



Zadání bakalářské práce

Název:	Dashboard pro online závodní hru
Student:	Ing. Marek Čermák
Vedoucí:	Ing. Pavel Štěpán
Studijní program:	Informatika
Obor / specializace:	Webové a softwarové inženýrství, zaměření Softwarové inženýrství
Katedra:	Katedra softwarového inženýrství
Platnost zadání:	do konce letního semestru 2024/2025

Pokyny pro vypracování

Hra Asseto Corsa Competition je jedna ze závodních her, ve které se u nás pořádá množství online závodů, které zahrnují i vytrvalostní závody na 24 h se střídáním jezdců. V takových závodech je pro úspěch mimo jiné potřeba v reálném čase sledovat množství telemetrických dat o voze. Zobrazení těchto dat v samotné hře je ale omezené anebo není poskytováno v dostatečné kvalitě. Pomocí API hry je pak možné potřebná data získat jinými programy, čehož by se využilo pro vytvoření nové aplikace, která by již nároky na jejich zobrazení splňovala.

Cílem je tedy vytvořit aplikaci pro Windows, která v reálném čase z API hry získá data, zobrazí je v potřebné formě uživateli, pro vybraná data zobrazí trendy vývoje včetně predikce dalšího vývoje a umožní uživateli ve vhodném formátu data exportovat.

1. Proveďte rešerši existujících řešení aplikací zobrazujících data pro danou hru.
2. Analyzujte API a data, která hra poskytuje. Vyberte z nich vhodná pro tvorbu predikcí a zvolte metodu predikce.
3. Vyberte vhodné technologie a navrhnete architekturu, která bude splňovat všechny požadované funkcionality a bude vhodná pro práci s daty v reálném čase.
4. Proveďte základní návrh uživatelského rozhraní s přehledným zobrazením a vhodným seskupením dat.
5. Implementujte prototyp aplikace a proveďte uživatelské testování s ověřením přesností generovaných predikcí.

Bakalářská práce

DASHBOARD PRO ONLINE ZÁVODNÍ HRU

Ing. Marek Čermák

Fakulta informačních technologií
Katedra softwarového inženýrství
Vedoucí: Ing. Pavel Štěpán
11. května 2024

České vysoké učení technické v Praze
Fakulta informačních technologií

© 2024 Ing. Marek Čermák. Všechna práva vyhrazena.

Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí a nad rámec oprávnění uvedených v Prohlášení, je nezbytný souhlas autora.

Odkaz na tuto práci: Čermák Marek. *Dashboard pro online závodní hru*. Bakalářská práce. České vysoké učení technické v Praze, Fakulta informačních technologií, 2024.

Obsah

Poděkování	vii
Prohlášení	viii
Abstrakt	ix
Seznam zkratek	x
1 Úvod do závodů a simulátoru	2
1.1 Závody série GT World Challenge Europe	2
1.1.1 Průběh závodu	2
1.1.2 Vlajky a penalizace	3
1.1.3 Závodní inženýr	4
1.1.4 Telemetrie	4
1.1.5 Závodní data	5
1.2 Závodní simulátor ACC	6
1.2.1 Technický popis	6
1.2.2 Aktuální podoba zobrazení dat ve hře	6
1.2.3 Problém se zobrazením dat	8
1.3 Obecná formulace požadavků	8
2 Rozbor API hry a poskytovaných dat	10
2.1 API hry pro poskytování dat	10
2.1.1 Vysvětlení fungování UDP připojení	10
2.1.2 UDP připojení ke hře	10
2.1.3 Vysvětlení fungování sdílené paměti	11
2.1.4 Sdílená paměť hry	11
2.2 Komunikace s hrou pomocí UDP	12
2.2.1 Začátek a ukončení komunikace	13
2.2.2 Request zprávy	13
2.2.3 Aktualizační zprávy	13
2.2.4 Zprávy při události	13
2.2.5 Ovládací zprávy	14
2.2.6 Formát dat	14
2.2.7 Problémy s UDP připojením	14
2.3 Popis fungování sdílené paměti	15
2.3.1 Čtení sdílené paměti	15
2.3.2 Formát dat	16
2.3.3 Problémy se sdílenou pamětí	16
2.4 Zhodnocení poskytovaných dat	16

3	Rešerše existujících řešení	17
3.1	ACC Tools	17
3.2	Telemetry Tool for ACC	19
3.3	Sim Racing Telemetry	22
3.4	ACC Drive	23
3.5	Race Element	25
3.6	Srovnání a zhodnocení	26
3.6.1	Finální srovnání	26
3.6.2	Vyhodnocení rešerše	28
4	Výpočty a predikce	29
4.1	Stručný úvod k predikcím	29
4.2	Výběr predikcí	30
4.2.1	Výpočet časových odstupů	30
4.2.2	Časový odhad zbývajícího paliva	31
4.2.3	Predikce soubojů na trati	32
4.2.4	Predikce pitstopu	33
4.2.5	Predikce vývoje technických údajů	35
5	Analýza	36
5.1	Funkční požadavky	36
5.2	Nefunkční požadavky	38
5.3	Případy užití	39
5.4	Realizace požadavků případy užití	41
5.5	Rozsah prototypu	42
5.6	Doménový model	43
5.6.1	Doménový model aplikace	43
5.6.2	Doménový model herních dat	44
6	Návrh a realizace	47
6.1	Výběr technologií	47
6.1.1	Cílová platforma	47
6.1.2	Jazyk a základní infrastruktura	47
6.1.3	Framework uživatelského rozhraní	48
6.1.4	Knihovny	49
6.1.5	Logování	49
6.1.6	Zobrazení grafů	50
6.1.7	Perzistence dat	50
6.1.8	Serializace	50
6.1.9	Matematické výpočty	50
6.1.10	Strojové učení	51
6.1.11	Shrnutí vybraných technologií	51
6.2	Návrh grafického rozhraní	51
6.2.1	Základní rozvržení	52
6.2.2	Domovská obrazovka	52
6.2.3	Nastavení	53
6.2.4	Závodní session	53
6.2.5	Závodní pořadí	54
6.2.6	Závodní situace	54
6.2.7	Informace o zastávce v boxech	54
6.2.8	Telemetrie vozu	56
6.2.9	Telemetrie kol	56
6.2.10	Implementace uživatelského rozhraní	56

6.3	Základní architektura	57
6.3.1	Vnitřní fungování aplikace	58
6.3.2	Komunikace s uživatelským rozhraním	59
6.4	Popis samostatných modulů	60
6.4.1	Modul UDP	60
6.4.2	Modul SM	61
6.4.3	Modul Sdílení	64
6.5	Model tříd	65
6.6	Model komunikace	65
6.7	Příklady implementace predikcí	65
6.7.1	Implementace predikce časových odstupů	65
6.7.2	Implementace predikce časových řad	69
6.8	Jednotkové testy	72
7	Výsledný prototyp	73
7.1	Instalace	73
7.1.1	Nezabalená aplikace	73
7.1.2	MSIX balíček	73
7.2	Výsledná podoba	74
7.3	Ověření predikcí	76
7.4	Uživatelské testování	77
7.4.1	Výběr uživatelů k testování	77
7.4.2	Průběh testování	78
7.4.3	Identifikované problémy při testování	78
7.4.4	Zpětná vazba z testování	78
7.4.5	Vyhodnocení testování	79
7.5	Budoucnost aplikace	80
A	Popis UDP zpráv	83
A.1	UDP zprávy pro ACC	83
A.2	UDP zprávy z ACC	84
A.3	Opakující se části ve zprávách	87
B	Konstanty pro predikci pitstopu	88
C	Dotazníky a scénáře pro testování	90
C.1	Dotazník	90
C.2	Scénáře	91
	Obsah příloženého média	98

Seznam obrázků

1.1	Start závodu GT WCE v Barceloně [2]	3
1.2	HUD ve hře ACC – Kvalifikace	7
1.3	Obrázek ze zastávky v boxu okolo poloviny komunitního 24h závodu	7
2.1	Ilustrace sdílené paměti [18]	11
3.1	ACC tools – příklad manažera pro nastavení vozů	18
3.2	ACC tools – příklad zobrazení výsledkového souboru	18
3.3	ACC tools – příklad závodního inženýra - kalkulačka tlaků	19
3.4	Telemetry Tool for ACC – grafy telemetrie [21]	20
3.5	Telemetry Tool for ACC – základní data [21]	20
3.6	Telemetry Tool for ACC – mapa okruhu [21]	21
3.7	Telemetry Tool for ACC – telemetrie kol [21]	21
3.8	Sim Racing Telemetry – zobrazení při záznamu dat	23
3.9	Sim Racing Telemetry – přehled kol	23
3.10	Sim Racing Telemetry – telemetrie v grafech	24
3.11	Sim Racing Telemetry – zobrazení na mapě	24
3.12	Acc Drive - koláž zobrazení [27]	25
3.13	Race Element – přehled nastavení	26
3.14	Race Element – výběr HUD elementu	27
4.1	Ilustrace vzdálenosti a časového odstupe	30
5.1	Model požadavků	36
5.2	Model případů užití	39
5.3	Doménový model aplikace	43
5.4	Doménový model závodního sezení	44
5.5	Doménový model situace hráče	45
5.6	Propojení modelů hráče a závodního sezení	46
6.1	Návrh domovské obrazovky	52
6.2	Návrh obrazovky nastavení	53
6.3	Návrh obrazovky závodní session	54
6.4	Návrh obrazovky závodního pořadí	55
6.5	Návrh obrazovky závodní situace	55
6.6	Návrh obrazovky zastávky v boxech	56
6.7	Návrh obrazovky telemetrie vozu	57
6.8	Návrh obrazovky telemetrie kol	57
6.9	Architektura modulů s naznačením jejich komunikace	59
6.10	Schéma MVVM [39]	60
6.11	Komunikace tříd při přijetí zprávy spouštěcí UDP přenos	62
6.12	Komunikace tříd při asynchronním čtení příchozích aktualizací	62
6.13	Model tříd aplikace	66
6.14	Model tříd závodního sezení	67

6.15	Model tříd situace hráče	68
6.16	Komunikace tříd při spuštění aplikace uživatelem	69
6.17	Komunikace tříd při spuštění UDP	70
6.18	Komunikace tříd při spuštění sdílené paměti	70
6.19	Komunikace tříd při spuštění predikcí	70
7.1	Instalace aplikace pomocí MSIX instalátoru	74
7.2	Základní stránka výsledné aplikace – světlá varianta	75
7.3	Výsledná stránka nastavení – světlá varianta	75
7.4	Nastavení predikcí – tmavá varianta	75
7.5	Výsledná stránka zastávky v boxu – tmavá varianta	76
7.6	Výsledné zobrazení predikcí v grafu – tmavá varianta	76
7.7	Srovnání predikcí a reálných hodnot tlaků pneu - graf z aplikace	77
7.8	Srovnání predikcí a reálných hodnot tlaků pneu - graf extrahovaných hodnot	77
7.9	Ukázka konfigurace simracing kokpitu pro testování	78
7.10	Znázornění zkušeností a dovedností uživatelů	80
7.11	Průměr udělených bodů z testování	80

Seznam tabulek

2.1	Zprávy zasílané hře	12
2.2	Zprávy zasílané hrou	12
2.3	Paměťové soubory	15
3.1	Plnění požadavků existujícími řešeními	28
5.1	Realizace funkčních požadavků	41
6.1	Výběr technologií	51
6.2	Přijímané zprávy UDP modulem	61
6.3	Zasílané zprávy od UDP modulu	61
6.4	Přijímané zprávy SharedMemory modulem	63
6.5	Zasílané zprávy od SharedMemory modulu	63
6.6	Přijímané zprávy Sharing modulem	64
6.7	Zasílané zprávy od Sharing modulu	64
B.1	Konstanty úkonů při zastávce v boxech	88
B.2	Traťové konstanty pro zastávku v boxech	89

Na tomto místě bych rád poděkoval Ing. Pavlovi Štěpánovi, který vedl tuto práci a poskytoval mi hodnotnou zpětnou vazbu. Dále bych chtěl poděkovat své přítelkyni, Ing. Janě Metelcové, která se mnou období tvorby práce přetrpěla, a dokonce souhlasila s rolí korektorky textu, což byla rozhodně náročná práce. V neposlední řadě patří poděkování i mému kamarádovi, Ing. Jakobovi Jánskému, za jeho podporu při konzultacích závodní problematiky, a stejně tak každému, kdo se účastnil uživatelského testování.

