pro tvorbu sektoru
- Add part - aktivuje v editačním panelu nástor pro tvorbu části sektoru, včetně jeho tvaru (aktivní bude pouze pokud je právě vybrán nějaký sektor)
- List - aktivuje v editačním panelu seznam všech sektorů

Záložka Nav

Záložka Nav rozbaluje menu s tlačítky:

- Add - aktivuje v editačním panelu nástroj pro tvorbu traťového bodu
- List - aktivuje v editačním panelu seznam všech traťových bodů

Záložka Add

Záložka Add rozbaluje menu s tlačítky:

- Add - aktivuje v editačním panelu nástroj pro tvorbu nového letadla
- List - aktivuje v editačním panelu seznam všech letadel

3.6.3 Scénáře letového provozu

Budou existovat dva typy scénářů letového provozu - letištní dráhy a letadlový. Tyto scénáře budou sloužit jako správci akcí, které budou vykonávány dráhami a letadly.

Scénáře letišťních drah budou obsahovat pouze seznam akcí, které budou aktivovány po dosažení definovaného času. Půjde jen o vzlety letadel. Dále se tedy budu zabývat pouze druhou skupinou scénářů.

GameObjekty, které budou využívat scénářů budou implementovat interface ScenarioManager.

Interface ScrenarioManager bude tvořen parametry actionList, currentAction a memoryAction. Dále bude vyžadovat implementaci funkcí pause, resume, removeCurrentAction, removeAllActions a perfromNow.

Vysvětlení významu jednotlivých parametrů:

- ActionList bude tvořen seznamem referencí na akce pro daný game objekt.
- CurrentAction bude reference na aktuálně prováděnou akci.
- MemoryAction bude reference, do které se bude ukládat odkaz na probíhající akci při zavolání funkce performNow.

Vysvětlení významu jednotilivých funkcí:

- Pause - zastaví vykonávání aktuální akce.
- Resume - obnoví vykonávání aktuální akce.
- RemoveCurrentAction - zastaví a odstraní aktuální akci ze scénáře. Pokud je uložena reference na memoryAction, obnoví její vykonávání.
- RemoveAllActions - zastaví a odstraní všechny akce ve scénáři.
- PerformNow - (vyžaduje v argumentu novou akce) zastaví vykonávání aktuální akce, uloží referenci na aktuální akci do memoryAction a spustí vykonávání nové akce.

Parametry letadel, jako jsou kurz, výška, rychlost, atd. bude možné měnit pouze za pomocí scénářů letového provozu a to díky definovaným akcím. Akcí bude několik typů a každy typ akce bude mít svou třídu. Všechny třídy akcí budou dědit z abstraktní třídy Action.

Abstraktní třída Action

Definované funkce abstraktní třídy:

- Start - začne vykonávat akci
- Stop - přestane vykonávat akci
- Destroy - zastaví a následně zničí akci
- StartInTime - nastaví čas pro start vykonávání akce

Třída Action také bude obsahovat několik funkcí, které se budou volat při nastání určité události (callback funkce). Pro každou akci bude možné do těchto funkcí vložit referenci na funkci jiného objektu, která bude následně vykonána. Tím bude možné zajistit řetězení vykonávaných akcí.

Callback funkce budou:

- AfterCompletation - bude zavolána po dokončení akce
- UnAble - bude zavolána, pokud z nějakého důvodu nemohla být akce dokončena
- AfterStart - bude zavolána po zahájení akce

Jak již bylo řečeno, každý typ akce bude mít svou třídu, ve které bude definováno co se má s jakou entitou v průběhu akce dělat. Zároveň je také potřeba do instancí akcí vložit reference na potřebné entity, se kterými budou tyto akce pracovat. Toho bude docíleno tak, že entity, které bude možné využívat v akcích, budou implementovat interface Actor. Tento interface zaručí, že budou na objektech dostupné funkce:

