

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

GRAFICKÉ ROZHRANÍ SYSTÉMU AUTOPILOTA

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

PAVEL ZUBATÝ

BRNO 2014



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

GRAFICKÉ ROZHRANÍ SYSTÉMU AUTOPILOTA

GRAPHIC USER INTERFACE FOR AUTOPILOT

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

PAVEL ZUBATÝ

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. PETER CHUDÝ, Ph.D.

BRNO 2014

Abstrakt

Tato práce je zaměřena na implementaci grafického rozhraní systému autopilota. Samotnému popisu implementace předchází popis prostorového modelu pohybu letounu, historie vývoje systému autopilota a návrh rozhraní tohoto systému. Cílem bylo vytvoření grafického rozhraní pro dotykové obrazovky a nastínění možného budoucího vývoje.

Abstract

This thesis is aimed on the graphic user interface for the autopilot systems. Before the description of the implementation itself, there is part describing flight model of an aircraft, the history of autopilot systems and finally the draft of the interface. The thesis describes creation of the graphic interface for touchscreens, it's integration into the simulator and an outline of the further development.

Klíčová slova

aeroworks, systém autopilota, autopilot, grafické rozhraní, dotyková obrazovka

Keywords

aeroworks, autopilot system, autopilot, graphic interface, touchscreen

Citace

Pavel Zubatý: GRAFICKÉ ROZHRAŇÍ SYSTÉMU AUTOPILOTA, bakalářská práce, Brno, FIT VUT v Brně, 2014

GRAFICKÉ ROZHRANÍ SYSTÉMU AUTOPILOTA

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Petera Chudého Ph.D. MBA. Další informace mi poskytl Mgr. Karol Rydlo. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....

Pavel Zubatý
21. května 2014

Poděkování

Chtěl bych poděkovat Ing. Peteru Chudému Ph.D. MBA za užitečné rady a trpělivost při vedení mé bakalářské práce. Dále bych chtěl poděkovat Mgr. Karolovi Rydlovi za užitečné rady při návrhu a pomoc při testování propojení implementační části práce a leteckého simulátoru.

© Pavel Zubatý, 2014.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

1	Seznam zkratek	2
2	Úvod	3
3	Model prostorového pohybu letounu	4
3.1	Geocentrická souřadná soustava	4
3.2	Letadlová souřadná soustava	5
3.3	Aerodynamická souřadná soustava	5
4	Požadavky kladené na systém autopilota	7
4.1	Historie systému autopilota	7
4.2	Popis systému autopilota	9
4.3	Použité veličiny a přístroje	10
4.4	Požadavky kladené na systém autopilota	11
5	Návrh	13
5.1	Motivace	13
5.2	Požadavky	13
5.3	Návrh vlastního řešení	14
6	Implementace	18
6.1	Použité technické prostředky	18
6.1.1	Framework Qt 5.2	18
6.1.2	Knihovna OpenGL	19
6.1.3	Mapové podklady OpenStreetMaps	19
6.2	Laboratoř 737-800 NG	21
6.2.1	Simulační model	21
6.2.2	Popis komunikačního protokolu	22
6.3	Inicializace aplikace	23
6.3.1	Načítání inicializačního souboru	23
6.3.2	Načítání mapových podkladů	24
6.3.3	Načítání modelu OBJ	25
6.4	Zobrazení scény	26
6.4.1	Výběr objektu scény	27
6.4.2	Výběr souřadnic mapy	29
6.4.3	Navigační body	29
6.5	Testování aplikace	30
6.6	Možné rozšíření	30
7	Závěr	31
A	Obsah CD	35
B	Konfigurační soubor	36

1 Seznam zkratek

AGL	Above Ground Level	Výška nad zemí
ALT	Altitude	Výška nadmořská
ALT HLD	Altitude Hold	Režim udržování aktuální výšky letu
AP	Autopilot	Systém automatického řízení letu
APR	Approach	Režim přistání pomocí ILS
CAS	Calibrated Air Speed	Kalibrovaná vzdušná rychlost
CAT IIb	Category 3b ILS	Přistání v nulové viditelnosti s využitím systému ILS
CAT IIc	Category 3c ILS	Přistání v nulové viditelnosti s využitím systému ILS a následným pojížděním po přistávací dráze
EAS	Equivalent Air Speed	Ekvivalentní vzdušná rychlost
FAA	Federal Aviation Administration	Americký Federální úřad pro letectví
FIT	Faculty of Information Technology	Fakulta Informačních Technologií
FL	Flight Level	Letová hladina
GPS	Global Position System	Vojenský globální poziční systém
GS	Ground Speed	Rychlost vzhledem k zemi
HDG	Heading	viz HEA
HEA	Heading	Kurz magnetický
IAS	Indicated Air Speed	Indikovaná vzdušná rychlost
ILS	Instrument Landing System	Systém asistovaného přistání pro přistání za zhoršených povětrnostních podmínek
LAT	Latitude	Zeměpisná šířka
LON	Longitude	Zeměpisná délka
MCP	Mode Control Panel	Panel nastavení autopilota a souvisejících systémů
MSL	Mean Sea Level	Střední hladina moře
M	Mach	Machovo číslo
NAV	Navigation Point	Navigační bod
OBJ	Wavefront .obj file	Textový soubor reprezentující 3D model
QML	Qt Meta-object Language	Meta-objektový jazyk podobný jazyku Javascript, součást frameworku Qt Quick
SET AP	Set Autopilot	Aktivace systému autopilota
TAS	True Air Speed	Skutečná vzdušná rychlost
TRK	Track	viz TAS
TT_	True Track	viz TAS
VOR	VHF Omnidirectional Radio range	radiomaják
VS_	Vertical Speed	Vertikální rychlost
VS	Vertical Speed	viz VS_
VUT		Vysoké Učení Technické v Brně

2 Úvod

Systém automatizovaného řízení letu, zkráceně autopilot, je od počátků letectví její nedílnou součástí. Autopilot umožňuje efektivní řízení moderního letadla tím, že usnadňuje práci pilota a přebírá za něj řadu úkonů. Je přitom napojený na řídicí plochy i senzory letounu, které měří relevantní veličiny. Každá chyba může vést k nežádoucímu chování letadla s nežádoucími až katastrofálními následky. Z toho důvodu jsou na každou součást systému autopilota kladeny vysoké nároky, což platí i pro uživatelské rozhraní tohoto systému.

Tradičně zde bývá použito čistě hardwarové řešení, ale s rozvojem digitální techniky se objevují systémy využívající dotykových obrazovek. Mezi jejich výhody patří nižší náklady na zapojení, možnost zachycovat větší množství informací na malé ploše, ale též intuitivnost ovládání. Právě intuitivnost vhodně navrženého rozhraní je při návrhu ovládání avioniky vyžadována. Pilot se v takovém případě může soustředit na chování letadla a věnovat jen minimum pozornosti ovládání jednotlivých přístrojů.

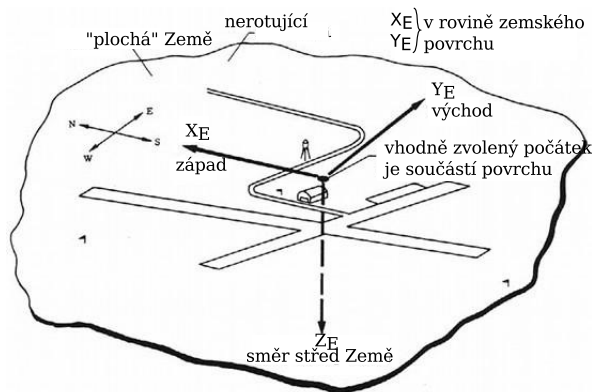
Cílem práce je návrh rozhraní systému autopilota, což v sobě zahrnuje popis relevantních fyzikálních veličin a historie vývoje tohoto systému. Z těchto údajů pak vychází návrh vlastního řešení implementující rozhraní pro dotykový displej využitelný v leteckém simulátoru, který je umístěn v prostorách Fakulty Informačních Technologií Vysokého učení technického v Brně (dále jen FIT VUT). Součástí práce je rovněž nastínění budoucího vývoje ovládání systému autopilota.

3 Model prostorového pohybu letounu

Na rozdíl od pohybu po zemi probíhá pohyb letounu ve třech dimenzích. Pro jeho popis se proto využívá níže uvedená trojice trojrozměrných kartézských systémů souřadnic. Veličiny v takovém systému uváděné pak definují ovládání letounu (vč. řízení autopilotem). Konkrétně se jedná o pohyb letounu vzhledem k zemskému povrchu, vlastnímu těžišti a vzhledem k proudění okolního vzduchu. Uvedené informace jsou převzaty z [4, 6, 13, 17].

3.1 Geocentrická souřadná soustava

Jelikož je uvažován pouze normální atmosférický let, tak je konvencí využívat systém s fixně definovaným povrchem Země. Počátek pravotočivého souřadnicového systému je umístěn na povrchu zemském a jedna z os směřuje do středu Země (viz 3.1). Při rotaci planety tyto souřadnice rotují tak, že geograficky zůstává počátek ve stejném umístění.

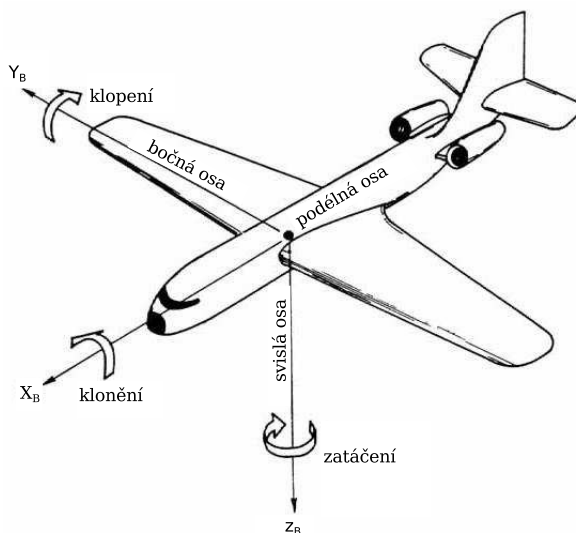


Obr. 3.1: Geocentrická souřadná soustava, převzato [17]

Počátek souřadnicového systému lze proložit rovinou, která bude tečná k povrchu. Vzniká tak model s deskovitým zemským povrchem, ve kterém je vertikální osa zaměnitelná za gravitační vektor. Tento model je vhodný pro navigaci nad menším územím a ovládání letounu z hlediska směru letu. Není však vhodný pro popis letu na delší vzdálenosti, kde už je třeba uvažovat zakřivení planety. Zde se používá sférický model.

3.2 Letadlová souřadná soustava

Počátek tohoto souřadnicového systému je umístěn v těžišti letounu. Konvence udává, že osa x prochází tělem trupu. Tato podélná osa prochází přídělí trupu (zpravidla v místě přídového kuželu). K ní je kolmá tzv. boční osa, jenž prochází rovinou křídel a ven směřuje přes pravé křídlo. Zpravidla se označuje jako osa y . Například ve starší literatuře je možné narazit na prohozené značení (nebude zde použito), kde je tímto písmenem označena tzv. kolmá osa letounu. Ta je standardně označena jako osa z , je kolmá na obě předešlé osy a směřuje dolů.



Obr. 3.2: Letadlová souřadná soustava, převzato [17]

Osy tohoto systému lze využít pro definici otáčivých pohybů, které letoun vykonává kolem svého těžiště při pohybu prostorem. Těmito pohyby je myšleno *klonění*, *klopení* a *zatáčení*. Reprezentují je tzv. Eulerovy úhly, což jsou úhly definované pravidlem pravé ruky kolem tří os souřadné soustavy.