Prohlášení

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

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. V souladu s ust. § 2373 odst. 2 zákona č. 89/2012 Sb., občanský zákoník, ve znění pozdějších předpisů, tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou a veškeré jejich dokumentace (dále souhrnně jen „Dílo“), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla, avšak pouze k nevýdělečným účelům. Toto oprávnění je časově, teritoriálně i množstevně neomezené.

V Praze dne 11. května 2024

Abstrakt

Práce se zabývá návrhem a implementací prototypu samostatné desktopové aplikace, která se dokáže připojit k závodní hře Asseto Corsa Competizione a v reálném čase zobrazovat telemetrická data uživatele závodního vozu a další data probíhajícího závodu (například pořadí závodníků a jejich dosažené časy na kolo). Nejprve je představena závodní problematika, provedena rešerše jiných podobných aplikací, popis obou možných způsobů získávání dat ze hry (UDP komunikace a sdílené paměti), popis získaných dat a jejich výběr pro predikce budoucího vývoje a návrh výpočtu konkrétních predikcí. Následuje softwarový návrh a implementace prototypu aplikace včetně popisu implementace výpočtu predikcí a provedení uživatelské testování na potenciálních budoucích uživateli. Prototyp aplikace je postaven na technologii WinUI 3, využívá obě možnosti získávání dat ze hry (UDP komunikaci a sdílenou paměť), ukládá data pomocí LiteDB databáze a umožňuje jejich export do CSV a JSON formátů. Pro výpočty predikcí je využito knihoven Math.NET Numerics a ML.NET. Prototyp rovněž umožňuje sdílení dat mezi instancemi této aplikace spuštěnými na různých počítačích.

Klíčová slova Asseto Corsa Competizione, online simracing, dashboard, závodní telemetrie, Windows desktopová aplikace, predikce herních dat, WinUI 3, .NET, sdílená paměť, UDP

Abstract

Thesis consists of architecture and implementation of standalone desktop application that is able to connect to the Asseto Corsa Competizione racing game and show telemetric data of user's racing car and another race data (i.e. drivers standings and their lap times), both in real-time. First, the introduction to racing is provided, research of similar applications is carried out, both ways how to get data from game (UDP communication and shared memory) analyzed and the data provided by game described. Also selection of obtained data for those suitable for predictions calculations is made and those calculations proposed. Design and implementation of application prototype follows, with predictions implementation and description of user testing results. Application prototype is build on WinUI 3 technology, utilizes both data source options (UDP and shared memory), saves data into LiteDB database and enables their export into CSV or JSON format. Calculations of prediction uses Math.NET Numerics a ML.NET. libraries. Sharing data between two instances of this application running on separate computers is also enabled by the app prototype.

Keywords Asseto Corsa Competizione, online simracing, dashboard, racing telemetry, Windows desktop app, game data prediction, WinUI 3, .NET, shared memory, UDP

Seznam zkratk

ACC	Asseto Corsa Competizione	–
API	Application Programming Interface	Aplikační programovací rozhraní
FiA	Fédération Internationale de l'Automobile	Mezinárodní automobilová federace
HUD	Head-up Display	Průhledový displej
LTS	Long term support	Dlouhodobá podpora
MVVM	Model-View-ViewModel	–
ORM	Object-relational Mapping	Objektově-relační mapování
UDP	User Datagram Protocol	–
SM	Shared memory	Sdílená paměť
STS	Standard term support	Standardní doba podpory
UC	Use Case	Případ užití
UI	User interface	Uživatelské rozhraní
UX	User experience	Uživatelský zážitek

Úvod

Hra Assetto Corsa Competizione (dále ACC) [1] je jedním ze závodních simulátorů, který směřuje k realistickému závodnímu zážitku. Pořádá se v něm množství online závodů, které zahrnují i vytrvalostní závody na 24 h se střídáním jezdců.

Ačkoli se může zdát, že online závody jsou „pouhé hraní“, lidé v různých komunitách berou simulované závody vážně. Dokonce se někdy o závodění na simulátorech mluví jako o novém druhu sportu. Ti nejlepší byli dokonce vyzkoušeni profesionálními závodními týmy a ukázali, že se svými výkony v reálných vozech na okruzích měřit s ostatními profesionálními jezdci.

Existuje množství různých simulátorů, více či méně oblíbených, a množství podpůrných aplikací, více či méně podařených, podporující různé simulátory a s různým finančním modelem. Zde se však budeme zabývat pouze simulátorem Assetto Corsa Copetizione, který se zaměřuje na třídu cestovních vozů (Grand Tour) GT3 a GT4 s oficiální licencí. I v tomto simulátoru se pořádá množství závodů, i když nepatří mezi top realistické simulátory jako například iRacing. Používá se tak spíše v rámci volnočasových závodních komunit vzhledem k dobrému poměru ceny, realističnosti a zábavnosti.

Ale i lidé v takových komunitách, jezdící především pro zábavu, často hledají možnosti, které by jim pomohly zvýšit výkonnost. Pokud se nebudeme bavit o různých podvodech a „cheatech“, hlavní možností získání výhody patří správné nastavení vozu a volba závodní strategie. K oboujmu je zapotřebí sledovat a analyzovat množství telemetrických dat a mít přehled o kompletní závodní situaci na trati. Zobrazení těchto informací v simulátoru ACC je ale omezené, data nejsou poskytována v dostatečném rozsahu či nejsou prezentována vhodným způsobem. Hra sama poskytuje několik API, pomocí kterých je možné potřebná data získat jinými programy. Vývojáři hry také poskytli příklad velmi jednoduchých prototypů takových aplikací, které ukazují, jak taková API využít.

Bohužel existujících aplikací pro použití s ACC není mnoho, liší se svou kvalitou a možnostmi, a u těch kvalitnější se jedná spíše o placené distribuce. Právě proto jsem byl osloven jednou ze závodních komunit, jestli bych nepřipravil vlastní aplikaci, která by jim data ze hry prezentovala potřebným způsobem. Velmi žádanou dodatečnou funkcionalitou, která by zvýšila zájem o aplikaci, byla schopnost zobrazovat predikce vývoje dat, i s ohledem na usnadnění volby budoucí strategie v závodech. Nová aplikace navíc umožní případné úpravy nebo implementaci nových funkcionalit dle budoucích specifických požadavků konkrétní komunity.

Cílem práce je implementace prototypu aplikace, která bude v reálném čase zobrazovat data ze simulátoru Assetto Corsa Competizione, ukládat je pro pozdější export a pro zvolená data generovat predikce jejich budoucích hodnot. V rámci teoretické části bude poskytnut krátký úvod do světa závodů, provedena rešerše obdobných řešení, analýza a popis API simulátoru a poskytnutých dat. Praktická část bude obsahovat výběr vhodných predikcí a návrh jejich výpočtů, základní analytický návrh aplikace, výběr vhodných technologií pro implementaci prototypu, popis návrhu architektury a uživatelského rozhraní. Výsledně bude provedeno uživatelské testování.

Úvod do závodů a simulátoru

Protože ne všem může být úplně jasná problematika závodů cestovních vozů, natož pak její ztvárnění simulátorem, je obojí v této úvodní kapitole krátce představeno. Toto přestavení má za cíl usnadnit orientaci v dalším textu a také ozřejmit některá rozhodnutí a nastíněné problémy. Zároveň je díky tomu ukázáno co a proč vedlo k tvorbě této práce.

Práce se týká sběru a zobrazení dat ze simulátoru/hry Asseto Corsa Competizione (dále ve zkratce ACC) [1], která zpracovává licencovanou podobu závodní série GT World Challenge Europe [2], dříve známou jako Blancpain GT series. Následující popis se proto zaměřuje na tuto závodní sérii a simulátor.