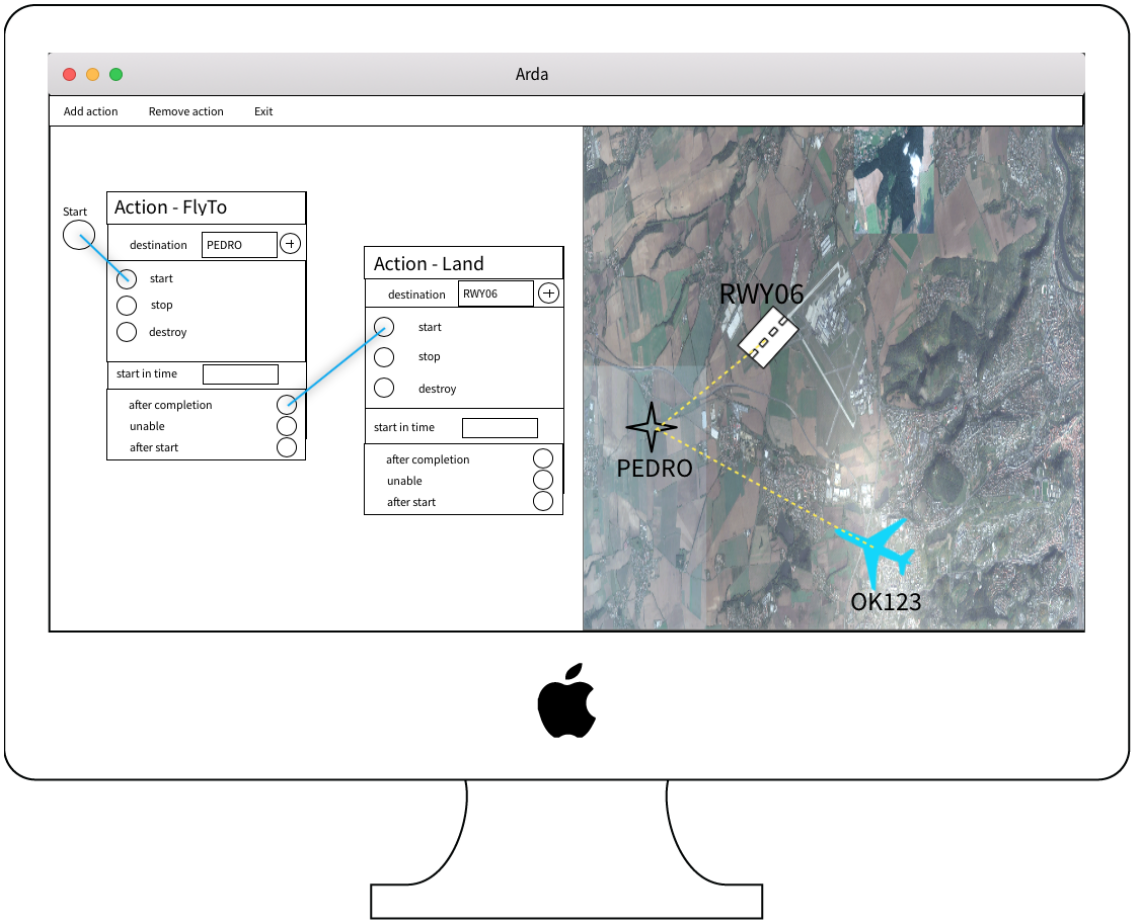
- GetPositionNode - vrací instanci třídy Node, která obsahuje data o pozici entity ve scéně.
- GetGeoPosition - vrací instanci třídy Vector2, která obsahuje zeměpisnou šířku a délku entity.

- GetAltitude - vrací float s hodnotou výšky nad střední hladinou moře.

Akce, které bude možné přidat do scénáře letadla jsou:

- FlyTo - letět na přistávací dráhu, traťový bod nebo bod na mapě.
- Land - přistát na dráhu.
- TakeOff - vzletět z dráhy.
- Descend - klesat o určitý počet stop.
- DescendTo - klesat na letovou hladinu nebo nadmořskou výšku.
- Climb - stoupat o určitý počet stop.
- ClimbTo - stoupat na letovou hladinu nebo nadmořskou výšku.
- TurnLeft - změnit kurz směrem doleva na určené stupně.
- TurnRight - změnit kurz směrem doprava na určené stupně.

Pro snadné vytváření akcí a scénářů bude v systému dostupný grafický editor. Podoba takového editoru je na obrázku 3.16.



Obrázek 3.16. Mockový návrh scénáře letového provozu.

Kapitola 4
Implementace

Cílem této práce je převážně návrh softwarového řešení, ale součástí je ověření funkčnosti návrhu ve zjednodušené formě implementace typu proof of concept. V této implementaci zanedbávám návrh systému ve formě klient-server a zaměřuji se převážně na funkčnost navržených algoritmů, zobrazování dat ve 3D enginu a ověření konceptu s použitím VR a AG technologií. Zdrojové kódy jsou psané v jazyce C#, který je jedním ze dvou podporovaných jazyků v Unity3D.