Klonění: Při tomto pohybu dochází k naklonění letounu podle osy x . Kladné klonění je definováno jako otáčení osy y směrem k ose z (původnímu směru osy z). Dochází tak k poklesu vnější hrany pravého křídla. Klonění se docílí vychýlením křídélek. Označuje se řeckým písmenem ϕ (phi).

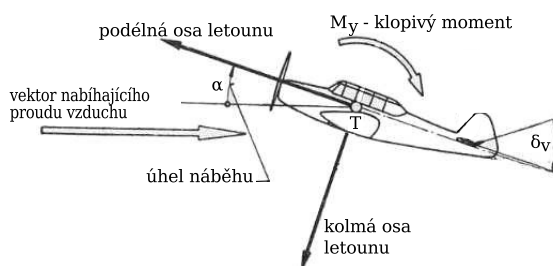
Klopení: Při klopení se letoun otáčí kolem osy y . Kladné klopení je definováno jako otáčení osy x směrem k ose z . Dochází tak ke zvedání přídělí letounu. Klopení se docílí vychýlením výškového kormidla. Označuje se řeckým písmenem θ (theta).

Zatáčení: Zatáčení je pohyb letounu kolem osy z . Kladné zatáčení je definováno jako otáčení osy x směrem k ose y . Z pohledu pilota přitom dochází k otáčení doprava. K tomuto pohybu slouží směrové kormidlo. Označuje se řeckým písmenem ψ (psi).

3.3 Aerodynamická souřadná soustava

V obecné aerodynamické souřadné soustavě je počátkem soustavy těžiště letounu. Základní osa této soustavy [13] je osa x . Označuje se jako odporová osa. Její směr je opačný proti

směru působení odporové síly. Osa z této soustavy směřuje směrem dolů. Označuje se jako vztlaková osa, vztlaková síla působí proti jejímu směru. Třetí osa je tzv. příčná osa y , která u nehybného letounu prochází osou pravého křídla. Tyto osy umožňují určit aerodynamické úhly.

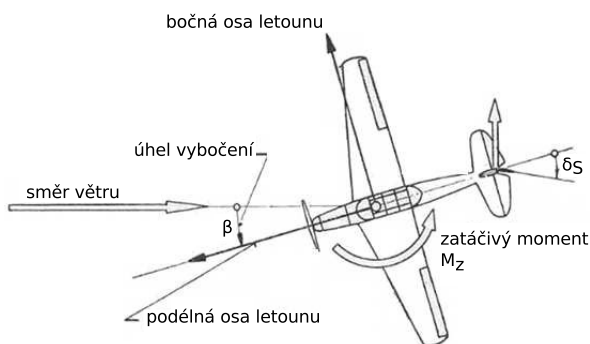


Obr. 3.3: Úhel náběhu, převzato [13]

Úhel náběhu α : Je dána podélná vertikální rovina, která je definována podélnou a kolmou osou letadlové souřadné soustavy. Úhel náběhu je úhel svíraný mezi projekcí osy x a podélnou osou letounu v této rovině. Kladný úhel náběhu odpovídá natočení křídla (letounu) nahoru proti proudu vzduchu. Úhel náběhu ovlivňuje celkový vztlak letounu. Ten zjednodušeně popisuje rovnice 3.1.

$$F_{vz} = C_y \cdot q \cdot S \quad (3.1)$$

Veličina q označuje dynamický tlak, S plochu a C_y součinitel vztlaku. Vedle úhlu náběhu tato účinnost závisí na hloubce, tvaru a výchylce výškového kormidla.



Obr. 3.4: Úhel vybočení, převzato [13]

Úhel vybočení β : Je dána podélná horizontální rovina, která je definována podélnou a bočnou osou letadlové souřadné soustavy. Úhel vybočení je úhel svíraný mezi projekcí osy x a podélnou osou letounu v této rovině. Kladný úhel vybočení odpovídá natočení letounu pravou stranou proti proudu vzduchu.

4 Požadavky kladené na systém autopilota

Systém autopilota lze definovat jako automatický systém řízení letu, který udržuje nastavenou letovou hladinu nebo kurz letounu. Autopilot může být nastaven pilotem nebo řízen dálkově. Použití autopilota snižuje pracovní zátěž kladenou na pilota a obecně zvyšuje bezpečnost letu [8].

4.1 Historie systému autopilota

Za vynálezce systému automatického řízení letu je považován Lawrence Sperry, jenž r. 1912 získal patent zařízení *automatické gyroskopické stabilizace*. Vycházel přitom z gyrokompasu, který vynalezl jeho otec Elmer Sperry. Tento americký vynálezce a zakladatel Sperry Gyroscope Company za svůj život podal přes 400 individuálních žádostí o patent. Jeho gyrokompas byl tehdy využíván na více než 30 lodích Vojenského námořnictva Spojených států amerických [28]. Tyto gyroskopy však byly rozměrnější a těžší než nová verze, kterou Lawrence Sperry vytvořil pro použití v letectví [24].



Obr. 4.1: První autopilot byl představen r. 1914, převzato [28]

Sperryho autopilot korigoval směr letounu na základě výchylek, které detekovala soustava čtyř gyroskopů rotujících rychlostí 7000 otáček za minutu. Tento systém byl hydraulicky napojen na ovládání řídicích ploch letadla, čímž udržoval stabilitu všech tří os letadlové souřadné soustavy (viz 3.2), především pak směr a výšku. Pokud se jeden z gyroskopů pohyboval proti směru letounu, došlo ke stisku pístu. Stlačený vzduch následně automaticky natočil napojenou řídicí plochu, čímž se letoun stabilizoval. Vedle toho byl systém vybaven anemometrem. V případě nepřiměřeně vysoké rychlosti systém způsobil zvýšení úhlu náběhu letounu, a snížení rychlosti [24].

Širší veřejnosti byl systém představen v červnu 1914 na přehlídce *Concours de la Sécurité en Aéroplane* – soutěži zaměřené na bezpečnost letadel, která se konala v Paříži u řeky Seiny. Autopilot byl umístěn na dvouplošný hydroplán Curtiss C-2. Při prvním průletu nad diváky Sperry demonstroval směrovou stabilitu zvednutím rukou, letoun „řídil“ pouze autopilot. Při druhém průletu dokonce francouzský mechanik Emil Cachin vylezl na křídlo (viz 4.1). Došlo k naklonění křídla do strany, které však autopilot vykompenzoval zpětným

natočením křídla. Při třetím průletu stál Cachin na jednom křídle, Sperry na druhém, kokpit byl dočasně prázdný. Na této přehlídce rovněž byla demonstrována schopnost systému plně řídit vzlet a přistání [24].

V r. 1932 přišla společnost Sperry Gyroscope Company s modelem autopilota, který v r. 1933 použil Wiley Post při prvním sólovém letu kolem světa [10]. Zvyšující se nároky vojenského letectva však vyžadovaly přesnější systémy automatického řízení, než jaké mohl gyroskopický systém nabídnout. Přesné navádění leteckých bomb nad cíl vyžadovalo přesné udržování kurzu, letové hladiny, polohy vůči umělému horizontu a nastavení ovládacích ploch, Komerční hydraulický systém konce 30. let 20. stol. ani osádka letounu takové přesnosti zdaleka nedosahoval [16].

Řada letounů B-17 byla v té době vybavena komerčním autopilotem Sperry A-3. Gyroskopy tohoto systému využívaly těžkopádné pneumatické hydraulické servosystémy. Žádný z přístrojů však neměřil rychlost ani akceleraci po úhlové dráze. Kompenzace systému za špatných letových podmínek způsobovaly oscilaci systému. Carl Frische, výzkumník společnosti, proto přišel s kompletně elektronickým autopilotem, který dosahoval vyšší přesnosti než pneumatický Sperryho i konkurenční systém elektrických serv společnosti Norden [16].



Obr. 4.2: Autopilot letounu Avro 19, r. 1947, převzato [11]

Sperry A-5 byl založen na trojici dvouprvkových elektronkových zesilovačů, z nichž každý zodpovídal za jiný úhlový pohyb letounu (klonění, klopení, zatačení). Pro zajištění spolehlivosti byly elektronky podrobeny sérii testů zrychlené životnosti, teplotních cyklů a vibrace. Každá elektronka zesilovala slabý signál získaný čidly autopilota umístěných na vysokorychlostních gyroskopech. Systém autopilota navíc bral v potaz i rychlost a akceleraci po úhlové dráze zjištěné ze změny signálu. Zesílené signály ovládaly nezávislé elektro-hydraulické servomechanismy, které stabilizovaly letoun. Výsledný systém byl rychlejší a reagoval rychleji na řídicí signály a poryvy větru. K prvnímu využití v praxi došlo 30. dubna 1941 v britském bombardéru, který poblíž dánského Tyboronu úspěšně navedl bombu na německou loď [16].

Od té doby prošel systém autopilota řadou vylepšení. Přesto i dnes existují letouny, které nejsou tímto systémem vybaveny. Především jde o starší a menší letouny s kapacitou pod 20 pasažérů. U větších letounů je systém autopilota vyžadován mezinárodními leteckými dohodami. Autopilot určený pro malé letouny nemusí kontrolovat všechny tři osy, ale může kontrolovat jen stabilizaci křídélek (klonění), popřípadě stabilizaci křídélek a výškového kormidla (klonění a klopení) [10].

Autopilot ve velkém moderním dopravním letadle (např. Boeing B737 nebo Airbus A380) kontroluje všechny tři osy. Moderní autopilot dokáže řídit všechny části letu, tj. vzlet z letiště, stoupání, let v dané letové hladině, klesání, přiblížení na přistání, přistání.

Autopilotem řízené přistání na přistávací dráze s udržováním letadla uprostřed dráhy je označováno jako přistání kategorie CAT IIb neboli automatické přistání. Mnoho dnešních velkých letišť používá tento systém, obzvlášť pokud se jedná o letiště s častým zhoršením počasí (např. mlha). Přistání kategorie CAT IIc v sobě zahrnuje i pojíždění letadla po letišti, ale zatím není v praxi běžné [10].



Obr. 4.3: Boeing 787, rozhraní autopilota nahoře uprostřed, převzato [12]

Moderní systémy autopilota jsou založeny na počítačovém softwaru, který ovládá letoun. Na základě aktuální polohy letounu tento software ovládá systém letové kontroly. Vedle klasických funkcí autopilota dokáže takový systém ovládat řízení tahu motorů pro optimalizaci rychlosti letu, přečerpávat palivo mezi nádržemi pro vyrovnání letadla. Let s autopilotem tak mívá nižší spotřebu paliva než let řízený lidským pilotem. Systém automatického řízení letu však nemusí reagovat dostatečně flexibilně na nečekané a nebezpečné situace, což je jejich nevýhodou [10]. Informace uvedené v sekci 4.1 jsou přejaty z [10, 11, 16, 24, 28].

4.2 Popis systému autopilota

Systémy automatického pilota lze nalézt v řadě dopravních prostředků (vč. řízených střel). Zajišťují přitom jednu či více činností, které by jinak musel vykonávat lidský pilot. V případě letectví je tento systém nazýván jako systém automatického řízení letu [7].

Tento systém je součástí systému avioniky, který v sobě zahrnuje přístroje pro ovládání letu, komunikaci, navigaci, systémy varování i monitorování počasí. Jak už bylo uvedeno výše, primárním určením systému autopilota je snížení pracovní zátěže kladené na pilota (např. dlouhé stoupání). Dnešní systémy však dokážou provést i přistání v nulové viditelnosti [7]. Systémy autopilota lze rozdělit do tří základních kategorií:

- Jednakanálové systémy ovládající 1 osu. Udržují laterální stabilitu pomocí křidélek.
- Dvoukanálové systémy ovládající 2 osy. Ovládají křídélka a výškové kormidlo.
- Tříkanálové systémy ovládající 3 osy. Ovládají křídélka, výškové i směrové kormidlo.