1.1 Závody série GT World Challenge Europe

Jedná se o mezinárodní sérii závodů pořádanou na různých okruzích v Evropě, namátkou např. slavné okruhy Spa Francorchamps, Paul Ricard nebo Nürburgring. Tyto závody probíhají podle pravidel FiA a účastní se jich převážně vozy kategorie GT3, což jsou pro závodní účely upravené běžné sportovní automobily (podle pravidel dokonce musejí být), jako například Ferrari 488, Audi R8, Porsche 911 nebo Aston Martin Vantage. [2, 3, 4]

Závody se dělí do dvou kategorií – vytrvalostní s délkou od 3h do 24h a sprinty s délkou 60 minut. Každému vozu je přidělena posádka více jezdců, pro vytrvalostní závody jsou jezdců 3, pro sprinty 2, kteří se v průběhu závodu střídají. Střídání probíhá během zastávky v boxech a musí se řídit určenými pravidly, např. každý z posádky musí během závodu řídit stanovenou minimální dobu. V každém ze závodů jsou vozy rozděleny do 4 výkonnostních tříd podle schopností a zkušeností posádky - Pro, Silver, Pro-Am a Am. V kategorii Pro jsou nejzkušenější a nejrychlejší profesionální jezdci, v kategorii Silver jsou jezdci méně zkušení, kteří jsou většinou o něco pomalejší. Kategorie Am zahrnuje amatérské jezdce, pro které jsou závody hobby a kteří si účast třeba zaplatili. Nakonec kategorie Pro-Am je, jak název napovídá, kombinací jezdců patřících do kategorií Pro a Am. [4]

1.1.1 Průběh závodu

Závodní víkend probíhá v několika dnech a dělí se na několik částí. Na začátku se odehrává několik měřených tréninků, které slouží především pro vyzkoušení chování vozu na trati, nalezení jeho optimálního nastavení, a k porovnání očekávaných časů na kolo s časy reálně naměřenými. Poté přijde řada na kvalifikace, kde se jezdci samostatně pokouší zajet co nejrychlejší čas na kolo. V každé kvalifikaci řídí jiný člen posádky, a jejich dosažené časy se nakonec sčítají. Podle



■ Obrázek 1.1 Start závodu GT WCE v Barceloně [2]

těchto časů se pak určí pořadí na startu závodu, kdy vozy (posádky) s lepším kvalifikačním časem startují vpředu. [2, 4]

Do závodu se startuje buď z pevných pozic, nebo se provádí letmý start, kdy startovní pole seřazené do dvoustupu projíždí v okamžiku startu cílovou rovinkou omezenou rychlostí. Závod trvá buď uvedený počet kol a končí, jakmile tohoto počtu dosáhne první jezdec, ostatní už pouze dojíždí své aktuální kolo. Variantou pro vytrvalostní závody je pevný čas závodu, kdy je ukončen prvním projetím cílem vedoucím vozidlem, které následuje po uplynutí určeného času (tj. když vedoucí vůz dokončí své kolo po uplynutí času závodu). Ostatním vozům následně končí závod po projetí cílem, ať už jsou v jakémkoliv kole (tj. pouze dojíždí svá aktuální kola a nedoplňují si zbývající kola, která ztrácejí na vítěze). Podle výsledného pořadí se pak posádkám celkově, ale i v rámci kategorií, udělují bodová ohodnocení do tabulky šampionátu. Prvním třem posádkám každé kategorie se také udělují při pódiovém vyhlášení trofeje. [2, 4]

1.1.2 Vlajky a penalizace

V průběhu závodu jsou jezdcům na trati tradičně signalizovány důležité informace pomocí vlajek. Každá vlajka má svůj význam daný její barvou a jezdec se jí musí bezpodmínečně řídit. Detailně je signalizace pomocí vlajek popsána v rámci pravidel FiA v dodatku H [5, str. 12]. Zde jsou uvedeny pouze nejdůležitější vlajky a jejich význam.

- Zelená vlajka – start závodu, restart závodu nebo konec omezení žlutou vlajkou
- Červená vlajka – zastavení závodu, často z důvodu nebezpečné situace na trati (např. velká nehoda nebo náhlé velmi nepříznivé počasí)
- Žlutá vlajka – problém na okruhu, jezdec musí zpomalit a ve vyznačeném úseku nesmí předjíždět
- Modrá vlajka – jezdec je o kolo(a) zpět a musí uvolnit místo pro předjetí rychlejšímu jezdcovi

- Černá vlajka – diskvalifikace
- Černá vlajka s oranžovým kruhem – jezdec má na vozidle technický problém se kterým nesmí pokračovat, musí zajet do boxu k nápravě nebo odstoupit
- Šachovnicová vlajka- konec závodu – jezdec projel cílem

Za nedodržování vlajek, překročení limitů trati (např. zkracování cesty), nebo překračování maximální povolené rychlosti v boxu se může vozu udělit trest v podobě časové penalizace, průjezdu boxovou uličkou nebo diskvalifikace. Časová penalizace se uplatňuje formou čekání v rámci první zastávky v boxech (během té doby se na vozidle nesmí pracovat), nebo v případě že se již žádná zastávka neodehraje, přičtením k výslednému času závodu. [6, str. 49]

1.1.3 Závodní inženýr

Každý závodní tým má minimálně jednoho, ale spíš více závodních inženýrů, kteří se starají o přípravu a nastavení vozu na závod a tvorbu závodní strategie. Při závodě pak pomocí obrazovek s dostupnými aktuálními daty ze senzorů ve vozech sledují jejich stav, výkonnost, a detekují možné problémy a možnosti jejich řešení. Na základě telemetrie (viz 1.1.4), radiové komunikace s aktuálním jezdce ve voze a podle zvolené strategie pak rozhodují o vhodném okamžiku pro zastávku v boxech, výměně pneumatik, úpravě aerodynamiky a pomáhají na dálku řešit různé technické problémy na voze. Často pomáhají jezdcem nastavit optimální tempo v průběhu závodu, informují ho o jeho pozici, časech, kdy se bránit a kdy se snažit soupeře předjet. [7]

1.1.4 Telemetrie

Podle pravidel GT3 [3, str. 2] je definice telemetrie doslova „*Transmission of data between a moving car and anyone connected with the entry of that car*“ neboli volně přeloženo „*Přenos dat mezi jedoucím vozem a kýmkoliv napojeným na příjem od tohoto vozu*“. Přenos některých dat je dokonce povinný, například informace o podélných a příčných přetížích, otáčkách apod. Na vozidle je proto namontováno množství senzorů, které sbírají tato povinná data, ale i řadu dalších parametrů vozu (tlaky a teploty pneumatik, množství paliva a jeho spotřebu, otáčky motoru a jeho různé teploty, ...). Přenášení dat v reálném čase je povoleno pouze jednosměrně z vozu do „počítače závodního inženýra“, který má na jejich zobrazování a vizualizaci speciální software, a který je také ukládá pro pozdější analýzu. Pro úpravy některých parametrů palubních systémů (nastavení motoru, ABS, apod.) je proto potřeba kontaktovat přímo jezdce (např. vysílačkou), aby příslušné změny v nastavení provedl. [3]

Pro přiblížení je níže uveden přehled vybraných základních dat. Dvoje vybraná jsou více vysvětlena v podkapitolách 1.1.4.1 a 1.1.4.2.

- Stav paliva, jeho spotřeba na kolo a na jak dlouho palivo zbývá
- Opotřebenění a teploty pneumatik a aktuální tlaky nahuštění
- Brzdné tlaky, opotřebenění brzdových destiček a kotoučů a jejich vývoj
- Přetížení, fungování asistentů trakce a ABS – sledovány hlavně v zatáčkách
- Prokluzování pneumatik
- Vyvážení brzd (brzdná síla přední vs. zadní)
- Teplota chladící kapaliny a spalin
- Směsi pneumatik a brzd
- Stav trati a přilnavost

1.1.4.1 Teplota pneumatik

Teplota pneumatik je jeden ze základních parametrů. Pneumatika má optimální přilnavost k povrchu trati pouze v určitém rozmezí teplot. Pokud je teplota příliš nízká nebo vysoká, přilnavost klesá a vůz v zatáčkách nedrží, klouže a má tendenci jít do smyku, což není dobrá vlastnost pro řízení ani pro výsledné časy na kolo. Teplotu pneumatik ovlivňuje povaha trati (např. se přehřívá pouze jedna pneumatika) nebo nevhodné nahuštění. Zároveň se dá z rozdílu teplot na vnějšku, středu a vnitřku pneumatiky teplot zjistit, jestli je dobře nastaven odklon kol od svislé roviny. [8]

1.1.4.2 Teplota brzd

Teplota brzd výrazně ovlivňuje schopnost vozu brzdit. Příliš studené brzdy znamenají menší účinek brždění, tedy i delší brzdnou dráhu. Je proto nutné být při brždění se studenými brzdami opatrný a brzdit včas. Pokud se naopak brzdy přehřívají, tak se jednak snižuje jejich životnost a účinnost brždění, jednak může dojít k tomu, že v případě přílišného přehřátí mohou přestat brzdit úplně a dokonce může brzda začít hořet. Z vývoje teplot brzd se pak dá zjistit, jestli se má víc nebo méně otevřít kanál přivádějící k brzdám chladící vzduch. Chlazení se ale nesmí otevřít příliš, protože by mohlo brzdy příliš ochlazovat a přívod velkého množství vzduchu působit zbytečný aerodynamický odpor. Naopak při příliš uzavřeném chladícím kanálu se brzdy začnou přehřívát. Jelikož nastavení chlazení brzd závisí mimo jiné na charakteru trati (jestli má místa s prudkým bržděním a vhodné rovinky k ochlazení brzd) a na počasí, jsou zde telemetrická data velmi potřebná. [9]

1.1.5 Závodní data

Během závodu je třeba sledovat, kde na trati se nachází nejbližší soupeři, jaké zajíždějí časy, jestli je výhodné pro jezdce se snažit soupeře dojet, kdy zajet do boxu, aby se vůz při výjezdu nedostal do silného provozu a tím neztrácel čas, a mnoho dalších podobným údajů. Protože jsou závody dlouhé a v jednu chvíli mohou závodit různé kategorie posádek vozidel zároveň, může se snadno stát, že auto co jedou zrovna blízko sebe mohou být o jedno i více kol vzájemně posunuta nebo vůbec nepatří do stejné kategorie. To vše dělá celou závodní situaci velmi složitou. Proto je třeba sbírat další množství dat, která umožní týmu udržovat si přehled o dění na trati.

Zde jsou uvedeny některé vybrané příklady závodních dat, která mohou být využita:

- Základní informace o závodu
- Čas posledního kola
- Nejrychlejší kolo
- Průměrný čas na kolo
- Průměrný čas posledních x kol
- Ztráta na jezdce před sebou a odhad, za kolik jej může aktuálním tempem dostihnout
- Náskok na jezdce za sebou a odhad, za kolik může být s aktuálním tempem dostihnout
- Srovnání časů na kolo s ostatními jezdci
- Přehled pořadí celého závodního pole
- Časová ztráta při zastávce v boxu
- Počet pozic ztracených při zastávce v boxu

- Aktuální kolo a rozdíl vůči nejlepšímu kolu
- Polohy ostatních vozidel na trati
- Počasí (teploty, déšť, vítr) a jeho možný vývoj
- Udělené časové penalizace
- Signalizace vlajek, přehled nehod

1.2 Závodní simulátor ACC

Závodní simulátor Asseto Corsa Competizione (ACC) se snaží co nejvěrněji ztvárnovat výše uvedené závody. Je dokonce oficiální podporovanou hrou/simulátorem, který poskytuje uživatelům možnost zažít na vlastní kůži závody v kategorii vozů GT3, a to jak proti umělým protivníkům, tak proti ostatním lidem přes internet. Obsahuje všechny typy vozů a tratí, které se v šampionátu GT World Challenge Europe vyskytují, a navíc také oficiální seznam týmů a jezdců, kteří se šampionátu účastní, včetně designu vozů. Organizátor šampionátu GT World Challenge Europe, ale i samy závodní komunity tak v ACC pořádají celé vlastní série závodů, kdy se snaží o co nejrealističtější zážitek a proto se jezdí závody s plnou reálnou délkou (i oněch 24h), kdy se hráči střídají podobně jako jezdci v reálném závodě. Zároveň se stejně jako závodní inženýři snaží s pomocí dat co nejlépe nastavit svůj závodní vůz tak, aby se dobře řídil a byl rychlý. Rovněž řeší závodní strategii – ideální počet a časy zastávek a jejich parametry (množství paliva k tankování, přezutí, výměnu brzd,...). V průběhu závodu také mohou provádět úpravy nastavení vozu (kontroly trakce, ABS, vyvážení brzd,...), například z důvodu změny vnější podmínky (počasí, stav tratě, ...) nebo přílišného nárůstu opotřebení pneumatik nebo brzd. [1, 10]

1.2.1 Technický popis

Hra vyšla 29.5.2019 a je určena jak pro osobní počítače, tak pro konzole. Verze pro osobní počítače je dostupná pouze přes platformu *STEAM*, kde se její cena v době psaní práce pohybuje okolo 40€ (údaj k 24.03.2024), a to pouze pro systém Windows (minimálně ve verzi October 2020 Update). Vývoj má na svědomí *Kunos Simulazioni* a vydání pak *505 Games*. [1, 11]

Pro zobrazení je možné použít jeden display o rozlišení až 1920x1080 pixelů, nebo použít rozšíření pro 3 monitory s rozlišením až 5760x1080 pixelů.¹ Uživatelé mohou pro ovládání závodního vozu použít klávesnici, gamepad, nebo pro maximální realističnost připojenou periferii volantu a pedálů.