4.1 Navigace letadel v systému

Pro výpočet změny zeměpisné šířky a délky při simulaci pohybu letadla v prostoru definovaným modelem WGS84 je využíván algoritmus:

```
float airSpeedInMetersInSec = this.kilometersInHToMetersInS (airSpeed);
float s = airSpeedInMetersInSec * Time.deltaTime;
float r = GEOUtils.earthRadiusByLatitude (latitude) + altitude;
float alpha = 0.0f;
if (r != 0) {
alpha = s / r;
}
float alphaInDeg = alpha*Mathf.Rad2Deg;
longitude = longitude + Mathf.Sin(flyDirection
* Mathf.Deg2Rad) * alphaInDeg;
latitude = latitude + Mathf.Cos (flyDirection
* Mathf.Deg2Rad) * alphaInDeg;
```

Algoritumus pro výpočet vzdálenosti mezi středem a povrchem Země:

```
float latitudeInRadian = latitude * (Mathf.PI / 180);
float f1 = Mathf.Pow ((Mathf.Pow(axisA, 2) *
Mathf.Cos(latitudeInRadian)), 2);
float f2 = Mathf.Pow ((Mathf.Pow(axisB, 2) *
Mathf.Sin(latitudeInRadian)), 2);
float f3 = Mathf.Pow ((axisA * Mathf.Cos(latitudeInRadian)), 2);
float f4 = Mathf.Pow ((axisB * Mathf.Sin(latitudeInRadian)), 2);
float radius = Mathf.Sqrt((f1 + f2) / (f3 + f4));
return radius;
```

Algoritmus pro umístění a rotaci letadla ve scéně:

```
// Update is called once per frame
void Update () {
UpdateAirplaneTransformPosition ();
UpdateHeadingRotation ();
}

void UpdateAirplaneTransformPosition() {
```

```
Vector2d latLongVector = new Vector2d
(airplane.latitude, airplane.longitude);
Vector3 worldPosition = mapbox.GeoToWorldPosition (latLongVector);

float y = airplane.altitude - (float) mapboxWorldPosition.
systemLevelAboveSea;
worldPosition.y = y;

this.transform.position = worldPosition;
}

void UpdateHeadingRotation() {
airplane.transform.rotation = Quaternion.Euler
(airplane.transform.rotation.x,
airplane.flyDirection, airplane.transform.rotation.z);
}
```

4.1.1 Renderování sektorů ve 3D zobrazení

V implementaci je hotové řešení pro tvorbu a vykreslení sektorů ve tvaru mnohoúhelníku. Uživatel je při tvorbě takového sektoru vyzván, aby zvolil na mapě od tří do libovolného počtu bodů, které vytyčí tvar sektoru. Při vytvoření nového bodu se dva předchozí body spojí úsečkou. Po opětovném vybrání prvního bodu v řetězci dojde k uzavření řetězu a ukončení tvorby tvaru sektoru. Následně se nastaví spodní a horní výška nad zemí a sektor je vykreslen.

K samotnému vykreslení se využívá algoritmu pro triangulaci, který obsah mnohoúhelníku rozdělí na konečný počet trojúhelníků. Body těchto trojúhelníku jsou potom využity k vytvoření Mesh spodní základny sektoru. Stejný mesh jako základna je použit i pro vrchní základnu sektoru. Dalším algoritmem je potom docíleno vytvoření bodů obdélníků, které tvoří strany mnohoúhelníku. Tyto body jsou také využity k vytvoření několika objektů Mesh (tolika, kolik je stran mnohoúhelníku). Všechny vytvořené objekty Mesh jsou následně sloučeny v jeden, který je předán k renderování.

Pro sektor ve 3D prostoru je také vytvořen průhledný materiál pomocí shaderu a na jednotlivé hrany je aplikována čára, aby bylo možné rozlišovat tvary sektoru.