Původně tříkanálové systémy navíc bývají vybaveny ovládáním přípustí (tahu) motoru. V takovém případě se jedná o čtyř a vícekanálové systémy. Běžný tříkanálový systém je schopen několika režimů. Jedním je laterální režim udržující směr letu. Ten je možné nastavit pomocí magnetického kurzu (HDG) nebo jiného zdroje navigace (VOR, GPS). Dále

to je režim vertikální navigace, jenž lze stanovit jako udržování aktuální výšky nebo rychlosti stoupání/klesání. Takový systém lze nastavit, aby udržovat daný směr letu až do určitého navigačního bodu (NAV). Po dosažení tohoto bodu plynule pokračuje podle nového kurzu. Stejnou metodu lze aplikovat i na režim vertikální navigace, kdy letoun stoupá do určité výšky (ALT), kterou následně autopilot udržuje. Režim APR navádí letoun k nejbližšímu bodu systému ILS pro asistované přistání. Informace uvedené v sekci 4.2 jsou přejaty z [2, 7, 8, 11].



Obr. 4.4: Ovládání autopilota letadla Boeing 737

4.3 Použité veličiny a přístroje

Veličiny použité v této práci vychází z veličin protokolu AW-COM, použitého pro leteckou simulaci na FIT VUT. Jedná se přitom o veličiny, které je možné najít na stávajících rozhraních systému autopilota, případně veličiny s tímto systémem související.

Indikovaná vzdušná rychlost (angl. *Indicated Airspeed, IAS*) je údaj získaný přímým odečtem z indikátoru letové rychlosti (rychloměru), který využívá pitot-statický systém. Hlavními součástmi pitot-statického systému je sonda celkového tlaku a statická tlaková komora. Pitotova trubice (obvykle na křídle) přivádí proud vzduchu do sondy. Rychlost je následně určena rozdílem tlaku mezi sondou a statickou tlakovou komorou [21].

$$IAS = \sqrt{7 \frac{p_0}{\rho_0} \left[\left(\frac{p_t - p_a}{p_0} + 1 \right)^{\frac{1}{3,5}} - 1 \right]} \quad (4.1)$$

Rychloměr je kalibrován na normovaný atmosférický tlak (u hladiny moře) se zanedbáním stlačitelnosti příchozího proudu. Chybí jakákoliv korekce chyby vznikající rozdílnou hustotou atmosféry, chyb vzniklých instalací přístroje, případně přístrojových chyb. Tato rychlost je používána výrobcí jakožto indikátor výkonu letounu. Vzletová, přistávací a pádová rychlost bývá udávána jako IAS a nemění se s výškou ani teplotou.

Výpočet IAS lze zapsat rovnicí 4.1, kde p_0 je tlak vzduchu na hladině moře, ρ_0 hustota vzduchu na hladině moře, p_t tlak celkový a p_a tlak statický dané letové hladiny [3]. V letectví bývá IAS udávána v *uzlech* [kn]. Potom může být označena jako KIAS. Méně běžná je rychlost uváděná v *km/h*.

Kalibrovaná vzdušná rychlost (angl. *Calibrated Airspeed, CAS*) je IAS s opravou instalační, poziční (umístění pitot-statického systému) a přístrojové chyby. Kalibrovaná dopředná rychlost odpovídá skutečné dopředné rychlosti ve standardních atmosférických podmínkách v úrovni hladiny moře. Instalační, poziční a přístrojová chyba je ale většinou zanedbatelná, proto se místo CAS používá IAS.

Ekvivalentní vzdušná rychlost (angl. *Equivalent Airspeed, EAS*) je CAS s proudem příchozího vzduchu upraveného pro danou výšku (je brána v potaz stlačitelnost vzduchu).

Ekvivalentní dopředná rychlost odpovídá CAS ve standardní atmosféře na hladině moře. Pokud je tedy poziční a přístrojová chyba zanedbatelná, platí opět rovnice 4.1.

Skutečná vzdušná rychlost (angl. *True Airspeed*, TAS) je CAS upravená pro danou výšku a nestandardní teplotu. Jedná se o rychlost letounu vzhledem ke stabilnímu proudění vzduchu. S rostoucí výškou hustota vzduchu klesá, letoun se proto musí pohybovat vyšší rychlostí, aby dosáhl stejného rozdílu tlaků mezi tlakem v impaktivě a statické komoře. Pro danou CAS tak skutečná dopředná rychlost roste s tím, jak roste výška. CAS tak pro určitou skutečnou dopřednou rychlost s rostoucí (nadmořskou) výškou letu klesá.

Machovo číslo M je poměr TAS vůči rychlosti zvuku v dané letové výšce (rovnice 4.2, kde c_s značí rychlost zvuku v dané letové hladině). Mach 1 označuje rychlost zvuku.

$$M = \frac{TAS}{c_s} \quad (4.2)$$

Rychlost vzhledem k zemi (angl. *Ground Speed*, GS) je skutečná rychlost letounu vůči zemi. Rychlost vzhledem k zemi se snižuje s větrem působícím proti směru letounu, ale zvyšuje se s větrem působícím ve směru letu letounu. Rychlost vzhledem k zemi odpovídá nějakému referenčnímu bodu (stanici, směrovému bodu systému autopilota, apod.).

Vertikální rychlost (angl. *Vertical Speed*, VS) udává rychlost stoupání nebo klesání obvykle ve stopách za minutu (ft/min). Jedna stopa odpovídá 0,3048 m [4]. Hodnotu VS lze vyčíst z indikátoru vertikální rychlosti (VSI, Vertical Speed Indicator), který využívá pitot-statický systém podobně jako rychloměr letounu.

Kurz magnetický (angl. *Heading*, HDG) vychází z klasického kompasu, jehož střelka ukazuje k magnetickému severu (nebo jihu). Indikátor směru neboli směrový gyroskop je přístroj citlivý na pohyb letounu a zobrazuje jeho magnetický kurz na displeji s azimutem rozsahu $\langle 0^\circ, 359^\circ \rangle$. Směrový gyroskop je vlastně magnetický kompas, který je oproti klasickému kompasu méně náchylný vůči chybám (způsobených např. turbulencemi okolního vzduchu). Využívá přitom magnetosféry Země, jejíž magnetický sever (jih) neodpovídá severu (jihu) geografickému, ale je o něco posunut.

GPS kurz využívá vojenskou síť GPS namísto magnetosféry nebo sítě radiomajáků. GPS neboli globální systém určení pozice využívá soustavu družic s celosvětovým pokrytím. Tato soustava umožňuje přijímačům určit jejich polohu (zeměpisnou délku, šířku a výšku) s přesností na metry i skutečný čas. Přijímač GPS musí být v dohledu alespoň tří družic pro dvourozměrnou pozici a pohyb. Pokud jsou v dohledu 4 a více satelitů, tak přijímač může obdržet trojrozměrnou pozici včetně pohybu ve 3 rozměrech. Větší počet satelitů znamená i vyšší přesnost [9].

Výška (angl. *Altitude*, ALT) typicky nad střední úrovní hladiny moře (MSL, Mean Sea Level), ve které letoun letí (pak se označuje jako absolutní výška). Obvykle se udává ve stopách [ft]. Standardně ji měří výškoměr pomocí změn tlaku [9].

Výška nad zemí (angl. *Above Ground Level*, AGL) je výška mezi letounem a zemí, nad kterou právě prolétá.

Letová hladina (angl. *Flight Level*, FL) udává výšku ve stovkách stop pro letouny ve výšce nad 18000 stop (5486 metrů) s tlakoměrem nastaveným na 1013,2 mb (milibarů). Letové hladiny proto určuje určitý tlak, který nemusí přesně odpovídat výšce.

Informace uvedené v sekci 4.3 jsou přejaty z [3, 8, 9, 14].

4.4 Požadavky kladené na systém autopilota

Moderní systémy autopilota jsou velice komplexní a jejich používání je klíčové pro práci pilota. Ten by jinak byl zbytečně zahlcen triviálními nebo zdlouhavými úkony, které autopilot ovládá (stabilizace os, stoupání do určité výšky). Podle [8] má pilot vynaložit 90 % času sledování dění mimo kokpit (pozice ostatních letounů, chování letounu, sledování skutečného horizontu). Pouze necelých 10 % má být určeno kontrole palubních přístrojů a konfrontaci jimi udávaných údajů s reálnou situací.

Dalším důvodem je bezpečnost. Kolem 80 % všech leteckých nehod bylo způsobeno selháním lidského faktoru [9]. Snížení pracovní zátěže (ať už fyzické nebo mentální) systémem autopilota je patrné především u řízení jedním pilotem. Pilot se tak může věnovat jiným povinnostem, má větší přehled o letové situaci a hrozí menší riziko letecké nehody (např. srážky se zemí). Nefunkčnost systému autopilota může zkomplikovat například přistání pomocí přístrojů na neznámém (pro pilota) letišti za špatných povětrnostních podmínek (hustá mlha, tmavá noc bez dostatečného osvětlení). Ještě nebezpečnější je selhání systému autopilota během kritické části letu, kdy pilot musí jednat pohotově bez přípravy alternativních řešení. Řada leteckých nehod nastala i kvůli selhání systému autopilota.

Příkladem může být poslední let vynálezce systému autopilota Lawrence Sperryho, který zahynul 23. prosince 1923 při letu v husté mlze nad kanálem La Manche. Jako možná příčina se uvádí přecenění schopností systému autopilota [28]. Dalším případem je smrt Johna F. Kennedyho (syna stejnojmenného amerického prezidenta) 16. července 1999. Jako příčina havárie malého letounu Piper PA-32R Saratoga II (a následné smrti všech tří osob na palubě) se uvádí dezorientace během nočního sestupu nad vodou [15].

Svou roli však mohlo sehrát i selhání dvoukanálového autopilota. Ten podle amerického Národního úřadu pro bezpečnost v dopravě (NTSB, National Transportation Safety Board) selhal jednou nebo dvakrát během tohoto letu, kdy změnil směr letounu. Autopilot tak musel být odpojen, a znovu připojen. V době fatálního pádu však byl autopilot odpojen a příčinou bylo selhání lidského faktoru [11].

Rozhraní systému autopilota proto musí být intuitivní, přehledné a nesmí pilota zbytečně zdržovat. V době psaní tohoto dokumentu probíhaly studie zkoumající využití dotykových panelů jako rozhraní pro ovládání avioniky letadel. Studii lidského faktoru dotykových systémů v kokpitu provádí společnost Honeywell s pověřením amerického Federálního úřadu pro letectví (FAA, Federal Aviation Administration). Další studie probíhá pod hlavičkou konsorcia ALICIA (All Condition Operations and Innovative Cockpit Infrastructure, Inovativní infrastruktura kokpitu pracující za všech podmínek) zaštiťovaného Evropskou komisí. Studie zkoumá, jak dotykové systémy snižují náklady na údržbu a výcvik pilotů [18].

Výrobci těchto dotykových systémů vycházejí z předpokladu, že lidské vnímání je primárně zrakové. Dotyková obrazovka přitom umožňuje zobrazit velké množství informací, a tím zredukovat plochu, kterou v pilotní kabině zabírají ovládací prvky avioniky.

Dotykové obrazovky v kabině bývají umístovány tak, aby na ně pilot snadno dosáhl. V kokpitech větších letadel bývají umístěny na středový sloupek. Je totiž blízko a lze o něj opřít ruku, což usnadňuje ovládání během turbulencí. V menších letounech výrobci preferují umístění na místě hlavních obrazovek vpředu před pilotem (kopilotem). Pilot tak má před sebou všechny údaje a může rychleji zareagovat na nečekanou situaci [18, 1, 5].