1.2.2 Aktuální podoba zobrazení dat ve hře

Pokud se nepočítají jiné již existující externí aplikace pro zobrazení dat, hra samotná poskytuje jezdcům telemetrická a závodní data pomocí HUD (head-up display) přímo ve hře, tedy zobrazování po okraji displeje. Na obrázku 1.2 je ilustrována aktuální situace včetně popisků zobrazovaných dat.

Z obrázku 1.2 je vidět že HUD zobrazení dat je velmi omezená, prakticky se jedná pouze o aktuální stav pro zhruba přehled o situaci a stavu vozu. Lze upřímně říci, že pro základní použití je zobrazení dostatečné, navíc lze HUD částečně doplnit simulovaným multifunkčním displejem (MFD) vozidla, jak je vidět na obrázku 1.3. Pokud někdo ale potřebuje komplexnější telemetrická data pro lepší nastavování, pak je toto HUD zobrazení naprosto nedostatečné. Některé hodnoty jsou uvedeny exaktně, některé pouze barevnou škálou (např. teploty pneumatik) a grafy jejich vývoje pak nejsou dostupné vůbec nikde, a to ani z jiných nabídek někde ve hře nebo nastavení.

¹Pro použití 3 monitorů je nutné virtuálně spojit 3 monitory v jeden široký, například pomocí NVIDIA Surround.



■ Obrázek 1.2 HUD ve hře ACC – Kvalifikace



■ Obrázek 1.3 Obrázek ze zastávky v boxu okolo poloviny komunitního 24h závodu

1.2.3 Problém se zobrazením dat

Pokud chce závodník pouze s pomocí hry sledovat vývoj některých dat či porovnávat data aktuální s minulými, je odkázán maximálně na svoji paměť, případně na nějakou formu poznámek, což je během řízení obtížně proveditelné. Problém tak nastává v případě, kdy potřebuje jezdec provést nějaké rozhodnutí, kdy se závodník snaží provést nějaké odhady, které se týkají až už budoucího, nebo minulého vývoje závodních nebo telemetrických dat. Takovým problémem může být třeba rozhodnutí jestli má zajet do boxu pro další palivo, nebo mu vystačí do konce závodu. V případě vytrvalostních závodů (např. na 24h) může jezdec chtít zjistit stav paliva, brzd a pneumatik a zároveň mít k dispozici výhled pro nejbližší kola, aby věděl, jak moc může oddalovat zastávku pro získání výhody. Nakonec může plánovat zastávku v boxu tak, aby ztratil co nejméně pozic a na výjezdu se nedostal do balíku jiných soupeřů čímž by ztratil další čas.

Příkladem může být jednoduché sledování stavu paliva. Závodník má obecně k dispozici údaje o aktuálním stavu paliva, spotřebě na kolo a ví kolik času mu zbývá do konce závodu. Aby věděl, jestli má dostatek paliva a není potřeba dotankovat, musí provést následující výpočet, kde f_{cr} je spotřeba paliva na kolo, t_r zbývající čas závodu a lt_m průměrný čas závodu:

$$f_{cr} * \frac{t_r}{lt_m} \quad (1.1)$$

Výpočet je sice relativně jednoduchý, avšak musíme uvažovat, že je závodník musí počítat sám v hlavě, zatímco se soustředí na samotné řízení. Navíc ani nemusí umět přesně odhadnout jaké bude mít časy na kolo ve zbytku závodu. Může se tak velmi snadno dopustit chyby i kvůli tomu, že se pro zjednodušení výpočtu bude snažit ne zrovna pěkné vstupy co nejvíce zjednodušit, čímž se snadno dostane mimo toleranci správného výsledku. Protože je snaha dojždět s minimální rezervou paliva cca 1 až 2 litry, větší chyba může znamenat, že dostatek paliva nakonec nezbyde.

Stejná situace je i u dalších údajů, jako například opotřebení brzdových kotoučů a destiček nebo pneumatik, kde navíc ani údaje o degradaci na kolo nejsou k dispozici a musí se získat z historického trendu. V takovém případě je ve své podstatě jezdec odkázán na svůj vlastní pocit z jízdy, brždění apod. Přitom zrovna takové výpočty jsou snadno řešitelné pomocí aplikace, která by tyto dostupné telemetrické údaje ze hry četla a jezdcům by na obrazovku poskytovala přímo výsledek buď v podobě výsledného čísla, nebo grafu budoucího trendu.

Další možností je, že by s danou problematikou mohl pomoci závodní inženýr, stejně jako v reálném závodě. Ti ale mají kompletní přehled telemetrických dat přímo ze závodního vozu, které dostávají v reálném čase a činí rozhodnutí na jejich podkladě.

1.3 Obecná formulace požadavků

Z výše uvedených skutečností tak přirozeně vyplynula potřeba externí aplikace pro vhodné zobrazení telemetrie. Zároveň z nich vyplývají obecné požadavky, které jsou na takovou aplikaci kladeny, a které problémy by měla řešit.

Navrhovaná aplikace by tak v základu a bez potřeby větší konfigurace měla umět zobrazovat závodní a telemetrická data v průběhu závodu a případně umožnit jejich export po závodě k další analýze. Sice je dostupných několik podobných aplikací, ale hlavní invencí oproti ostatním by měla být schopnost provádět jejich zpracování a možnost predikovat jejich vývoj nebo jiné potřebné údaje. Takové predikce by mohly poskytnout závodníkovi výhodu, protože by byl schopen provádět kvalifikovaná rozhodnutí o strategii a dalších aspektech na základě jasných dat místo více či méně přesné vlastní predikce založené převážně na zkušenosti a nejasném nebo neúplném přehledu o situaci.

Pro případ, kdy by bylo potřeba zobrazit data na jiném stroji, je vhodné umožnit vysílání kompletních dat k zobrazení a zpracování po lokální síti. Například pro případ, kdy někdo má po ruce kolegu jako závodního inženýra, nebo při požadavku detailnějších predikcí s větším potřebným HW výkonem, který by jinak zpomaloval samotnou hru.

Příkladem základních kalkulovaných predikcí telemetrických dat může být vývoj stavu pneumatik, teploty trati, spotřeby paliva nebo opotřebení brzd. Případně u závodních dat se může jednat o předpověď počtu kol k dojetí protivníka vepředu, nebo počet pozic ztracených zastávkou v boxu. Výběr konkrétních predikcí a způsob jejich výpočtu je uveden v příslušné kapitole.

Rozbor API hry a poskytovaných dat

V této kapitole jsou popsány oba způsoby, kterými se dají ze hry získat potřebná data. Pro každý z nich je uveden krátký teoretický úvod přibližující jeho fungování a jakým způsobem se s daným zdrojem pracuje a jaká data pomocí něj lze získat.

2.1 API hry pro poskytování dat

Hra poskytuje data pomocí dvou technicky rozdílných způsobů, a to prostřednictvím UDP připojení a sdílené paměti. Dokumentace k datům, která se dají získat pomocí UDP zpráv nebo sdílené paměti, není oficiálně přímo dostupná, nicméně potřebné informace lze získat nepřímo. Pro oba typy přenosu dat (API) je proto uvedeno jakým způsobem se tyto informace získaly. Dále v kapitole je pak pro obě API zvlášť popsáno jaká data poskytují a jakým způsobem se s těmito zdroji dat pracuje.

2.1.1 Vysvětlení fungování UDP připojení

UDP (User Datagram Protocol) je jeden z protokolů transportní vrstvy OSI modelu, který přenáší zprávy mezi aplikacemi prostřednictvím datagramů. Pro identifikaci zdrojových a cílových síťových aplikací využívá čísla portů. Po síti je pak datagram přenášen jako data v IP paketu. Je jednoduchý, s malou režii a nevytváří spojení mezi komunikujícími počítači, což znamená, že také neposílá navíc žádné zprávy pro ustavení spojení (např. známé 3krokové) ani pro potvrzení o doručení jako protokol TCP. Díky tomu je UDP protokol velmi rychlý a efektivní, ale na druhou stranu u zpráv neposkytuje záruku doručení a pořadí (to naopak zaručuje protokol TCP) a musí se s možnými ztrátami počítat a případně je řešit jiným způsobem. Je tak spíše vhodný pro jednoduchou výměnu dat, kde nevádí jejich občasná ztráta, ale je vyžadována nízká latence a vysoká rychlost (často se jedná o komunikaci žádost-odpověď). Jako příklad použití lze uvést přenos videa a hlasu, hraní online her, nebo služby DNS. [12, 13, 14]

2.1.2 UDP připojení ke hře

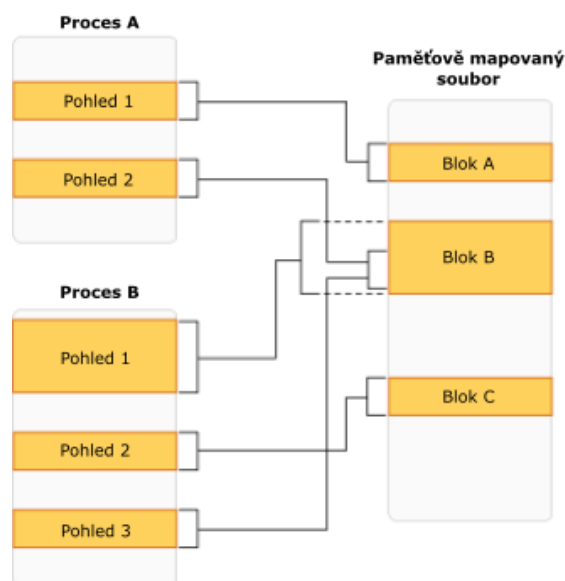
Z pátrání a diskuzí na herních fórech vyplynulo, že žádný oficiální dokument popisující práci s UDP API simulátoru ACC neexistuje. Naštěstí je možné z portálu Steam stáhnout bezplatný dedikovaný server pro hostování závodních session, určený právě pro ACC [15]. Ve své instalační složce má tento server podsložku `sdk`, kde je mimo jiné obsažena funkční aplikace, která funkčnost

UDP připojení demonstruje, včetně svého zdrojového kódu. Vzhledem ke svému umístění není pochyb o autorství vývojářů hry ACC. Definice o podobě a fungování UDP zpráv byly proto použity přímo z tohoto demonstrátoru a také popis věnovaný UDP připojení v části 2.2 tak vychází především z informací získaných tímto způsobem.