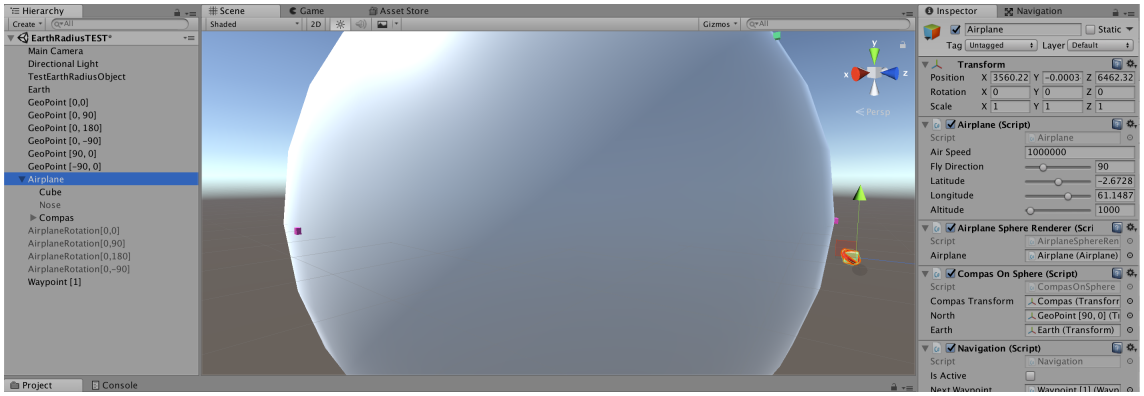
Na obrázku 4.1 je zobrazena testovací scéna pro ověření funkčnosti algortimů pro pohyb letadla po křivce kolem Země vzhledem k jeho rychlost, výšsce nad zemí a kurzem. V pravé části lze vidět parametry, které se dají v během simluace dynamicky měnit a letadlo podle nich upravuje směr a rychlost letu.

Ukázka vyrenderovaného sektoru ve tvaru mnohoúhleníku je na obrázku 4.2.

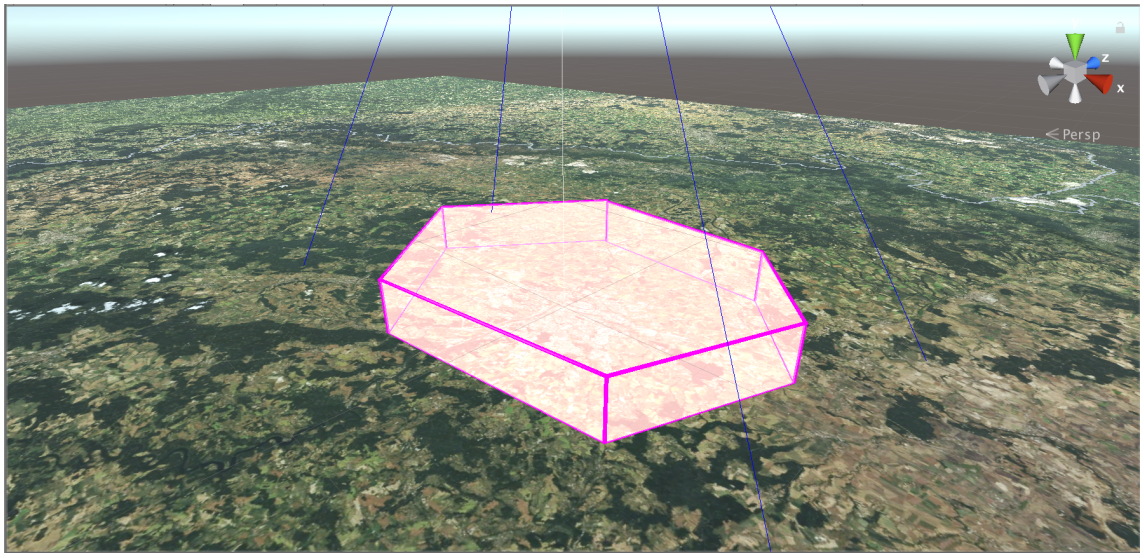
Funkce Update, která je v game objektu sektoru a stará se výpočet a renderování mnohoúhelníkového sektoru, je na obrázku 4.3

4.2 Augmentovaná realita

K implementaci augmentované reality je využit Unity AR Kit Plugin a Unity AR Interface. S těmito pluginy bylo možné postavit 3D scénu tak, aby v ní byla vyrenderována mapa (Mapbox) a ve které je možné zároveň umístit letadla, traťové body, přistávací



Obrázek 4.1. Testovací scéna pro ověření funkčnosti pohybových algoritmů.