U dnešních dotykových obrazovek chybí hmatová odezva a hrozí riziko nechtěné aktivace některého aktivního prvku na obrazovce. To snižuje použití resistivní dotykové technologie. Vyžaduje větší tlak dotyku než projektivně kapacitní obrazovky, ale lze ji ovládat i v rukavicích. Nechtěné aktivace lze odfiltrovat pomocí specializované softwarové knihovny [18].

5 Návrh

5.1 Motivace

Cílem této práce bylo vytvoření uživatelského rozhraní systému autopilota pro simulátor velkého dopravního letadlo, který se nachází v prostorách FIT VUT. Výsledná aplikace je navržena pro dotykovou obrazovku.

5.2 Požadavky

Návrh konkrétního řešení uživatelského rozhraní systému autopilota vychází z existujících systémů, jakým je rozhraní v hlavním kontrolním panelu MCP firmy Rockwell Collins, který lze nalézt uvnitř Boeingu 737 řady 1278. Tento panel je využitý např. v simulátoru Boeingu 737 na FIT VUT [2]. Inspirace byla čerpána rovněž z dalších systémů.



Obr. 5.1: Rozhraní autopilota malého letadla, převzato [22]

Příkladem je rozhraní autopilota Triden firmy Century Flight Systems [22]. Tento autopilot je však určen pro sportovní letouny a ovládá pouze dvě osy. Výsledný návrh uživatelského rozhraní je přitom určen pro systém autopilota ovládajícího tři osy. Návrh musí reflektovat omezení používaných veličin a s nimi související hypotetické použití v reálném dopravním letadle. S minimálními úpravami by jej mělo být možné použít i v malém sportovním letounu. Konkrétně se pak jedná o nastavení těchto pěti veličin (viz sekce 4.2):



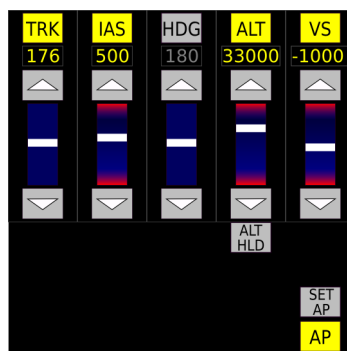
Obr. 5.2: Popis rozhraní autopilota v Boeingu 737

1. TRK – skutečný kurz ve stupních.
2. IAS – indikovaná vzdušná rychlost v uzlech.
3. HDG – magnetický kurz ve stupních.
4. ALT – výška nad hladinou moře ve stopách.
5. VS – vertikální rychlost ve stopách.

Z těchto pěti veličin byl následně vytvořen konceptuální návrh uživatelského rozhraní, který by byl určen pro dotykové obrazovky s několika omezeními. Ty jsou dána jednak cílovou technologií, jednak cílovou platformou školního simulátoru využívajícího software Microsoft Flight Simulator. Prvním omezením je rozlišení 540×540 px při fyzických rozměrech $16,5 \times 16,5$ cm, které platí pro obrazovku umístěnou na palubní desce uprostřed. Dále je třeba počítat s tím, že rozhraní má být přehledné a nenáročné na ovládání.

5.3 Návrh vlastního řešení

Prvotní návrh rozhraní obsahuje pět ovládacích prvků, které jsou mezi sebou odděleny mezerou. Zvolené rozměry byly zvoleny tak, aby šířka tlačítka odpovídala 1,8 cm. Každé veličině odpovídá jeden sloupec sestávající z tlačítka pro nastavení veličiny (např. „TRK“).



Obr. 5.3: Prvotní konceptuální návrh

Po své aktivaci se systém autopilota snaží dodržovat tuto veličinu. Dále sloupec obsahuje ukazatel hodnoty veličiny (např. „176“) a táhlo vč. šipek pro změnu hodnoty. Za normálního stavu jsou ovládací prvky neaktivní, aby nedošlo k nechtěné změně hodnoty a způsobení nechtěného chování celého letadla. Pro aktivaci ovládacích prvků je určeno tlačítko SET AP. Stisknutím tlačítka AP je aktivován systém autopilota.



Obr. 5.4: Prvotní návrh (implementace)

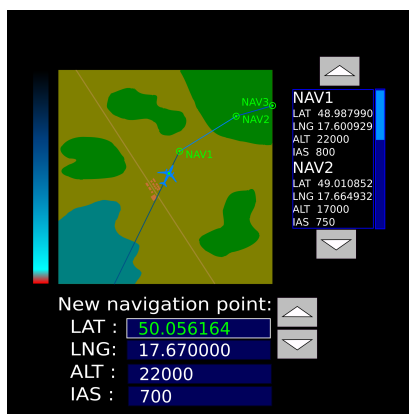
Aktivní veličiny jsou zde vyznačeny žlutou barvou. Některé veličiny však nemohou být aktivní zároveň (např. TRK a HDG). Tlačítko ALT HLD reprezentuje funkci Altitude Hold, při které autopilot dodržuje aktuální výšku letu. Aktivací ALT HLD je hodnota ALT

změněna na hodnotu aktuální výšky, ve které letadlo letí. Pro následnou implementaci bylo zvoleno vývojové prostředí QT Creator verze 3.0.1, využívajícím QT 5.2.1., framework jazyka C++.

Tlačítko SET AP bylo při implementaci odstraněno. Místo něj bylo pro změnu hodnoty určeno tlačítko hodnoty `valueButton` umístěné pod tlačítkem nastavení dané veličiny. Každou veličinu zde reprezentuje objekt s několika tlačítky a táhlem. Mezi definované atributy objektu patří výchozí, minimální a maximální hodnota veličiny i velikost přírůstku (např. „100 ft“).

Účelem tlačítka nastavení veličiny je zaslání hodnoty nadřazenému objektu. Ten reprezentuje entitu rozhraní v rámci systému simulátoru, spolu s údajem, zda-li je veličina nastavená. Tlačítko hodnoty zobrazuje aktuální hodnotu veličiny. Stiskem tlačítka hodnoty dochází ke změně barvy tlačítka, táhla a aktivaci táhla. Tím se spustí časovač, který po svém skončení vrací tlačítko hodnoty a táhlo do výchozího stavu. Dalším stiskem dochází k uložení nové hodnoty, změně barvy na výchozí a deaktivaci táhla. Táhlo sestává z táhla jako takového a dvojice tlačítek pro přesnou změnu hodnoty. V případě samotného táhla je nová hodnota zaokrouhlena podle velikosti přírůstku. Táhla jsou ve výchozím stavu neaktivní, což zabraňuje nechtěné změně hodnoty.

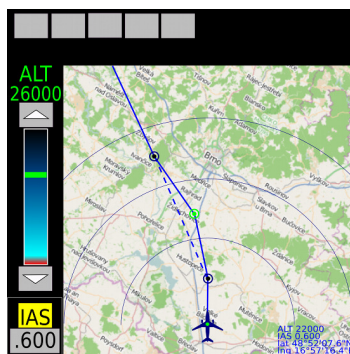
Prvotní návrh umožňuje nastavení pětice veličin, které jsou potřeba pro ovládání systému autopilota v simulátoru. Ošetřena je též nechtěná aktivace. Po těchto stránkách svůj účel splnil. Jeho slabou stránkou je však možná záměna jednotlivých veličin uživatelem, který se dostal do stresové situace. Navíc je poměrně konzervativní. Tyto nedostatky vedly k vytvoření druhého návrhu.



Obr. 5.5: Druhý návrh (koncept)

Druhý konceptuální návrh byl inspirován GPS navigací. Jeho základní myšlenkou je nastavení trajektorie letu s využitím mapy, kde trajektorii představovuje série uživatelem nastavených navigačních bodů neboli směrových bodů (angl. *waypoint*). Koncept počítá s existencí tlačítka úprav, které aktivovuje režim úprav. Jeho aktivní stav je navíc omezen časovačem. K jeho spuštění dochází, pokud je tlačítko aktivní, ale nedochází ke interakci uživatele s aplikací. Po ukončení časovače dochází k návratu do výchozího stavu aplikace. Tehdy sice rozhraní ukazuje aktuální polohu letounu, jeho výšku, rychlost. Není však možné provádět úpravy těchto veličin.

V režimu úprav je možné přidávat nové navigační body nebo upravovat ty stávající. Kliknutím do mapy dochází k vytvoření nového navigačního bodu s letovou výškou nastavenou podle táhla vlevo. Uživatel má rovněž k dispozici nabídku pro úpravu aktuálního vybraného nebo nového bodu (dole). Každý bod je v této nabídce reprezentován čtveřicí proměnných. Kliknutím na jednu z těchto hodnot dochází k aktivaci šipek pro inkrementaci/dekrementaci. Vpravo od mapy je umístěn seznam existujících navigačních bodů. Kliknutím na jeden z bodů dochází k jeho výběru a přesunu do nabídky pro úpravu. Tento koncept byl zavrhnut pro velké množství údajů znesnadňujících efektivní ovládání.

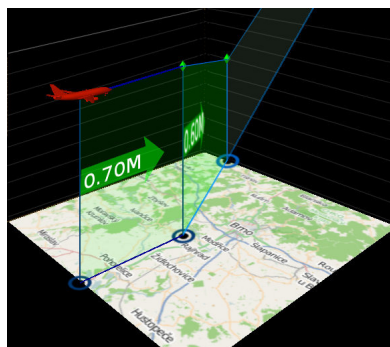


Obr. 5.6: Třetí návrh (koncept)

Třetí konceptuální návrh klade ještě větší důraz na výběr navigačních bodů v mapě. Aktuální hodnoty jsou zobrazeny v rohu mapy. Vedle nich je možné vidět znázornění vzdálenosti od letounu (trojice soustředných kruhů). Myšlenka tlačítka úprav zde byla zachována, stejně jako nastavení výšky pro každý vybraný navigační bod. Opět je možné vytvářet navigační body kliknutím do mapy. Dochází tím k vytvoření bodu o aktuální zeměpisné šířce a délce, výšce odpovídající táhlu vlevo a zadané rychlosti. Pro nastavení zadané rychlosti slouží tlačítko vlevo dole (s rychlostí uvedenou v jednotkách Mach).

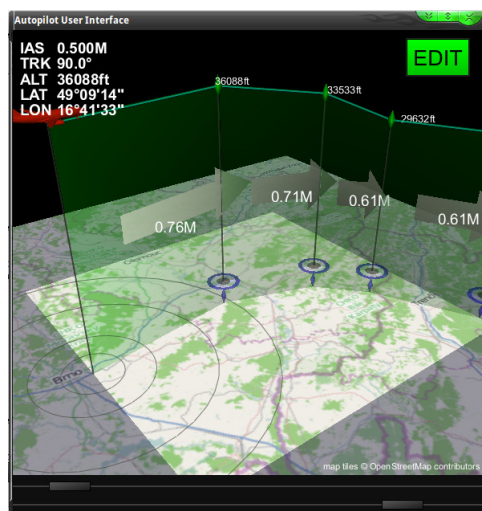
Stávající navigační body je možné v režimu úprav přesouvat po mapě. Je možné měnit rychlost letadla před tímto bodem i výšku s využitím táhla vlevo. Táhllo pak odpovídá vybranému bodu nebo nového bodu, není-li žádný bod vybrán. Podobně probíhalo nastavování rychlosti, které mělo mít vlastní táhllo. Pro práci na tomto konceptu byly zvoleny mapy OpenStreetMap, které byly využity i ve finální implementaci projektu (viz 6.1.3). Velkým omezením tohoto konceptu je nastavení vertikálního profilu letu. Nepřesnosti při jeho nastavení vedou v reálné situaci k vyšší spotřebě paliva a finančním ztrátám provozovatele. Pro přesnější nastavení vertikálního profilu letu je potřeba dvojnásobné zobrazovací plochy nebo využití trojrozměrného rozhraní.