2.1.3 Vysvětlení fungování sdílené paměti

Sdílená paměť, nebo také *Paměťově mapované soubory* (memory-mapped files) je způsob v systému MS Windows, jak umožnit více procesům (aplikacím) na stejném počítači sdílet data. Funguje tím způsobem, že jeden proces/aplikace voláním Win32 metody `CreateFileMapping` rezervuje potřebnou velikost sdílené paměti a pojmenuje ji. Jiný proces/aplikace se voláním další Win32 metody `OpenFileMapping` a uvedením příslušné identifikace na stejnou paměť připojí. Obě metody pak vracejí ukazatel na dané místo sdílené paměti, se kterým oba procesy/aplikace mohou zacházet jako s jakýmkoliv jiným vlastním ukazatelem do dynamické paměti. Díky tomu se jedná o extrémně rychlý způsob sdílení dat, který oproti jiným způsobům také umožňuje náhodný přístup ke sdíleným datům, tedy lze číst pouze to co je aktuálně požadováno. Čtení nebo zápis do sdílené paměti je pak možný pomocí interní datové struktury, která bude binární obsah sdíleného souboru správně mapovat do svých datových typů – obě strany pak musí mít toto mapování stejné. [16, 17, 18, str. 971]

Přístupu k souborům sdílené paměti může být uděleno různé oprávnění, které zabraňuje editaci z nechtěných zdrojů. Souborů, které se mezi procesy mapují může také být více, mapovat jenom části, ale také různá mapování překrývat. Schéma procesů se sdílenou pamětí je ukázáno na obrázku 2.1. [17, 18]



■ Obrázek 2.1 Ilustrace sdílené paměti [18]

2.1.4 Sdílená paměť hry

Pro data poskytovaná sdílenou pamětí dokument popisující poskytovaná data existuje. Lze jej stáhnout z oficiálního informačního fóra [19] zaměřeného na simulátory vývojářů *KUNO'S Simulazioni Srl*. Bohužel tento dokument je ke stažení pouze v neaktuální starší verzi. Není to ale nijak

zásadní problém, protože u sdílené paměti je zde zachována zpětná kompatibilitu. Jedinou nevýhodou neaktuální dokumentace je tak pouze to, že mohla být přidána další data, která se bez novější dokumentace nedají použít. V případě, že by došlo ke zveřejnění novější dokumentace, lze snadno rozsah čtených dat adekvátně rozšířit. Popis čtení sdílené paměti v části 2.3 proto využívá uvedený dokument.

2.2 Komunikace s hrou pomocí UDP

Je definováno několik druhů zpráv, které lze odesílat ve směru z aplikace do hry ACC, a naopak, které se posílají ve směru ze hry ACC do aplikace. Přenos je sice veden pomocí UDP, ale komunikace je zároveň identifikována pomocí `CONNECTION_ID`, ke kterému se zasílané zprávy vztahují. Díky tomu může zasílat pro několik spojení zároveň (dokonce i pro stejnou cílovou adresu a port), které se pak odlišují právě pomocí `CONNECTION_ID`.

UDP připojení je nutno nejprve nastavit v konfiguračním souboru `broadcasting.json`, který obsahuje definici portu pro UDP komunikaci, heslo pro připojení a heslo pro ovládání hry pomocí zpráv. Příklad fungující konfigurace je uveden níže. Stejně hodnoty se musí použít při ovládání komunikace.

```
{
  "updListenerPort": 9000,
  "connectionPassword": "asd",
  "commandPassword": ""
}
```

V tabulce 2.1 je uveden výčet zpráv zasílaných od aplikace hře a v tabulce 2.2 výčet zpráv zasílaných naopak hrou aplikaci. Jejich kompletní popis včetně definovaného obsahu je uveden v příloze A. Dále je v této části popsáno, jakým způsobem probíhá komunikace a jak se při ní tyto zprávy využívají.

■ **Tabulka 2.1** Zprávy zasílané hře

Název	Účel
REGISTER	Žádost o registraci a start vysílání
DEREGISTER	Žádost o ukončení vysílání
REQUEST_TRACK_DATA	Žádost o informace o trati
CHANGE_HUD_PAGE	Požadavek na změnu zobrazení ve hře
CHANGE_FOCUS	Požadavek na změnu vozu sledovaného kamerou
INSTANT_REPLAY	Požadavek na přehrání záznamu

■ **Tabulka 2.2** Zprávy zasílané hrou

Název	Účel
REGISTRATION_RESULT	Informace o vysílání a výsledku registrace
TRACK_DATA	Informace o trati
ENTRY_LIST	Informace o aktivních vozech (seznam id)
CAR_INFO_DATA	Informace o konkrétním voze a jeho posádce
REALTIME_UPDATE	Aktualizace aktuálního stavu sezení (závodu)
REALTIME_CAR_UPDATE	Aktualizace aktuálního stavu konkrétního vozu
BROADCASTING_EVENT	Informace o nastalé události

2.2.1 Začátek a ukončení komunikace

Komunikace s hrou začíná žádostí o zahájení vysílání odesláním `REGISTER` zprávy. Ta obsahuje vlastní název pro UDP spojení, hesla pro připojení a ovládání hry, a požadovaný interval pro zasílání aktualizací zpráv. Hesla musí odpovídat těm definovaným v konfiguraci hry (viz. 2.2). Pokud je session ve hře aktivní, hra vrátí potvrzovací `REGISTRATION_RESULT` zprávu obsahující identifikaci spojení definovanou `CONNECTION_ID`, dále informace o úspěchu, jestli spojení umožňuje editaci hry, a případnou chybovou hlášku. Pokud bylo ustavení spojení úspěšné, hra v tuto chvíli začne vysílat aktualizací zprávy v definovaném intervalu a zprávy o událostech.

Pro ukončení spojení je potřeba zaslat `DEREGISTER` zprávou žádost o ukončení vysílání s příslušným `CONNECTION_ID`¹. Po přijetí této zprávy hra bez dalšího přestane s jakýmkoliv dalším vysíláním a nijak ukončení nepotvrzuje.

Spojení je ukončeno automaticky také v případě, kdy uživatel ukončí session přímo ve hře. Není to ale ohlášeno žádnou zprávou z hry, které tak jednoduše přestane vysílat. Pro tuto situaci je tedy nutné stanovit čas pro vypršení spojení (timeout).

2.2.2 Request zprávy

Po zahájení spojení je třeba zaslat žádosti o základní data. Jedná se především o informace o trati, vozidlech a jezdcích, kteří se sezení účastní. Všechny žádosti obsahují pouze identifikaci spojení, ke kterému se vztahují, prostřednictvím `CONNECTION_ID`. Typ žádosti je pak určen čistě typem zprávy.

Žádost o data o trati se provádí prostřednictvím zprávy `REQUEST_TRACK_DATA`, na kterou hra požadovaná traťová data vrátí v obsahu zprávy `TRACK_DATA`. Tato zpráva obsahuje například název trati, její identifikaci, délku a informace o kamerách podél trati. Tato data se v průběhu nemění a proto by mělo stačit ji zavolat pouze jednou po začátku komunikace bez potřeby další aktualizace.

Prostřednictvím zprávy `REQUEST_ENTRY_LIST` je možné požádat o informace o vozech a jejich posádkách. Hra na tuto žádost vrátí jednu zprávu `ENTRY_LIST` obsahující jednoduchý seznam identifikátorů (ID) všech aktivních vozů. Pro každý aktivní vůz pak zašle další zprávou `CAR_INFO_DATA` bližší informace o voze, jako opět jeho ID, název týmu, závodní číslo, ale také seznam jeho řidičů v posádce (včetně jmen) odkaz na toho, kdo aktuálně vůz řídí.

2.2.3 Aktualizační zprávy

Na základě intervalu, definovaném při zahájení spojení, hra periodicky zasílá dva typy zpráv.

První zprávou je `REALTIME_UPDATE` obsahující obecná data o session, jako aktuální a zbývající čas, počasí, teploty vzduchu a vozovky, typ a fázi session apod. Tato zpráva je v rámci jednoho aktualizacího intervalu zasílána pouze jednou.

Druhou zprávou je `REALTIME_CAR_UPDATE` obsahující data o konkrétním voze, především jeho lokace na trati, závodní pozice, rychlost, dosažené a aktuální časy, apod. V rámci jednoho aktualizacího intervalu je zasílána jedna zpráva pro každý z aktivních vozů (může být až 60 vozů).

2.2.4 Zprávy při události

V případě, že se nějakému vozu přihodila událost, je hrou zaslána zpráva `BROADCASTING_EVENT`. Takovou událostí může být například začátek nebo ukončení závodu. Pro jednotlivé vozy to například může být dokončení kola, dosažení nejlepšího času, obdržení penalizace nebo nehoda.

¹Zmíněný příklad aplikace pro UDP posílá tuto zprávu chybně bez `CONNECTION_ID` a k ukončení spojení nedochází.

Zpráva pak obsahuje typ události, identifikaci dotčeného vozu a příslušný text týkající se dané události.

2.2.5 Ovládací zprávy

Poslední kategorií jsou zprávy které přímo ovládají hru. V rámci aplikace nejspíše nebudou využívány a jsou zde uvedeny jenom pro pořádek. Svě použití by měly pro situace, kdy by předmětem byla aplikace spíše pro účely komentátora. Ve své podstatě se také jedná o žádosti na které hra neposílá odpověď, ale provede uvnitř simulátoru příslušnou změnu.

První zprávou je `CHANGE_HUD_PAGE`, která přepíná uživateli zobrazení prvků na displeji. Zvolené cílové zobrazení je ve zprávě definované svým názvem.

Druhou je `CHANGE_FOCUS` zpráva měnící vůz zaměřený/zobrazovaný aktuální kamerou ve hře a případně i změnu kamery jejíž obraz je poskytován uživateli. Ve zprávě je tak nutné definovat identifikaci požadovaného vozu a název kamery která se má použít.

Třetí zprávou je `INSTANT_REPLAY` startující na obrazovce uživatele přehrání záznamu. Zasláná zpráva pak specifikuje čas a o délku záznamu k přehrání, vozidlo na které se má záznam zaměřit a kamera, která se má použít.