Obrázek 4.2. Vyrenderovaný sektor ve tvaru mnohoúhelníku.

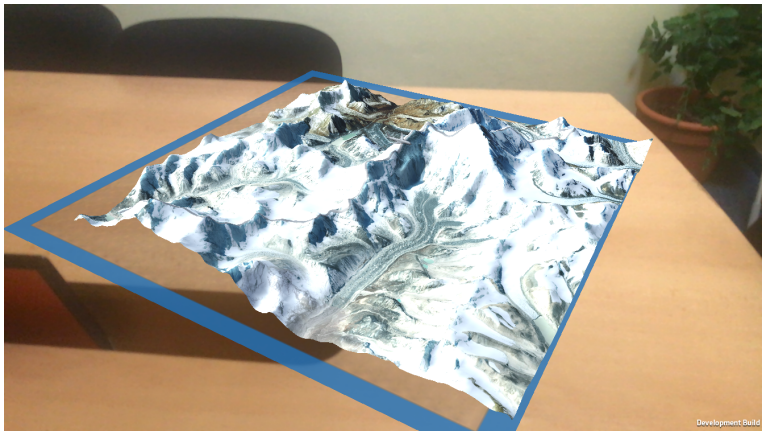
dráhy atd.

Detekce ploch pro umístění scény je řešena pomocí algoritmů v pluginu AR Kit. Po vybrání plochy se hledání těchto ploch deaktivuje a na plochu se zobrazí mapa a všechny ostatní entity.

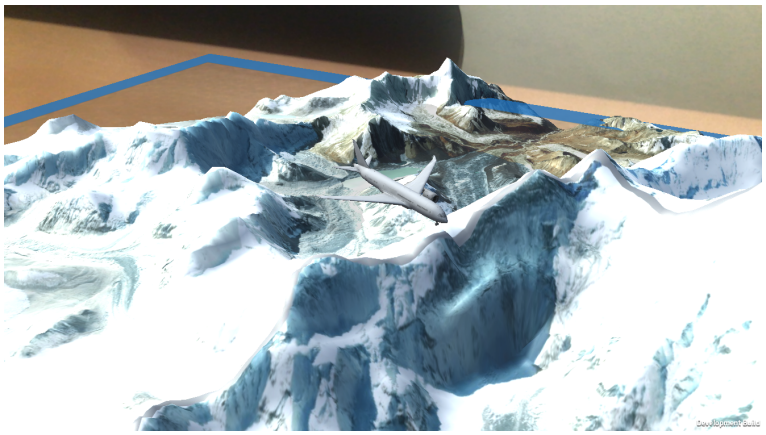
Obrázky 4.4, 4.5 a 4.6 ukazují zapauzovanou scénu s využitím augmentované reality z několika úhlů.

```
void Update () {
        if (vertices2D.Length != editPoints.Count) {
            vertices2D = new Vector2[editPoints.Count];
        }
        for (int i = 0; i < editPoints.Count; i++) {
            Vector2 vertex = new Vector2
(editPoints[i].transform.position.x, editPoints[i].transform.position.z);
            vertices2D [i] = vertex;
        }

        Triangulator tr = new Triangulator(vertices2D);
        int[] indices = tr.Triangulate();

        // Create lower Vector3 vertices
        Vector3[] lowerVertices = new Vector3[vertices2D.Length];
        for (int i=0; i<lowerVertices.Length; i++) {
            lowerVertices[i] = new Vector3(vertices2D[i].x,
vertices2D[i].y, 0);
        }

        // Create lower mesh
        Mesh lowerMesh = new Mesh();
        lowerMesh.vertices = lowerVertices;
        lowerMesh.triangles = indices;
        lowerMesh.RecalculateNormals();
        lowerMesh.RecalculateBounds();

        // Create upper Vector3 vertices
        Vector3[] upperVertices = this.getUpperVertices(lowerVertices);

        Mesh upperMesh = new Mesh ();
        upperMesh.vertices = upperVertices;
        upperMesh.triangles = indices;
        upperMesh.RecalculateNormals();
        upperMesh.RecalculateBounds();

        // Create side vertices
        ArrayList sideVerticesAndIndices =
this.getSideVerticesAndIndicesHard(upperVertices, lowerVertices);
        Vector3[] sideVertices = (Vector3[]) sideVerticesAndIndices [0];
        int[] sideIndices = (int[]) sideVerticesAndIndices [1];

        Mesh sideMesh = new Mesh ();
        sideMesh.vertices = sideVertices;
        sideMesh.triangles = sideIndices;
        sideMesh.RecalculateNormals ();
        sideMesh.RecalculateBounds ();

        // Combine into one mesh

        MeshFilter filter = gameObject.GetComponent<MeshFilter> ();

        CombineInstance[] combine = new CombineInstance[3];
        combine [0].mesh = lowerMesh;
        Matrix4x4 matrix = filter.transform.localToWorldMatrix;
        Quaternion rotation = Quaternion.Euler(90.0f, 0.0f, 0.0f);
        matrix = Matrix4x4.Rotate(rotation);
        combine [0].transform = matrix;

        combine [1].mesh = upperMesh;
        Matrix4x4 matrix2 = filter.transform.localToWorldMatrix;
        Quaternion rotation2 = Quaternion.Euler(90.0f, 0.0f, 0.0f);
        matrix2 = Matrix4x4.Rotate(rotation2);
        combine [1].transform = matrix2;

        combine [2].mesh = sideMesh;
        Matrix4x4 matrix3 = filter.transform.localToWorldMatrix;
        Quaternion rotation3 = Quaternion.Euler(90.0f, 0.0f, 0.0f);
        matrix3 = Matrix4x4.Rotate(rotation3);
        combine [2].transform = matrix3;

        filter.mesh.CombineMeshes (combine);
        gameObject.GetComponent<MeshCollider> ().sharedMesh =
gameObject.GetComponent<MeshFilter> ().sharedMesh;

        this.drawLines (lowerVertices, upperVertices);

    }
```