Výsledný konceptuální návrh konverguje k pokročilemu systému automatického řízení letu. Rozvíjí koncept trojrozměrného rozhraní, jelikož zvětšování zobrazovací plochy by znamenalo plýtvání místem. Opět je zde využito vytváření navigačních bodů nad mapou, kterými letadlo prolétá podle zadané rychlosti a výšky. Nově se tak děje v režimu perspektivního zobrazení.



Obr. 5.7: Čtvrtý návrh (částečně implementovaný koncept)

Objekt letounu je zde umístěn na okraj aktivní mapy, která se posouvá s tím, jak se reálný (simulovaný) letoun pohybuje. Jeho rychlost, výšku a směr letu udává nejbližší navigační bod. Výšku bodu lze nastavovat tahem zelené značky nad umístěním bodu v mapě.



Obr. 5.8: Čtvrtý návrh (implementace)

Pohybem nahoru dochází ke zvyšování výšky, kterou letoun daným bodem proletí, až na přednastavené maximum. Pohybem dolů dochází ke snížení výšky až na přednastavené minimum. Tahem šipky na spojnici mezi dvěma body dochází ke změně rychlosti, kterou letoun poletí do druhého bodu. Délka šipky reprezentuje hodnotu rychlosti spolu s číselným údajem. Aktuální hodnoty jsou zobrazeny v jednom z rohů vykreslované oblasti. Opět je zde uvažováno tlačítko úprav s časovačem. Po dosažení posledního přednastaveného bodu zůstává nastavena poslední hodnota skutečného kurzu TRK, skutečné letové výšky ALT a rychlosti IAS. Rychlost je zde udávána v jednotkách Mach.

6 Implementace

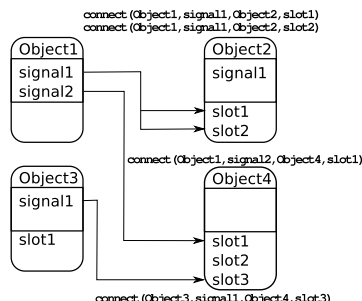
Po prostudování problematiky vzniklo několik návrhů rozhraní (viz předchozí kapitola), které vedly k vytvoření výsledné verze aplikace. Nejedná se o samostatný funkční celek, ale součást systému autopilota, se kterým komunikuje pomocí komunikačního protokolu systému AW-COM. Během svého ladění bylo rozhraní spuštěno na klientském počítači s operačním systémem Xubuntu 12.10. Simulační software vč. autopilota byl spuštěn na serveru s operačním systémem Microsoft Windows 7 64bit.

Pro správný chod aplikace jsou vyžadovány mapové podklady, OBJ model letadla, OBJ model navigačního bodu a inicializační soubor `init.csv`. Jejich významová funkce a umístění je popsána v sekcích 6.1.3, 6.3.3 a 6.3.1.

6.1 Použité technické prostředky

6.1.1 Framework Qt 5.2

Vývoj aplikace probíhal ve vývojovém prostředí Qt Creator 3.0.1 IDE, využívajícím framework Qt 5.2.1. Tento multiplatformní framework umožňuje vývoj pro desktop, mobilní i vestavěné operační systémy. Nabízí obsáhlou dokumentaci s množstvím příkladů a návodů. Primárně byl navržen pro jazyk C++, ale byl portován i do dalších jazyků (Java, Python, C#). Vedle toho nabízí možnost využít jazyk QML, deklarativní jazyk navržený pro rychlý vývoj grafického uživatelského rozhraní [26].



Obr. 6.1: mechanismus slotů a signálů

Tato možnost však uplatněna nebyla. Vývoj probíhal s pomocí rozsáhlé knihovny jazyka C++, kterou framework nabízí. Důvodem pro volbu jazyka C++ byla jeho velká rozšířenost, podpora řady dalších knihoven. Jednou z typických vlastností Qt je využívání mechanismu signálů a slotů. Nastane-li určitá událost (např. stisknutí tlačítka) v rámci jednoho objektu, objekt vyšle signál objektu jinému. Ten jej přijme a aktivuje příslušnou metodu deklarovanou jako slot (např. inkrementace hodnoty). Framework Qt dále nabízí zjednošení síťové komunikace, práce s vlákny i vykreslování. v této souvislosti je vhodné zmínit vykreslovací třídu `QPainter`. Tato třída je sice podporována, ale od jejího používání se ustupuje ve prospěch QML (Qt Quick). Přesto byla využita v metodě načítání mapy (viz sekce 6.3.2). Důvodem byla čitelnost kódu a optimalizovanost metod této třídy [26].

6.1.2 Knihovna OpenGL

Veškerá práce s vykreslováním 3D scény probíhá pomocí knihovny OpenGL verze 2.1.2. Jedná se o otevřenou knihovnu, kterou framework Qt využívá v rámci třídy `QGLWidget`. Třída `QGLWidget` nabízí tři virtuální funkce, které jsou v projektu využity v rámci třídy `GLWidget`, jenž slouží pro vykreslování a interakci uživatele se scénou. Jedná se o následující metody¹:

`paintGL()` – Slouží pro renderování scény. Kromě prvního zavolání po vytvoření ovládacího prvku (angl. *widget*) je volána pomocí metody `updateGL()`. To se děje v případě překreslování scény.

`resizeGL()` – Nastavuje úhel pohled na scenu, projekci, apod. Kromě prvního zavolání po vytvoření ovládacího prvku je volána po každé změně velikosti ovládacího prvku.

`initializeGL()` – Vytváří obsah OpenGL, který se bude renderovat. Je volána jednou před voláním `paintGL()` a `resizeGL()`.

Příkazy OpenGL jsou využívány pro přímé vykreslování některých primitiv. Příkladem je vykreslování mapy, mapové značky navigačního bodu nebo spojnice dvou navazujících navigačních bodů (viz 6.4.3). Složitější objekty jsou vykreslovány pomocí třídy `objloader` po načtení modelu ze souboru OBJ (viz 6.3.3).

6.1.3 Mapové podklady OpenStreetMaps

Pro zobrazení území, nad nímž letadlo se letadlo pohybuje, byly zvoleny mapové podklady služby OpenStreetMaps [23]. Tyto podklady jsou volně² ke stažení s podmínkou, že nebude docházet k nadměrnému stahování ze serveru služby. Za nadměrné stahování se považuje rovněž stahování mapových podkladů s přiblížením velikosti 17 a vyšším. Druhou podmínkou je zřetelné uvedení původu podkladů v aplikaci³ a její dokumentaci.

$$http : // [abc].tile.openstreetmap.org / zoom / x / y .png \quad (6.1)$$

Mapové podklady jsou tvořeny dlaždicemi. Každá dlaždice je uložena jako rastrový obrázek formátu PNG s rozlišením 256×256 px. Pro účely aplikace rozhraní autopilota byly zvoleny dlaždice na výchozím serveru služby OpenStreetMaps. Cestu souboru dlaždice na serveru zachycuje pseudo – rovnice 6.1, kde `tile.openstreetmap.org` je název serverové domény s nepovinnou subdoménou a,b nebo c.

¹<http://qt-project.org/doc/qt-5/qglwidget.html>

²<http://www.openstreetmap.org/copyright>

³ve tvaru „© OpenStreetMap contributors“.

Umístění mapových podkladů na jiných serverech služby se může lišit. Stejně tak může být použit jiný formát pro uložení jednotlivých dlaždic (např. JPG). Neměnná zůstává cesta od výchozího mapového adresáře, kde:

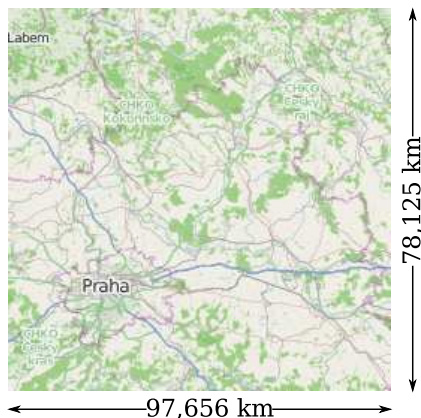
- Veličina **zoom** označuje velikost přiblížení dlaždice. Nabývá celočíselných hodnot od 0 po 18 (největší přiblížení), přičemž toto rozmezí se liší podle zvoleného serveru.
- Veličina **x** označuje souřadnici dlaždice ve smyslu zeměpisné délky (od 180° západní délky po 180° východní délky). Nabývá hodnot od 0 po $2^{\text{zoom}} - 1$. Její výpočet zachycuje následující rovnice.

$$x = \left\lfloor \frac{\text{lon} + 180}{360} \cdot 2^{\text{zoom}} \right\rfloor \quad (6.2)$$

- Veličina **y** označuje souřadnice dlaždice ve smyslu zeměpisné šířky (od 85,0511° jižní šířky po 85,0511° severní šířky). Nabývá hodnot od 0 po $2^{\text{zoom}} - 1$ ve smyslu Mercatorového zobrazení. Její výpočet zachycuje následující rovnice.

$$y = \left\lfloor \left(1 - \frac{\ln \left(\tan \left(\text{lat} \cdot \frac{\pi}{180} \right) + \frac{1}{\cos \left(\text{lat} \cdot \frac{\pi}{180} \right)} \right)}{\pi} \right) \cdot 2^{\text{zoom} - 1} \right\rfloor \quad (6.3)$$

Velikost přiblížení byla experimentálně stanovena jako velikost 8. Dlaždice takové velikosti zobrazuje dostatečné, ale nikoliv nadbytečné množství geografických údajů. Zachyceny jsou státní hranice, řeky, krajská města, větší lesní porosty nebo pozemní dopravní cesty. Všechny 256 × 256 souborů dlaždic zabírá 115,6 MB.



Obr. 6.2: Dlaždice OpenStreetMaps přiblížení 8

Obvod Země činí zhruba 40 075 km na rovníku. Obvod rovnoběžky se snižuje směrem k pólu a zvyšuje směrem k rovníku, což je reflektováno na fyzické vzdálenosti reprezentované šířkou dlaždice. Pro zeměpisnou šířku Prahy⁴ má obvod rovnoběžky hodnotu 25 000 km. Vyděleno počtem dlaždic vychází, že dlaždice má fyzickou šířku 97,656 km.

⁴<<https://help.openstreetmap.org/questions/1069/how-to-calculate-the-size-of-a-map-tile-on-the-ground>>

6.2 Laboratoř 737-800 NG

V laboratoři je kokpit 737 NG. Je tam možno řešit projekty v oblasti počítačové grafiky, vestavěných systémů, sítí, bezpečnosti. Nachází se tam cylindrický projekční systém o průměru 5 metrů a výšce 3 metrů. Pro nasvícení projekční plochy je použito 3 projektorů. Ke komunikaci s displeji a simulaci je použit protokol AWCCom.



Obr. 6.3: Pohled do simulátoru, přejato [20]

6.2.1 Simulační model

Simulační model použitý v laboratoři vychází z dopravního letadla Boeing 737-800 NG, jehož charakteristika popisuje tabulka 6.1. Aplikace rozhraní autopilota byla testována i na simulaci jiných letadel (podle dat, které odesílal server). Její chování (rozměry mapy, použitá měřítko) však odpovídají simulačnímu modelu Boeingu 737 NG. Údaje uvedené v tabulce 6.1 jsou převzaty z [2, 19].

Tabulka 6.1: Charakteristika Boeing 737-800 NG

pasažérů	189
motory	2 × CFMI CFM56-7
tah motoru	121,4 kN (27 300 lb)
dolet	5 765 km (3 115 námořních mil)
max. vzlet. hmotnost	79 010 kg
typická rychlost	0,785 M (ve výšce 35 000 ft)
max. let. výška	41 000 ft (12 497 m)
rozpětí křídel	35,8 m vč. křidélek
celková délka	39,5 m
výška zadní části	12,5 m

6.2.2 Popis komunikačního protokolu

Komunikační protokol AW-COM umožňuje zasílat velké množství letových veličin mezi simulátorem (běžícím na serveru) a klientskou aplikací (nemusí běžet přímo na serveru). Spojení je realizováno pomocí vrstvy TCP. Aplikace autopilota k tomuto účelu využívá metod třídy `QTcpSocket`. Pro systémovou komunikaci zde slouží samostatné vlákno, implementované třídou `inThread` (dědí ze třídy `QThread`). Samotná komunikace probíhá v těle metody `getData()`.