2.2.6 Formát dat

Všechny zprávy obsahují byte identifikující typ zprávy a veškerý další obsah v binárním formátu. To zmenšuje velikost zpráv, ale musí se s tím počítat v případě vytváření a čtení zpráv. Proto musí být zprávy vhodně mapovány (deserializovány) podle definovaného obsahu a jeho datových typů, aby nedošlo k chybnému čtení a výsledné nesrozumitelnosti nebo chybovosti dat.

2.2.7 Problémy s UDP připojením

UDP připojení vykazuje několik problémových bodů, které mohou znesnadňovat její použití. Zde jsou ty z nich, na které je možné narazit, uvedeny.

Prvním problémem je, že změně seznamu aktivních vozidel ve hře, například když se někdo do hry přihlásí či z ní odejde, hra sama automaticky o změně žádné informace nezašle. Pro aktualizace seznamu aktivních vozů a jejich informací je proto nutné aktivně provést další dotaz zprávou `REQUEST_ENTRY_LIST`. Jednou z možností řešení je periodické dotazování na tyto změny, které ale může zahlcovat zbytečně síť, a v případě větších intervalů má aktualizace vždy nějaké zpoždění. Další z možností je se spoléhat na informace ze sdílené paměti, která přehled aktivních vozů také obsahuje, a zažádat o aktualizaci dat po detekování změny s její pomocí. Do třetice je možnost zažádat o aktualizaci seznamu vozidel při obdržení aktualizací zprávy o neznámém vozu, což ale nezohledňuje situace, kdy nějaký aktivní vůz ubyde.

Další problém představuje situace, kdy v některých případech po zaslání zprávy, např. registrační `REGISTER`, dojde k pádu hry. Z chybové hlášky hry nelze poznat co je příčinou, ale lze vidět, že se jedná o problém uvnitř kódu hry.

```
Fatal error!

Unhandled Exception: 0xe06d7363

0x00007ff90cf0ab89 KERNELBASE.dll!UnknownFunction []
0x00007ff8e3486ba7 VCRUNTIME140.dll!UnknownFunction []
0x00007ff74509043e AC2-Win64-Shipping.exe!UnknownFunction []
0x00007ff74508fc1c AC2-Win64-Shipping.exe!UnknownFunction []
0x00007ff742cd7f3d AC2-Win64-Shipping.exe!UnknownFunction []
0x00007ff745090ef4 AC2-Win64-Shipping.exe!UnknownFunction []
0x00007ff742cd8f7d AC2-Win64-Shipping.exe!UnknownFunction []
```

```

0x00007ff742a81230 AC2-Win64-Shipping.exe!UnknownFunction []
0x00007ff742a82409 AC2-Win64-Shipping.exe!UnknownFunction []
0x00007ff7443b75c4 AC2-Win64-Shipping.exe!UnknownFunction []
0x00007ff7443a1505 AC2-Win64-Shipping.exe!UnknownFunction []
0x00007ff744a0d787 AC2-Win64-Shipping.exe!UnknownFunction []
0x00007ff744a0edfd AC2-Win64-Shipping.exe!UnknownFunction []
0x00007ff742e108db AC2-Win64-Shipping.exe!UnknownFunction []
0x00007ff742e15347 AC2-Win64-Shipping.exe!UnknownFunction []
0x00007ff744a16702 AC2-Win64-Shipping.exe!UnknownFunction []
0x00007ff744a18377 AC2-Win64-Shipping.exe!UnknownFunction []
0x00007ff744710b4c AC2-Win64-Shipping.exe!UnknownFunction []
0x00007ff744623dce AC2-Win64-Shipping.exe!UnknownFunction []
0x00007ff7423a7fc5 AC2-Win64-Shipping.exe!UnknownFunction []
0x00007ff7423ae9bc AC2-Win64-Shipping.exe!UnknownFunction []
0x00007ff7423aea2a AC2-Win64-Shipping.exe!UnknownFunction []
0x00007ff7423b94cf AC2-Win64-Shipping.exe!UnknownFunction []
0x00007ff7454f6342 AC2-Win64-Shipping.exe!UnknownFunction []
0x00007ff90f447344 KERNEL32.DLL!UnknownFunction []
0x00007ff90f6426b1 ntdll.dll!UnknownFunction []

```

V takových případech bylo zjištěno, že vina je na straně špatné podoby zasílané zprávy. Například u registrace v případě neurčeného hesla pro ovládání hry byla tato část registrační zprávy úplně vynechána. Bohužel, ačkoliv je délka zprávy definovaná, hra očividně stále s existencí této části správy počítá. Řešením proto bylo přidat pro takové prázdné stringy dva byty 0x00 0x00 definující string s nulovou délkou.

2.3 Popis fungování sdílené paměti

Jak již bylo vysvětleno v části 2.1.3, sdílená paměť funguje na principu sdílených souborů v paměti. Hra v okamžiku startu session prostřednictvím těchto paměťových souborů začne poskytovat aktuální data pro každý grafický krok simulátoru. Pro aplikaci stačí číst tato data v pravidelných intervalech, které budou stejné nebo menší než obnovovací frekvence hry.

2.3.1 Čtení sdílené paměti

Soubory v paměti jsou celkem tři, GRAPHICS_PAGE, PHYSICS_PAGE a STATIC_PAGE. Jejich podoba je definována ve zdrojovém dokumentu [19]. V následující tabulce jsou uvedeny jejich adresy a obsah.

■ Tabulka 2.3 Paměťové soubory

Paměťový soubor	Adresa souboru	Obsah souboru
GRAPHICS_PAGE	Local\acpmf_graphics	Data dostupná v grafickém rozhraní hry
PHYSICS_PAGE	Local\acpmf_physics	Fyzikální data vozu uživatele (telemetrie)
STATIC_PAGE	Local\acpmf_static	Statická data o uživateli a nastavení

Pro čtení sdílené paměti je potřeba přečíst adekvátní počet bytů na příslušné adrese uvedené v tabulce 2.3. Získaná data jsou v binárním formátu a proto je potřeba provést mapování na datovou strukturu způsobem, který příslušný jazyk/framework bude dovolovat.

Situace s definovanými paměťovými soubory je ale ještě o něco složitější. Definice paměťových souborů je totiž určená nejen pro simulátor ACC (Asseto Corsa Competizione), ale je společná také pro další simulátor Asseto Corsa stejného vývojového studia. Tento další simulátor je sice

o něco starší, ale zahrnuje výrazně větší obsah a možnosti, například výrazně širší nabídku typů a tříd vozů. V důsledku toho nutně produkuje více různorodých dat pro sdílenou paměť, která nejsou v rámci ACC použitelná a aplikovatelná (např. pro herní režimy, které sama neobsahuje). Protože Asseto Corsa produkuje více dat, jsou v případě ACC některé hodnoty v souborech sdílené paměti nevyužity (nejsou aktualizované). Tato data se nenacházejí na konci, ale jsou různě rozmístěna v celých paměťových souborech. Proto je nutné i tato neaktualizovaná data mapovat pro zaručení správného čtení paměťového souboru, i když se dále nevyužijí. V popisu dokumentu jsou data neposkytovaná hrou ACC také příslušně zvýrazněna [19].

2.3.2 Formát dat

Paměťové soubory obsahují řadu základních typů známých z většiny programovacích jazyků (int, short, float, char), a případně i jejich pole konstantní velikosti. Poskytovaná data ale nejsou ucelená, některé informace jsou pouze indexy, jehož konkrétní význam se musí dodat až po přečtení souboru. Například intenzita deště je poskytnuta pouze indexem z výčtu hodnot, nebo jako model vozu je uveden pouze jeho identifikátor a příslušný název a další parametry (název, koeficienty pro přepočítání dat,...) se musí dodat. Pro správnou interpretaci dat je tak potřeba několika výčtů a tabulek obsahující potřebné dodatečné informace, které je možné pro tyto účely použít. Takové tabulky jsou také uvedeny v popisujícím dokumentu [19].

2.3.3 Problémy se sdílenou pamětí

Sdílená paměť funguje víceméně bez problému, pokud je správně provedeno mapování a interpretace dat. Právě mapování dat je zde důležité, protože v případě chyby v mapování správného počtu bytů pro konkrétní typ od daného místa do konce souboru se všechna další data stávají naprosto chybná a případně nesrozumitelná.

Jeden z nepříjemných problémů je pozorované chování, kdy v případě závodu pouze proti počítačem řízeným protivníkům (tj. offline session) může uživatel hru pozastavit. V takovém případě hra soubory sdílené paměti nastaví na výchozí hodnoty pro všechny datové typy a přestane je aktualizovat, což může způsobovat problémy pro kontinuální čtení. Pro takové situace je v jednom paměťovém souboru uvedena informace jestli došlo k vyvolání okna nastavení (přerušena hra), a tak lze tuto situaci také zohlednit.

Dalším nalezeným problémem byla špatná informace o tom, že hra ACC neposkytuje data o poškození zavěšení (pole suspensionDamage), které se ukázalo být správně aktualizovaným a použitelným. Může se jednat o jeden z vlivů neaktuální dokumentace

2.4 Zhodnocení poskytovaných dat

Oba způsoby sdílení dat hrou, tak jak jsou jsou výše popsány, jsou dobře použitelné. Ač se svou podstatou od sebe velmi liší, a oba poskytují velké množství využitelných dat. Fungování sdílené paměti je pak o něco jednodušší a spolehlivější. Na druhou stranu neposkytuje všechna data, proto připojení pomocí UDP je také nezbytné realizovat. Protože bude potřeba data různě kombinovat, nabízí se použít stejnou frekvenci aktualizací pro oba typy, aby se předešlo nečekaným a nesprávným stavům dat. Například pokud by se data z jednoho zdroje, již v novějším stavu, zkombinovala s daty z druhého zdroje, stále ve starším stavu.

Srovnáním obsahu definovaných UDP zpráv a souborů sdílené paměti je navíc vidět, že některá data jsou poskytována pomocí obou API zároveň, a tedy duplikovaná. U takových dat je třeba při implementaci zkontrolovat, že jsou opravdu totožná, a v takovém případě preferovat ta poskytnutá sdílenou pamětí. V případě odlišnosti je možné určit vhodnější z nich pro použití v aplikaci.

Rešerše existujících řešení

V této kapitole je uveden přehled již existujících „konkurenčních“ řešení podobné problematiky.

Různých aplikací, které se připojují na závodní hry/simulátory, je větší množství, stejně tak je velké množství různých simulátorů. Je samozřejmě možné najít spoustu veřejně dostupných více či méně zdařilých pokusů o aplikaci zpracovávající telemetrii hry ACC. Spousta z nich jsou ale jednorázové, dále neudržované aplikace, některé pouze ve stádiu pokusu, jiné pouze se základní funkcí. Proto se takové aplikace v této rešerši ani neobjevují.