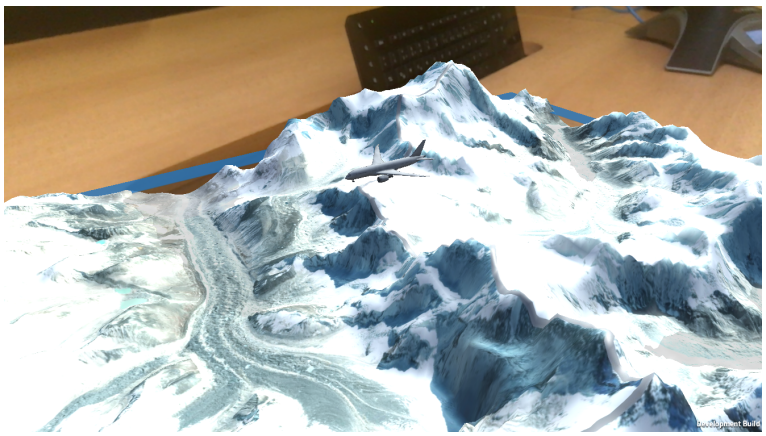
Obrázek 4.3. Funkce Update() pro game objekt sektoru.



Obrázek 4.4. Simulace v AG, pohled 1.



Obrázek 4.5. Simulace v AG, pohled 2.



Obrázek 4.6. Simulace v AG, pohled 3.

Kapitola 5
Závěr

Cílem práce bylo vytvoření návrhu a částečné implementace. Práci, tak jak je napsána, by mělo jít předat programátorovi a on podle ní implementuje systém, který bude plnit všechny definované funkční i nefunkční požadavky.

Na závěr bych také rád nastínil několik vizí pro budoucí návrh a implementaci software pro tvorbu letových tratí, prostorů a sektorů.

Technologie virtuální a augmentované reality má dle mého názoru obrovský potenciál a je velmi pravděpodobné, že je pouze otázkou času, než pronikne do každodenní praxe v řízení letového provozu. Hlavní výhodou bude opravdová vizualizace ve 3D, což při implementace návrhů, jako je Free flight, může být velmi výhodné. S využitím augmentované reality a zařízením pro zobrazování virtuální reality bude možné, aby několik řídících letového provozu pracovalo u jednoho stolu, a vzájemně kooperovali ve světě, který budou vidět před sebou, ovšem ten svět bude pouze projekce. Navrhovaný software by tak mohl sloužit jako testovací prostředek pro zavádění těchto technologií do praxe.