Tabulka 6.2: Formát příchozí zprávy [14]

Část zprávy	Jednotka	Popis
DATA		Hlavička zprávy, která určuje posílání letových dat ze serveru.
<i>:veličina :(double *.6)</i>		Veličina reprezentuje název hodnoty, hodnota veličiny je zapsána vždy s přesností na 6 desetinných míst. Pokud se daná veličina vyskytne ve zprávě dvakrát, je brán první výskyt zleva. Aplikace využívá veličiny, které můžete vidět níže.
ALT	ft	Výška
LAT	°	Zeměpisná šířka
LON	°	Zeměpisná délka
TT_	°	Kurz skutečný
IAS	m/s	Indikovaná rychlost letu
VS_	m/s	Vertikální rychlost
:END		Ukončovací sekvence na konci zprávy.

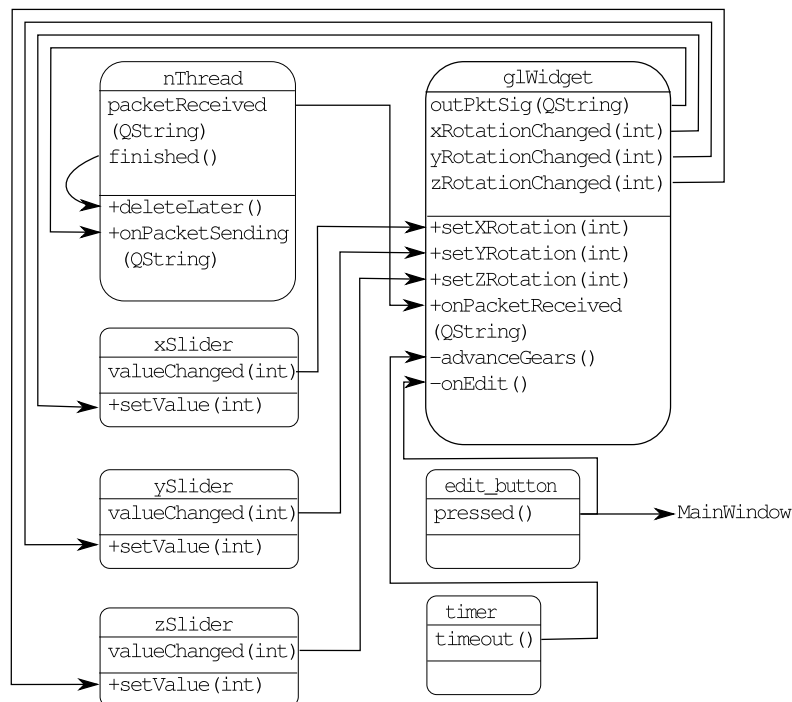
Metoda `getData()` je volána každých 50 ms. Klient se spojí se serverem dané IP adresy na zadaném portu. Po vytvoření spojení je poslána zpráva `"GETDATA\n"`. V případně kladné odezvy server odpoví zprávou, jejímž obsahem jsou aktuální letová data. Formát příchozí zprávy je popsán v tabulce 6.2. Po obdržení příchozí zprávy se klient odpojí a počká 50 ms. Následně na server pošle zprávu, která obsahuje nové hodnoty letových dat. Ty jsou výstupem aplikace rozhraní autopilota. Tvar odchozí zprávy popisuje tabulka 6.3.

Tabulka 6.3: Formát odchozí zprávy [14]

Část zprávy	Jednotka	Popis
SET: <i>Seznam.hodnot</i> :END		Formát zprávy pro zasílání letových dat na server. Seznam.hodnot tvoří textový řetězec hodnot ve smyslu tabulky 6.2. Aplikace využívá následující veličiny.
APALT	ft	Nová výška
APTT_	°	Nový kurz skutečný
APIAS	m/s	Nová indikovaná rychlost letu
APVS_	m/s	Nová vertikální rychlost

6.3 Inicializace aplikace

Aplikaci uživatelského rozhraní systému autopilota lze rozdělit do tří současně běžících částí. Jednou z nich je instance třídy `MainWindow`, která řídí vykreslování jednotlivých ovládacích prvků uvnitř jednoho okna. Druhou částí je vlákno `nThread` třídy `inThread`, jehož činnost je popsána v sekci 6.2.2. Třetí částí je ovládací prvek `glWidget` třídy `GLWidget`, který zapouzdřuje samotné trojrozměrné uživatelské rozhraní.



Obr. 6.4: Propojení jednotlivých částí signály a sloty

Nejdřív však konstruktor třídy `MainWindow` vytvoří instanci třídy `InitData`, čímž načte inicializační data (viz 6.3.1). Na jejich základě vytvoří vlákno `nThread` a ovládací prvky vč. `glWidget`. Všechny tyto části následně propojí mechanismem signálů a slotů (viz sekce 6.1.1). Schéma tohoto propojení je znázorněno na obr. 6.4.

6.3.1 Načítání inicializačního souboru

Načítání dat inicializačního souboru provádí instance třídy `InitData` v rámci svého konstruktoru. Požadovaná data jsou obsažena v souboru `init.csv`, který je umístěn přímo v pracovním adresáři aplikace. Jedná se o textový soubor formátu CSV. Každý načítaný řádek tohoto souboru má tvar: `<proměnná>, <hodnota>\n`

Veličina `<proměnná>` reprezentuje určitou proměnnou aplikace. Za znakem čárky následuje `<hodnota>` této veličiny. Použité proměnné popisuje tabulka 6.4. Příkladem může být řádek `aircraft_model, "/models/Boeing737/Boeing737.obj"` načítající 3D model letadla. Ostatní řádky jsou ignorovány. Pokud nějaká proměnná v souboru chybí, je nastavena na nenulovou výchozí hodnotu.

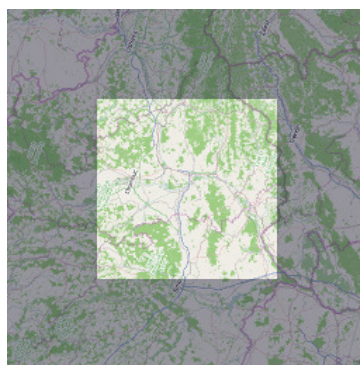
Inicializační soubor obsahuje data pro spojení se simulačním serverem, data pro výšku a rychlost letu, velikost přiblížení mapy (viz sekce 6.1.3) a umístění použitých 3D modelů formátu Wavefront .obj. Ty jsou použity pro 3D model letadla a 3D model značky navigačního bodu (viz 6.3.3).

Tabulka 6.4: Formát inicializačního souboru

Proměnná	Typ	Popis
tcp_ip	QString	Adresa simulačního serveru
tcp_port	int	Port běžící simulace
max_alt	int	Maximální výška ve stopách
min_alt	int	Minimální výška ve stopách
max_speed	float	Maximální rychlost, Mach
min_speed	float	Minimální rychlost, Mach
aircraft_model	QString	Relativní cesta souboru 3D modelu letadla (viz 6.3.3)
aircraft_model_ratio	float	Zvětšení 3D modelu letadla
beacon_model	QString	Soubor 3D modelu navigační značky (viz 6.3.3)
beacon_model_ratio	float	Zvětšení 3D modelu značky
zoom	int	Přiblížení mapy

6.3.2 Načítání mapových podkladů

K načítání mapových podkladů slouží metoda `getGround()` třídy `GLWidget`. Tato metoda je volána na začátku metody `paintGL()` (viz 6.1.2), jelikož vykreslovaná mapa se v průběhu času mění. Nejdříve metoda `getGround()` vypočítá pozici aktuální dlaždice (podle rovnic 6.1 a 6.3). Využívá k tomu instanční proměnné `latitude`, `longitude` aktuální zeměpisné šířky a délky.



Obr. 6.5: Načtená mapa OSM

Podobným způsobem je následně vypočítána průměrná velikost fyzické plochy, kterou zabírá jeden pixel této dlaždice. Výstup je uložen do instančních proměnných `px_x` a `px_y`. Uložená desetinná hodnota `px_x` reprezentuje rozdíl zeměpisné délky mezi 2 sousedícími pixely (`px_y` reprezentuje rozdíl zeměpisné šířky). Až po tomto výpočtu dochází k načítání mapových dlaždic. Ty jsou načítány do proměnné `result`, objektu třídy `QPixmap`.

Ta zde plní roli vyrovnávací paměti, resp. matice, do které je načtena aktuální dlaždice a její zdvojené osmi-okolí⁵. Pro úpravy proměnné `result` je využita třída `QPainter`, která ještě před načtením jednotlivých dlaždic otočí celou maticí kolem jejího středu. Jednotlivé dlaždice jsou potom umístěny na odpovídající pozice kolem aktuální dlaždice. Velikost otočení určuje instanční proměnná `heading`, jenž reprezentuje aktuální skutečný kurz TRK.

Následně dochází k vyříznutí oblasti kolem středu matice o rozměrech dlaždice. Ještě předtím ale byl střed oblasti posunut tak, aby střed dolního okraje oblasti odpovídal aktuální pozici letadla. Tato oblast je uložena do instanční proměnné `tile_buf` jakožto aktivní oblasti mapy, na které je následně možné vytvářet navigační body (viz 6.4.3). Její okolí je po ztmavení uloženo do instanční proměnné `n_tiles`. Skutečná velikost okolí je 512×512 px. Na obrázku 6.5 je zachycen výstup pro kurz 70° a souřadnice brněnského letiště Tuřany.

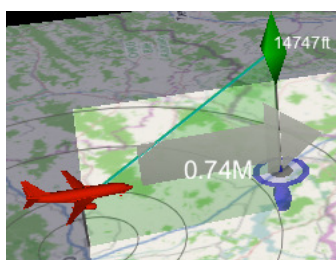
6.3.3 Načítání modelu OBJ

Formát Wavefront `.obj` je jeden z nejjednodušších formátů pro uložení 3D modelů formou textového souboru. Aplikace proto využívá vlastní třídu `objloader` pro načítání těchto modelů [27]. Základní syntaxe formátu je následující:

Tabulka 6.5: Formát souboru `.obj`

#		Komentář
<code>v</code>	<code>x y z</code>	Souřadnice vertexu (bodu v prostoru)
<code>vn</code>	<code>nx ny nz</code>	Souřadnice normálového vektoru bodu
<code>f</code>	<code>[v]//[vn] [v]//[vn] [v]//[vn]</code>	Trojúhelník s odkazy na identifikační číslo řádku vertexu <code>v</code> a identifikační číslo odpovídajícího normálového vektoru <code>vn</code>

V celé aplikaci jsou načítány dva modely. Jedním je model reprezentující letadlo⁶, druhým model značky navigačního bodu. Umístění těchto modelů lze specifikovat pomocí inicializačního souboru `init.csv` (viz 6.3.1). Jejich zobrazení zachycuje 6.6. Značka navigačního bodu je vykreslena dvakrát (zeleně pro nastavení výšky, modře pro nastavení pozice v mapě). Soustředné kružnice pod objektem letadla reprezentují přibližné pozemní vzdálenosti. Jejich poloměr počínaje vnitřní kružnicí činí 12,5 km, 25 km, 50 km, 75 km.