Z výše popsaných důvodů byla rešerše provedena pouze pro nejpoužívanější podpůrné aplikace, které lze použít pro určený simulátor Asseto Corsa Competizione. Zároveň jsou všechny aplikace vybrané k rešerši v době psaní práce dostupné a použitelné.

Sledovými hledisky u této rešerše budou základní vlastnosti (dostupnost, licencování, cena, instalace, podpora češtiny, ...), způsob připojení na hru, vzhled uživatelské rozhraní a jeho použitelnost, zobrazovaná data a grafy, možnosti predikce dat, a konečně možnost exportu a sdílení dat.

3.1 ACC Tools

ACC tools je aplikace od firmy/skupiny Playlogy, která má ambice spojit řadu původně samostatných utilit. Tato aplikace byla vyzkoušena a popsána v uvedené verzi 1.7.0.0. [20]

Základní vlastnosti Aplikace je dostupná jako volně stažitelný instalační balík MSI přímo z webové stránky. Díky použití MSI balíku je instalace jednoduchá a nenastaly při ní ani při spuštění žádné problémy. Aplikace podporuje pouze angličtinu, alespoň pak její testovaná verze.

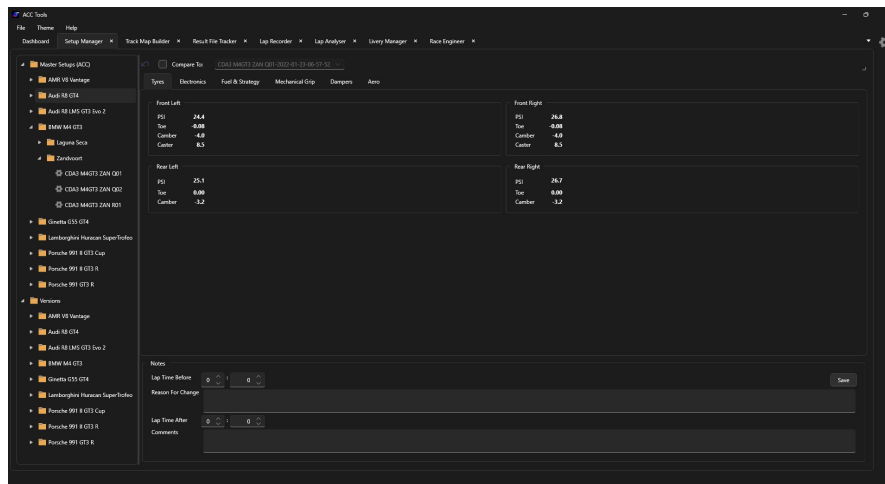
Připojení na hru Aplikace se na hru připojuje oběma dostupnými způsoby, tedy jak pomocí sdílené paměti, tak pomocí UDP připojení. Aplikace ale opakuje chybu, která je uvedena už ve vzorové aplikaci přibalené k dedikovanému serveru (viz 2.2.1), a která způsobuje že nedojde k ukončení spojení, které tak hra udržuje a dále posílá data.

Uživatelské rozhraní a použitelnost Grafické rozhraní je čisté a přehledné, jednotlivé nástroje jsou členěny do záložek. Je zde podpora řady grafických témat, rozmístění informací a ovládání je vhodně zvoleno. K vytknutí je pouze možnost snadno zavřít záložky křížkem. Pokud se o stane, pak se záložka musí znovu otevírat z ne úplně intuitivně umístěné nabídky.

Zobrazovaná data a grafy Aplikace cílí spíše na podporu před závodem a po závodě. Z toho důvodu obsahuje především nástroje pro vyhodnocení zaznamenaných dat, jako jsou výsledkové soubory ze závodů (které hra sama generuje), nebo zaznamenané časy kol. Tato zaznamenaná data lze dobře porovnávat, a to i s pomocí grafů. Dalšími dostupnými nástroji jsou ty, které pomáhají organizovat uživatelské soubory hry, což jsou především uložená nastavení vozů nebo vlastní vytvořené potisky vozů. Posledním nástrojem je kalkulačtor pro správné nastavení tlaků a teplot pneumatik a paliva pro start závodu.

Predikce dat Vzhledem ke svému zaměření na vyhodnocení dat, případně podpoře pro správné nastavení vozu na startu, neobsahuje aplikace žádné predikce.

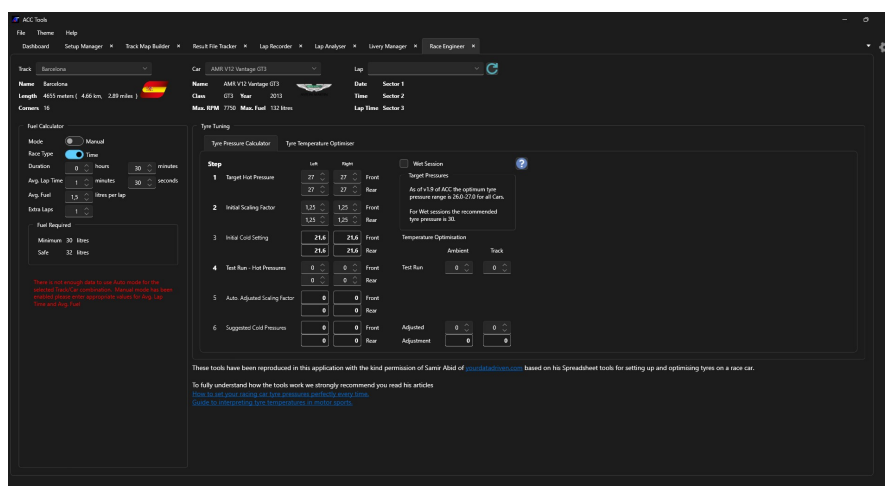
Export a sdílení dat Aplikace nepodporuje žádné sdílení dat na další zařízení. Export sice umožňuje, ale na výběr je pouze export do formátu CSV, přičemž se exportují pouze základní údaje o nejrychlejších kolech v různých sezeních (závodech) a nikoli samotná telemetrická data.



■ Obrázek 3.1 ACC tools – příklad manažera pro nastavení vozů

Date/Time	Type	Track	Race Day	Start Time	Duration	Total Laps	Player Count	Player List	Invalid Laps	Average Lap Time	Best Lap	Best Sector 1	Best Sector 2	Best Sector 3	Finish Position
01.03.2024	Race	Unknown	Sunday	00:00	00:00	1	1	00:19:305	00:00:000	01:23:647	01:23:647	01:23:647	01:23:647	01:23:647	1
23.06.2023	Practice	Unknown	Friday	00:00	00:00	5	2	00:01:081	01:04:340	00:37:527	01:04:389	00:37:833	01:04:389	00:37:833	1
23.06.2023	Race	Unknown	Sunday	00:00	00:00	26	15	01:34:079	01:27:039	00:38:336	00:32:231	00:36:078	00:36:078	00:36:078	15
24.03.2020	Race	Unknown	Sunday	00:00	00:00	27	9	01:52:257	01:56:793	00:28:331	00:22:290	00:26:075	00:26:075	00:26:075	16
24.03.2020	Qualifying	Unknown	Saturday	00:00	00:00	5	2	01:15:153	01:26:472	00:38:080	00:22:347	00:25:266	00:25:266	00:25:266	4
24.03.2020	Qualifying	Unknown	Saturday	00:00	00:00	5	3	01:55:330	01:28:077	00:38:561	00:22:884	00:26:600	00:26:600	00:26:600	13

■ Obrázek 3.2 ACC tools – příklad zobrazení výsledkového souboru



■ Obrázek 3.3 ACC tools – příklad závodního inženýra - kalkulačka tlaků

3.2 Telemetry Tool for ACC

Telemetry Tool for ACC je aplikace podporující připojení více různých simulátorů, např. série F1 her, Project Cars 2 a další, přičemž právě ACC je také jedním z podporovaných. Podle informací uvedených na zdrojové stránce je autorem Iko Rein a aplikace je jeho osobní projekt, který vyvíjí a vylepšuje již řadu let. Z důvodu placeného přístupu nebyla aplikace osobně vyzkoušena, ale rešerše byla provedena především na základě volně dostupných informací na uvedeném webu a z dostupných video návodů k používání aplikace. [21, 22, 23]

Základní vlastnosti Aplikace je dostupná na stránkách Overtake Race Department [21]. Pro její stažení musí být uživatel registrován na uvedeném webu, přičemž registrace je možná pouze s výběrem jednoho z několika druhů předplatného. Předplatné je možné platit měsíčně (2,99 €), čtvrtletně (7,99 €), půlročně (9,99 €) nebo ročně (11,99 €). Instalace by měla být jednoduchá a prováděná pouze rozbalením zip souboru a specifikací cesty ke konfiguračním souborům.

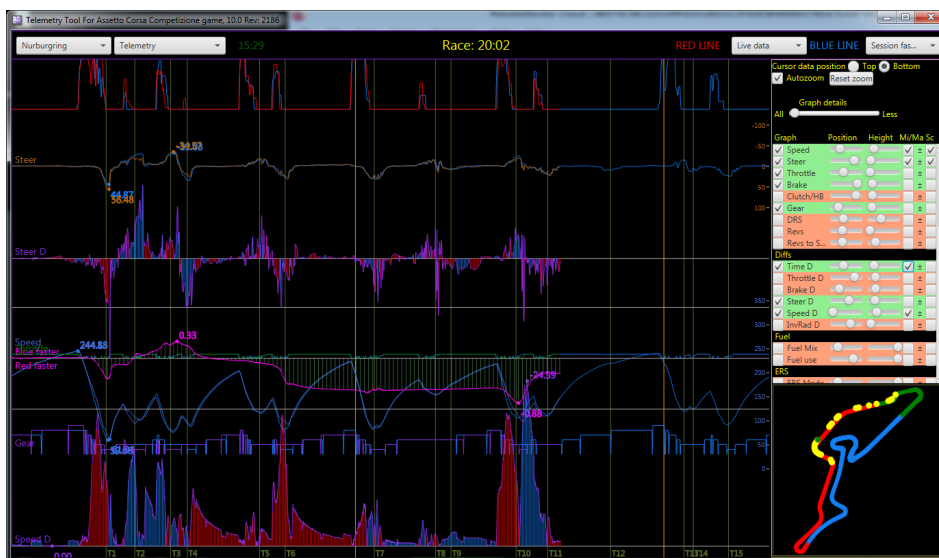
Připojení na hru Připojení je zajištěno jak pomocí sdílené paměti, tak UDP připojením a může tak využít všechna poskytovaná data. Pro připojení na hru ACC uživatel musí ručně nastartovat čtení sdílené paměti pomocí přiloženého bat souboru, a po ukončení použití také nezapomenout zastavit. To je z uživatelského hlediska velmi nepohodlné a autor pro toto dokonce musel vytvořit vysvětlující video [22].