Z důvodu časové tísně se v práci nenachází kapitola o testování. Má vize pro testování byla využít automatických testů tak, aby pouštěly aplikaci, simulovaly provoz a následně vyhodnocovaly, zdali byly naplněny definované testovací scénáře, případně s jakými odchylkami. Výhody tohoto přístupu jsou získání odchylek v navigaci při spuštěné simulaci a z programátorského hlediska by měly zaručit okamžitou detekci zanesené chyby do již hotového řešení, protože při narušení funkčních celků začnou tyto automatické testy detekovat odlišnosti v simulacích.

Díky podpoře síťových funkcí bude také možné systém využít pro výukové programy řídících letového provozu. Díky scénářům letového provozu půjde názorně předvádět postupy užívané v praxi.

Vzhledem k podpoře více platforem, včetně mobilních zařízení, je také možné software upravit tak, že vznikne počítačová, či mobilní hra pro simulaci řízení letového provozu.

Všechny zmíněné vize jsem měl na paměti po celou dobu návrhu, a tudíž by měl být návrh dostatečně flexibilní, aby bylo možné tyto vize snadno realizovat.

Literatura

[1] Ing. Petr Veselý. *Zavedení postupů navigace podle požadavků PBN (Performance Based Navigation) na regionálním letišti*. Disertační práce, 2015. http://hdl.handle.net/11012/38093.

[2] Ph.D. Doc. Ing. Jiří Šebesta. *Globální navigační systémy*. 2012. http://www.urel.feec.vutbr.cz/~sebestaj/RAR/literatura/Globalni_navigacni_systemy.pdf.

[3] s.p. Řízení letového provozu České republiky. *Vzdušný prostor České republiky*. 2018. https://lis.rlp.cz/vfrmanual/actual/pdf/enr_1_cz.pdf.

[4] masatm@seznam.cz. *Letecký zákon - letové provozní služby*. http://aa.fd.cvut.cz/wp-content/uploads/2013/06/letove-provozni-sluzby.pdf.

[5] Úřad procivilní letectví. *Kde se nachází jaký druh vzdušného prostoru?* http://www.caa.cz/letadla-bez-pilota-na-palube/kde-se-nachazi-jaky-druh-vzdusneho-prostoru-tma-ctr-atz.

[6] s.p. Řízení letového provozu České republiky. *Vertikální rozdělení a pravidla užívání vzdušného prostoru ČR*. 2017. http://www.laacr.cz/SiteCollectionDocuments/rozdeleni-vp/2017%20rez%20VP.pdf.

[7] s.p. Řízení letového provozu České republiky. *Letové provozní služby*. 2017. https://lis.rlp.cz/vfrmanual/actual/pdf/gen_6_cz.pdf.

[8] s.p. Řízení letového provozu České republiky. *Naše služby*. 2018. http://www.rlp.cz/sluzby/nase/Stranky/default.aspx.

[9] s.p. Řízení letového provozu České republiky. *Letecká informační příručka*. 2018. http://lis.rlp.cz/ais_data/www_main_control/frm_cz_aip.htm.

[10] Bc. Aleš Oharek. *Zvyšování bezpečnosti a výkonnosti navigace dopravních letadel po trati letu*. 2009. https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=16243.

[11] Dr. Andrew Cook. *European Air Traffic Management: Principles, Practice, and Research*. Ashgate Publishing, Ltd., 2007.

[12] European Organisation forthe Safety of Air Navigation(EUROCONTROL). *Introducing Performance Based Navigation (PBN) and Advanced RNP (A-RNP)*. 2013. http://www.eurocontrol.int/sites/default/files/publication/files/2013-introducing-pbn-a-rnp.pdf.

[13] Ph.D. Ing. Stanislav Pleninger. *Přednášky z předmětu CNS systémy*. 2017.

[14] Ph.D. Ing. Miloš Strouhal. *Přednášky z předmětu Air Traffic Management*. 2017.

[15] Ing. Přemysl Poláš. *Aplikace Free flight v rámci evropského vzdušného prostoru*. 2012. https://www.vutbr.cz/studenti/zav-prace/detail/47431.

[16] National Research Council. *The Future of Air Traffic Control: Human Operators and Automation*. Washington, DC: The National Academies : Ashgate Publishing, Ltd., 1998.

[17] Martin Čáp. *Porovnání herních platforem Unity a Unreal Engine*. 2017. https://dspace.cvut.cz/handle/10467/69059.