Obr. 6.6: Model letadla a navigačního bodu

Při testování načítání 3D modelu byly tyto modely nahrazovány jen úpravou dvou řádků v inicializačním souboru. Pro změnu použitých modelů tudíž není nutné kompilovat ce-

⁵sousední dlaždice a sousední dlaždice všech těchto dlaždic

⁶model byl převzat z <http://tf3dm.com/3d-model/boeing-737-25104.html>

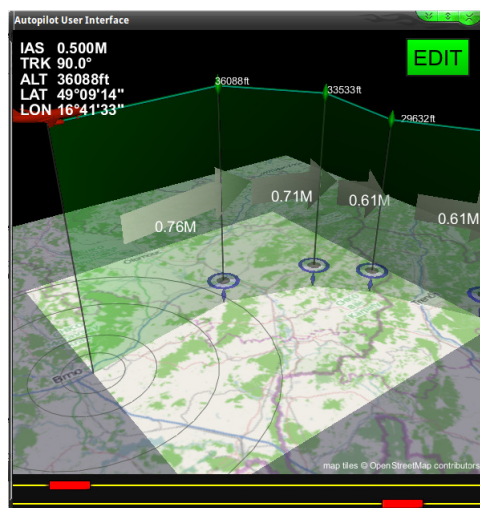
lou aplikaci. Pro samotné načítání modelu slouží metoda `load(char *)` třídy `objloader`, jejímž argumentem je cesta k načítanému souboru (relativní nebo absolutní). Po zavolání této metody je soubor načítán řádek po řádku. Metoda `load(char *)` volá metodu `parseLine(char *)`, které předá načtený řádek k dalšímu zpracování. Varianty dalšího zpracování jsou následující:

- **parseVertex** – Pro řádek začínající `v` je volána metoda `parseVertex(char *)`. Načtené souřadnice bodu jsou uloženy do vektoru `vertices`, jenž obsahuje jednotlivé body modelu.
- **parseNormal** – Pro řádek začínající `vn` je volána metoda `parseNormal(char *)`. Načtený normálový vektor je uložen do vektoru `normals`, jenž obsahuje jednotlivé normálové vektory modelu.
- **parseFace** – Pro řádek začínající `f` je volána metoda `parseFace(char *)`. Načtený trojúhelník je uložen do vektoru `faces`, jenž obsahuje strany tvořící model. Uvádění normálových vektorů není povinné. Načítání textury implementováno není, protože ji aplikace nevyužívá.

Pro vykreslování modelu slouží metoda `draw()`, která pro každou stranu modelu vykreslí trojici bodů. Každý bod je vykreslen pomocí příkazu `glNormal3f()` a `glVertex3f()` na základě dat uložených ve vektorech `faces`, `normals` a `vertices`.

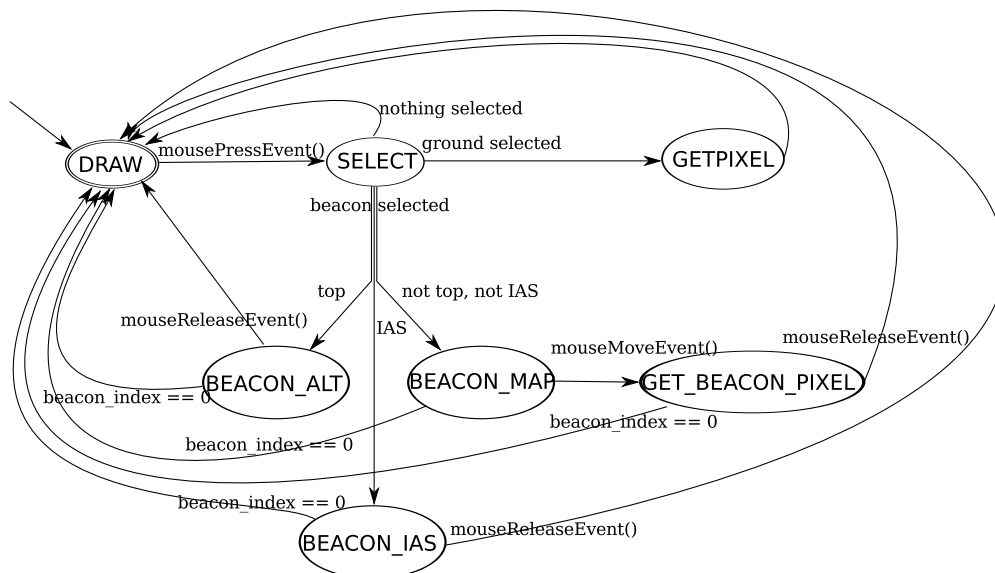
6.4 Zobrazení scény

Pro zobrazení scény slouží několik metod třídy `GLWidget`. Průběh jejího běhu záleží na aktuálním stavu instance této třídy. Ve výchozím stavu `DRAW` dochází pouze k vykreslování scény a jejímu otáčení uživatelem. K tomuto otáčení slouží dvojice táhel (na obrázku 6.7 zvýrazněny červeně). Vrchní táhlo otáčí scénou podle osy `x`, dolní podle svislé osy `y`. Pohyb kurzoru bez označeného objektu scény otáčí scénou podle osy `y`.



Obr. 6.7: Tlačítko `EDIT` je aktivní

V inicializaci scény `initializeGL()` je vytvořeno několik objektů pro vykreslování (letadlo, aktivní oblast mapy, okolí aktivní oblasti) a pro výběr ze scény (letadlo, aktivní oblast mapy). Dalšími objekty jsou navigační body⁷, které jsou vytvářeny až po inicializaci. K samotnému vykreslování slouží metoda `paintNormal`, která je spouštěna v metodě `paintGL()`. Metoda `paintNormal` vykresluje mapu pomocí metody `makeGround()` (volané dvakrát – jednou pro aktivní oblast, jednou pro okolí), letadlo nad okrajem mapy metodou `drawCraft()` a konečně navigační body pomocí metody `paintBeacons()`.



Obr. 6.8: Základní stavy aplikace

Pro výběr objektů scény a možnost zadávat či upravovat navigační body je třeba aktivovat tlačítko **EDIT**. To je v neaktivním stavu šedé, v aktivním zelené. Při aktivním tlačítku **EDIT** může instance třídy `GLWidget` nabývat následujících stavů (režimů):

- **DRAW** – Bude probíhat běžné vykreslování scény.
- **SELECT** – Je vybírán objekt scény.
- **GETPIXEL** – Byla vybrána země, přidán nový navigační bod.
- **BEACON_ALT** – Byla vybrána značka nav. bodu pro úpravu výšky, do nav. bodu je uložena nová hodnota výšky.
- **BEACON_MAP** – Byla vybrána značka nav. bodu pro úpravu umístění bodu na mapě.
- **GET_BEACON_PIXEL** – Byly získány nové souřadnice pro umístění bodu.
- **BEACON_IAS** – Byla vybrána značka rychlosti do nav. bodu, do nav. bodu je uložena nová hodnota rychlosti.

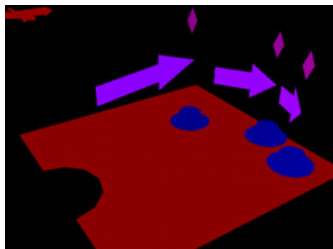
6.4.1 Výběr objektu scény

Pro výběr objektu scény musí aplikace přejít do stavu **SELECT**. Děje se tak stisknutím vybraného bodu obrazovky. Podle pozice tohoto bodu je vybrán objekt, který je umístěn nejbližší pomyslné kameře scény a je vykreslen přes tento bod. Tento výběr probíhá s využitím

⁷Nejedná se o bod v pravém smyslu, ale objekt na tento bod odkazující

algoritmu výběru na základě unikátní barvy [25]. Jeho možnou alternativou je algoritmus výběru s využitím trasování paprsku, který je však výpočetně náročnější.

Podstatou algoritmu výběr na základě unikátní barvy je přiřazení celočíselného identifikátoru každému objektu scény. Tento identifikátor je následně přepočítán do trojice 8bitových barevných kanálů RGB. Tímto způsobem lze adresovat až $256^3 - 1$ objektů (pozadí a ignorované objekty mají barevnou hodnotu RGB(0,0,0)). Každý objekt je vykreslen jednolitou barvou tohoto identifikátoru (bez stínování či textur). Velikost těchto objektů se od normálního zobrazování může lišit. To je případem složeného objektu navigačního bodu, jehož jednotlivé části jsou pro snadnější výběr vykreslovány zvětšeně.



Obr. 6.9: Výběr pomocí barevného identifikátoru (snížený barevný rozsah)

Pro výběr objektu jsou vykreslovány tyto objekty: Objekt letadla, objekt aktivní oblasti mapy, vektor objektů navigačních bodů. Každý navigační bod sestává z trojice podobjektů (šipka rychlosti, mapová značka a vertikální značka), které jsou vykreslovány zvětšeně. Pod letadlem je navíc vykreslován kruh s barvou pozadí a poloměrem 12,5 km. Ten zajistí, aby nedocházelo k vytváření navigačních bodů příliš blízko letícímu letadlu. Hodnoty jednotlivých objektů zachycuje tabulka níže⁸.

Tabulka 6.6: Barevné identifikátory objektů

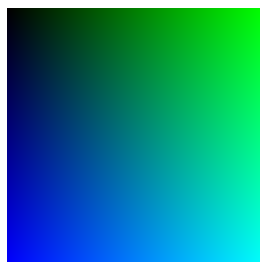
Objekt scény	ID	Barva
Pozadí, ignorováno	0	RGB(0,0,0)
Mapa	-	RGB(1,0,0)
Letadlo	-	RGB(2,0,0)
Navigační bod:	id	R = 0; G = id div 255; B = id mod 255;
Šipka rychlosti	id	RGB(B,G,255)
Mapová značka	id	RGB(R,G,B)
Vertikální značka	id	RGB(B,G,B)
Šipka rychlosti	6	RGB(6,0,255)
Mapová značka	6	RGB(0,0,6)
Vertikální značka	6	RGB(6,0,6)

Pro výběr objektu slouží metoda `paintPicking()`, jenž je volána v těle vykreslovací metody `paintGL()`. Po jejím skončení dochází k vymazání vyrovnávací paměti pro barvu a hloubku. Scéna je poté vykreslena normálním způsobem. Není potřeba dodatečná vyrovnávací paměť. Pokud byla vybrána aktivní oblast mapy, aplikace přechází do stavu `GETPIXEL`.

⁸id označuje pořadové číslo navigačního bodu ve vektoru navigačních bodů počínaje jedničkou

6.4.2 Výběr souřadnic mapy

Podobný algoritmus, jako v předchozí sekci, slouží k výběru konkrétního bodu mapy. Děje se tak v případě přidávání nového navigačního bodu (stav `GETPIXEL`) nebo posouvání po mapě pomocí mapové značky tohoto bodu (stav `GET_BEACON_PIXEL`). Aktivní oblast mapy je zde pokryta gradientem, v němž každý pixel má barvu odpovídající jeho pozici v aktivní oblasti ve tvaru `RGB(1, <číslo_řádku>, <číslo_sloupce>)`. Výběrem tohoto pixelu dochází na základě informace uložené barvou k výpočtu zeměpisné šířky, délky a nového umístění navigačního bodu.



Obr. 6.10: Gradient sloužící k výběru bodu mapy

Aktivní oblast má rozměry 255×255 px, což odpovídá rozsahu dvou barevných kanálů. Po prvním vytvoření je gradient uložen do pracovního adresáře aplikace pro opakované použití. Po přesunu nebo vytvoření navigačního bodu dochází k vymazání vyrovnávací paměti pro barvu a hloubku. Scéna je poté vykreslena normálním způsobem.

6.4.3 Navigační body

Pro vykreslování navigačního bodu a uložení relevantních informací slouží třída `Beacon`. Každá instance této třídy obsahuje údaje o zeměpisné šířce a délce navigačního bodu, jeho výšce a rychlosti letu před dosažením tohoto bodu. Tato třída popisuje objekt scény, jenž je složený ze tří částí:

- Mapová značka tvořená modrým terčíkem a modelem značky navigačního bodu. Slouží k přesouvání bodu po mapě ve smyslu (viz 6.4.2).
- Vertikální značka tvořená zeleným modelem značky navigačního bodu. Slouží ke změně výšky daného bodu.
- Značka rychlosti tvořená šipkou před objektem navigačního bodu. Její délka reprezentuje velikost rychlosti letu před navigačním bodem.

Hodnoty pro výšku a rychlost nového bodu odpovídají výšce a rychlosti bodu předchozího. Pokud žádný předchozí bod neexistuje, je vybrána aktuální výška a rychlost letu. Pro řízení letu je využit první navigační bod obsažený ve vektoru navigačních bodů. Systému řízení letu jsou odeslány informace o výšce tohoto bodu, jím stanovené rychlosti a skutečný kurz vypočítaný na základě zeměpisných souřadnic bodu. Po dosažení minimální vzdálenosti mezi bodem a letadlem je navigační bod odebrán (řízení letu je předáno dalšímu navigačnímu bodu).

6.5 Testování aplikace

V počátečních fázích testování se aplikace běžící v prostředí systému Xubuntu 12.10 připojovala k serveru, na kterém běžela simulace. Testování výsledné aplikace probíhalo v prostředí leteckého simulátoru laboratoře 737-800 NG (viz sekce 6.2). Rozhraní bylo zobrazeno na druhé obrazovce zleva. Navíc jej bylo možné ovládat pomocí rezistivní dotykové obrazovky zařízení Dell XT2, hybridního zařízení kombinujícího tablet PC a notebook⁹. Umístěno bylo na středový sloupek podle doporučení uvedených v kapitole 4.4.



Obr. 6.11: Snímky z testování aplikace [20]

Řídicí počítač simulátoru využívá operační systém Windows 7 Professional 64-bit, Dell XT2 využívá Windows 7 Professional 32-bit. Aplikace byla proto přeložena v tomto operačním systému (na notebooku Asus M51SN). Přeložené zdrojové soubory stačilo pouze zkopírovat (spolu se soubory potřebných knihoven).

6.6 Možné rozšíření

Testování na simulátoru prokázalo použitelnost aplikace. Je zde však prostor pro další vývoj. Mezi možné rozšíření patří:

1. V režimu editace vypnout otáčení mapy. Ukázalo se, že při použití rezistivní obrazovky může být odezva nižší, než jaký byl předpoklad. Odstranění nechtěného otáčení mapy zvýší uživatelskou přívětivost rozhraní.
2. Přidat tlačítko přidávání navigačních bodů. Navigační body by v režimu přidávání nebylo možné editovat.
3. Přidat tlačítko odebírání navigačních bodů. Navigační body by v režimu odebírání nebylo možné editovat.
4. Ještě zvětšit vertikální značku navigačního bodu, co se výběru objektu scény týče (viz sekce 6.4.1).
5. Načítání navigačních bodů ze serveru nebo souboru.

⁹specifikace viz <http://www.pcworld.com/product/54137/dell-latitude-xt2.html>

7 Závěr

Cílem této práce bylo navržení rozhraní systému autopilota, které je intuitivní a přehledné. Vývoj probíhal ve spolupráci s výzkumnou skupinou AeroWorks, která se zabývá oblastí letecké simulace na FIT VUT. Návrh vycházel z existujících systémů rozhraní autopilota s využitím teoretického základu popisujícího chování letadla za letu. Rovněž zde byly použity poznatky z nasazování technologie dotykových obrazovek ve velkých dopravních letadlech, pro něž je rozhraní určeno.

Vývoj probíhal s ohledem na další možné rozšíření a možnost konfigurovat některé části aplikace bez zásahů do kódu. Původní konzervativní návrh byl kompletně přepracován do podoby trojrozměrného rozhraní pokročilého automatického řízení letu. Přesto se zde nabízí prostor pro další vylepšování a rozšiřování.

Cenné zkušenosti získané vývojem této aplikace bude možné využít při práci na dalších projektech této oblasti. Předpokládané další nasazení aplikace je využití v prostředí leteckého simulátoru laboratoře 737-800 NG

Literatura

- [1] Adams, C.: Human Factors In Avionics Design, *Avionics Today*. [online]. http://www.aviationtoday.com/av/commercial/Human-Factors-in-Avionics-Design_80444.html, 1. 11. 2013, [cit. 4. 2. 2014].
- [2] Brady, C.: Automatics, *Boeing 737 Technical Site*. [online]. <http://www.b737.org.uk/glareshield.htm>, aktualizováno 24. 10. 2013, [cit. 4. 2. 2014].
- [3] Cavcar, M.: *Airspeed Definitions*. Eskisehir, Turecko: Anadolu University. [online] <http://home.anadolu.edu.tr/~mcavcar/common/Airspeeds.pdf>.
- [4] Cook, M.: *Flight dynamics principles, A linear systems approach to aircraft stability and control*. 2. vydání, Elsevier, 2007, ISBN 9780750669276.
- [5] Croft, J.: Honeywell Nearing Launch Of Touchscreen-Enabled Avionics, *Aviation Week*. [online]. http://www.aviationweek.com/Article.aspx?id=/article-xml/avd_10_21_2013_p03-01-628244.xml, 21. 10. 2013, [cit. 4. 2. 2014].
- [6] Daněk, M.: *Aerodynamika a mechanika letu*. 1. přepr. vyd., Košice: VVLŠ, 1990, 289 s.
- [7] Drahota, J.: Autopilot. *flyMag*, [online]. <http://flymag.cz/article.php?id=9089>, 1. 11. 2013, [cit. 4. 2. 2014].
- [8] Federal Aviation Administration: Airplane Flying Handbook, FAA-H-8083-3B. http://www.faa.gov/regulations_policies/handbooks_manuals/aircraft/airplane_handbook/media/FAA-H-8083-3B.pdf, 2004.
- [9] Federal Aviation Administration: Pilot's Handbook of Aeronautical Knowledge, FAA-H-8083-25A. http://www.faa.gov/regulations_policies/handbooks_manuals/aviation/pilot_handbook/media/FAA-H-8083-25A.pdf, 2008.
- [10] Gangadhar Kallur: History of Autopilot, [online]. <http://gangmadhu.blogspot.cz/2010/09/history-of-autopilot.html>, 12. 9. 2010, [cit. 4. 2. 2014].
- [11] Harris, W.: How Autopilot Works. *How Stuff Works*, [online] [cit. 4. 2. 2014]. <http://science.howstuffworks.com/transport/flight/modern/autopilot.htm>.
- [12] Honig, Z.: Boeing 787 review: ANA's Dreamliner flies across Japan, we join for the ride, *Engadget* [online]. <http://www.engadget.com/2011/12/16/boeing-787-review-ana-dreamliner-flies-across-japan-we-join/>, 16. 12. 2011, [cit. 4. 2. 2014].
- [13] Jíra, R.: *Aerodynamika a mechanika letu pro plachtaře*. 1. vydání, Naše vojsko, 1960.
- [14] Karol, R.: *Vizualizace vzdušného prostoru ve 3D*. Diplomová práce, MUNI-FI v Brně, Brno, 2012, [online] http://is.muni.cz/th/389970/fi_m/.

- [15] National Transportation Safety Board: NTSB Accident Report NYC99MA178. [online] http://www.nts.gov/aviationquery/brief.aspx?ev_id=20001212X19354&key=1, 7. 6. 2000.
- [16] Searle, L.: The Bombsight War: Norden vs. Sperry. *IEEE Spectrum*, září 1989: s. 60–64, [online] [cit. 4. 2. 2014]. http://www.thevaluesell.com/images/LSearle_bombsight.pdf.
- [17] Talay, T.: SP-367 Introduction to the Aerodynamics of Flight [online]. <http://history.nasa.gov/SP-367/contents.htm>, 1975, [cit. 4. 2. 2014].
- [18] Woodrow Bellamy: NBAA 2013: Touchscreen Cockpit Systems and Next Generation Business Aircraft, *Avionics Today*. [online]. http://www.aviationtoday.com/av/business-and-general-aviation/NBAA-2013-Touchscreen-Cockpit-Systems-and-Next-Generation-Business-Aircraft_80495.html, 24. 10. 2013, [cit. 4. 2. 2014].
- [19] WWW stránky: 737-800 Technical Characteristics, [online] [cit. 17. 5. 2014]. http://www.boeing.com/boeing/commercial/737family/pf/pf_800tech.page.
- [20] WWW stránky: Aeroworks, FIT VUT, [online] [cit. 7. 5. 2014]. <http://merlin.fit.vutbr.cz/AeroWorks/>.
- [21] WWW stránky: How does a speedometer in an airplane work? *How Stuff Works*, [online]. <http://science.howstuffworks.com/transport/flight/modern/question597.htm>, [cit. 4. 2. 2014].
- [22] WWW stránky: Triden Series Autopilot, [online] [cit. 4. 2. 2014]. <http://www.centuryflight.com/products/systems/triden-series-autopilot.html>.
- [23] WWW stránky: Wiki stránky OpenStreetMaps, [online] [cit. 7. 5. 2014]. <http://wiki.openstreetmap.org/>, 11. 5. 2014.
- [24] WWW stránky: Lawrence Sperry: Autopilot Inventor and Aviation Innovator. *HISTORYnet.com*, [online]. <http://www.historynet.com/lawrence-sperry-autopilot-inventor-and-aviation-innovator.htm>, 12. 6. 2006, [cit. 4. 2. 2014].
- [25] WWW stránky: OpenGL Selection Using Unique Color IDs, *The Game Programming Wiki*. [online]. http://content.gpwiki.org/index.php/OpenGL_Selection_Using_Unique_Color_IDS, 12. února 2012, [cit. 17. 5. 2014].
- [26] WWW stránky: dokumentace Qt 5.2, Qt Project. Digia plc., [online] [cit. 7. 5. 2014]. <https://qt-project.org/doc/qt-5/index.html>, 2013.
- [27] WWW stránky: Waples J.: WaveFront Loader Tutorial, *JamWaffles*. [online]. <http://jamwaffles.co.uk/tutorials/opengl/wavefrontloader>, 2014, [cit. 17. 5. 2014].

- [28] WWW stránky: George the Autopilot. *Historic Wings*, [online].
<http://fly.historicwings.com/2012/08/george-the-autopilot/>, 30.8.2012,
[cit. 4.2.2014].

A Obsah CD

Příložené CD obsahuje:

- **bin-Ubuntu.zip** – přeložená aplikace, na systému Xubuntu 12.10, bez mapových podkladů OSM
- **bin-Windows.zip** – přeložená aplikace pro Windows 7 s potřebnými knihovnami, bez mapových podkladů OSM
- **osm.zip** – OpenStreetMaps přiblížení 8 (zoom = 8) pro vložení do složky OSM v pracovním adresáři přeložené aplikace
- **src.zip** – zdrojové kódy
- **dokumentace.pdf** – dokumentace aplikace
- **ibp-xzubat02.pdf** – technická zpráva bakalářské práce
- **ibp-xzubat02.zip** – zdroj. kódy technické zprávy bakalářské práce (soubory L^AT_EX, obrázky)

B Konfigurační soubor

Vzorový obsah textového konfiguračního souboru `init.csv` vypadá následovně:

```
tcp_ip, "147.229.13.183"  
tcp_port, 12321  
  
max_alt, 11200  
min_alt, 600  
  
max_speed, 0.8  
min_speed, 0.2  
  
http://tf3dm.com/3d-model/boeing-737-25104.html  
aircraft_model, "./models/Boeing737/Boeing737.obj"  
aircraft_model_ratio, 0.04  
  
beacon_model, "./models/dual_pyramid.obj"  
beacon_model_ratio, 0.7  
  
zoom, 8
```