[18] Mapbox. *https://www.mapbox.com*. 2018. https://www.mapbox.com.

[19] Lukáš Skala. *Virtuální realita a její použití v herním průmyslu*. 2016. https://is.muni.cz/th/422258/fi_b/Bakalarka_Final.pdf.

[20] David Nield. *5 problems virtual reality needs to solve to go mainstream*. 2017. https://www.t3.com/news/5-problems-virtual-reality-needs-to-solve-to-go-mainstream.

[21] Bc. Roman Lukš. *Examining Motion Sickness in Virtual Reality*. 2018. https://theses.cz/id/hu85by.

[22] Lucie Chatrná. *Využití virtuální reality v rámci filmové tvorby*. 2017. https://is.muni.cz/th/428641/ff_b/BAKALARSKAPRACE-LCH.pdf.

[23] Peter Eeles Peter Cripps. *Architektura softwaru: Nepostradatelný průvodce návrhem softwarové architektury, která funguje*. Computer Press, 2011.

[24] Lan Gorton. *Essential Software Architecture*. Springer, 2011.

[25] MICROSOFT PATTERNS a PRACTICES TEAM. Microsoft. *Application Architecture Guide*. 2009. http://msdn.microsoft.com/en-us/library/ff650706.aspx.

[26] Mark Cade. *Sun Certified Enterprise Architect for Java EE Study Guide*. 2010.

Příloha A
Zkratky a symboly

A.1 Zkratky

ACAS	Airborne collision avoidance system (Palubní protisrážkový systém)
ADF	Automatic direction finder (Automatický radiokompas)
AG	Augmented reality (Augmentovaná realita)
AGL	Above ground level (Výška nad povrchem Země)
AIP	Aeronautical information publication (Letecká informační příručka)
AL	Alert limit
ARDA	Airspace and Route Design Analyzer
ASAS	Airborne Separation Assistance System (Palubní systém pro podporu rozestupů)
ASM	Air space management (Uspořádání vzdušeného prostoru)
ATC	Air traffic control (Řízení letového provozu)
ATFCM	Air traffic flow and capacity management (Uspořádání toku a kapacity letového provozu)
ATM	Air traffic management (Uspořádání letového provozu)
ATS	Air traffic services (Služby letového provozu)
ATZ	Aerodrome traffic zone (Letištní provozní zóna)
CD&R	Conflict Detection & Resolution (Detektor a řešitel konfliktu)
CDTI	Cockpit Display of Traffic Information
CNS	Communication navigation surveillance (Komunikace navigace dohled)
CTR	Control zone (Řízený okrsek)
DME	Distance measuring equipment (Měřič vzdálenosti)
FANS	Future Air Navigation System
FTE	Letově technická chyba (Flight technical error)
GNSS	Global Navigation Satellite System (Globální družicový polohový systém)
GS	Ground speed (Traťová rychlost)
IAS	Indicated airspeed (Indikovaná vzdušná rychlost)
ICAO	International Civil Aviation Organization (Mezinárodní organizace pro civilní letectví)
IFR	Instrument flight rules (Let podle přístrojů)
ILS	Instrument landing system
INS	Inertial navigation system (Inerciální navigační systém)
IR	Integrity risk
JSON	JavaScript Object Notation
LNS	Letové navigační služby
NDB	Non directional beacon (Nesměrový radiomaják)
NSE	Chyba navigačního systému (Navigation system error)

PASAS	Predictive Airborne Separation Assistance System (Prediktivní palubní systém pro podporu rozestupů)
PDE	Chyba definice letové cesty (Path definition error)
PS4	Playstation 4
REST	Representational state transfer
RNAV	Area navigation (Prostorová navigace)
SSL	Secure sockets layer (vrstva bezpečných socketů)
SSR	Secondary surveillance radar (Sekundární radar)
TCP	Transmission Control Protocol
TMA	Terminal control area (Koncová řízená oblast)
TMZ	Transponder mandatory zone (Oblast s povinným odpovídačem)
TSE	Celková chyba systému (Total system error)
TTA	Time to alert
VFR	Visual flight rules (Let za viditelnosti)
VOR	VHF omnidirectional radio range (Všesměrový radiomaják)
VHF	Very high frequency (Velmi krátké vlny)
VR	Virtual reality (Virtuální realita)

A.2 Symboly

ft	Feet (Stopa)
kt	Knot (Uzel)
M1	Mach 1