Uživatelské rozhraní a použitelnost Uživatelské rozhraní je vytvořeno se strohou a jednoduchou grafikou. Informace jsou rozděleny do oken, které se mohou různě umisťovat, ukládat jejich pozice, a další okna s některými detaily lze dále vyvolat. Množství oken a zobrazených dat ve spojení s malým fontem může ztěžovat čitelnost, zvláště pokud by měl informace sledovat sám jezdec během závodu.

Zobrazovaná data a grafy Aplikace zobrazuje dosažená kola, mapu okruhu s pozicí oponentů, telemetrická data v číselně i v grafech, a navíc i statistiky sezení, jako jsou průměrné hodnoty časů a hodnot telemetrie. Některé funkce ale nejsou dostupné při použití s ACC, protože neposkytuje pro ně potřebná data, například srovnání telemetrie s oponenty (telemetrie je u ACC dostupná pouze pro vůz uživatele).

Predikce dat V okně věnovaném zastávce v boxu je sice ukázána časová ztráta na zastávku, ale jedná se o pevně stanovenou hodnotu, kterou si uživatelé mohou upravit pro každou trať. Takový přístup se určitě nedá počítat mezi predikce. Všechny ostatní budoucí hodnoty, které tato aplikace zobrazuje jsou ty, které hra poskytuje sama mezi ostatními daty (např. na kolik kol zbývá palivo). Proto je konstatováno, že aplikace nenabízí žádnou vlastní možnost predikce dat.

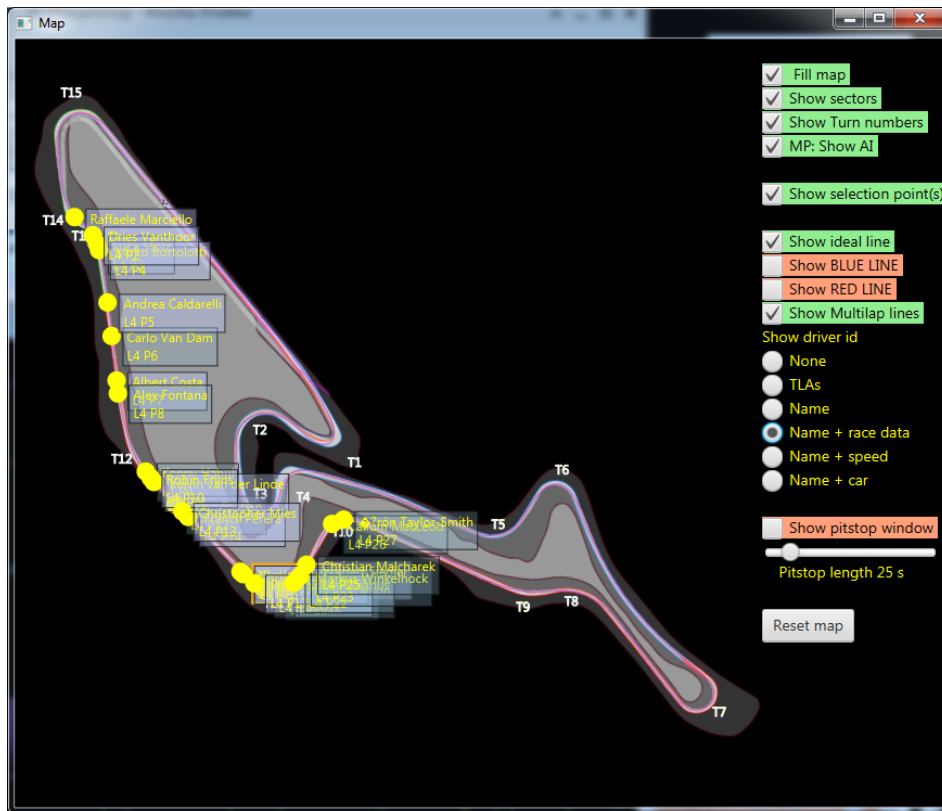
Export a sdílení dat Export dat je možný, ale pouze pro předdefinované struktury dat, které poskytují jen omezené možnosti exportu. Aplikace umožňuje sdílet data z hry po síti, proto není problém používat aplikaci pro sledování telemetrie jiným člověkem na jiném počítači.



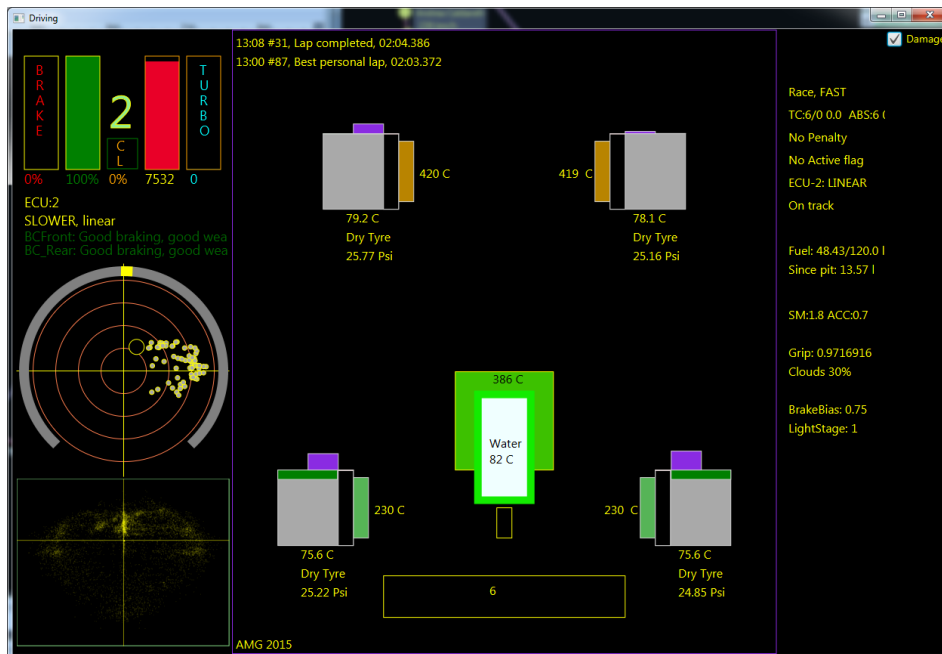
■ Obrázek 3.4 Telemetry Tool for ACC – grafy telemetrie [21]



■ Obrázek 3.5 Telemetry Tool for ACC – základní data [21]



■ Obrázek 3.6 Telemetry Tool for ACC – mapa okruhu [21]



■ Obrázek 3.7 Telemetry Tool for ACC – telemetrie kol [21]

3.3 Sim Racing Telemetry

Sim Racing Telemetry, za níž stojí *UNAMedia*, podporuje nemalé množství simulátorů (ACC, série F1, Project Cars2, ...). Nevýhodou je, že se jedná o placenou aplikaci, u které se navíc platí zvláště za každý plugin, který odemká jeden z podporovaných simulátorů. Zdarma je pouze základní verze, která poskytuje možnost vyzkoušet si omezenou (trial) variantu, a která by se měla lišit jenom v rozsahu dostupných dat pro záznam a analýzu. Proto aplikace nebyla přímo vyzkoušena ve své plné podobě, ale byla využita zmíněná zkušební varianta ve verzi 1.14.0.

Základní vlastnosti Jak základní aplikace, tak její zkušební varianta je dostupná přes službu Steam [24]. Zároveň má také vlastní webové stránky [25]. Pro své plné využití je nezbytné pořídit doplněk pro ACC, který odemkne plný potenciál aplikace (každý za 9,99 €). Není problém s instalací a během na systému Windows, ačkoliv první pokus o instalaci zastavil antivirový program pro podezření z instalace viru. Při testování také nastala situace, kdy kvůli této aplikaci zamrzl celý počítač a bylo nutné provést „tvrdý“ restart. Podporovanými jazyky jsou pouze angličtina a italština.

Připojení na hru Aplikace se na hru připojuje pouze pomocí sdílené paměti. Připojení se realizuje čistě tlačítkem, za kterým se skrývá další tlačítko pro potvrzení, což je ještě relativně v pořádku. Při zkoušení nastala ale i situace, kdy se aplikace odmítala na hru připojit, ačkoliv hra byla spuštěná.

Uživatelské rozhraní a použitelnost Samotná aplikace vypadá po grafické stránce dobře, je responzivní, jednoduchá, přehledná a působí moderně. Rozvržení některých ovládacích prvků ale působí dojmem, jako by bylo spíše uzpůsobeno pro mobilní zařízení.

Zobrazovaná data a grafy Protože se aplikace na hru připojuje pouze pomocí sdílené paměti, má k dispozici pouze data omezená na uživatele vůz. Zaměření dat je tak pouze a čistě na telemetrii vozu bez jakéhokoliv řešení soupeřů.

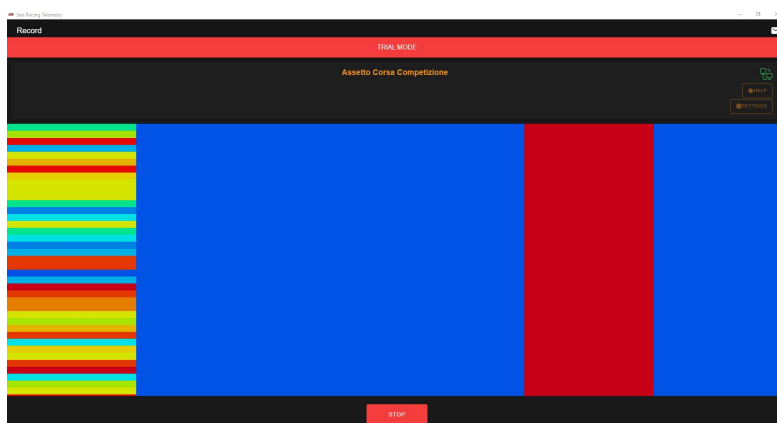
Aplikace při čtení dat, tedy v průběhu samotné jízdy, zobrazuje pouze barevné pruhy (viz 3.8). Není známo jestli je to záměr nebo chyba, případně jestli se to týká pouze zkušební verze. Výsledkem nicméně je aplikace nepoužitelná v průběhu závodu, a užitečná pouze pro zpětné vyhodnocování dat.

Po ukončení čtení jsou všechna data dostupná jako samostatná kola – víceméně se jedná o záznam všech možných dat, která jsou následně roztržena do kol (viz 3.9). Pro každé kolo jsou v základu k dispozici grafy přetížení (g-force), rychlosti, otáček a převodového stupně, případně lze zvolit další z dostupných dat (viz 3.10). Velmi pěkná je dostupná mapa okruhu, kde lze s pomocí barevné škály zobrazit rychlost, přetížení a sešlápnutí plynu nebo brzdy (viz 3.11). Po výběru konkrétního místa mapy jsou tato data navíc dostupná číselně a v další záložce lze najít i všechna ostatní data, která ACC poskytuje. Pro každé kolo lze také zobrazit statistiky minimálních, maximálních, průměrných hodnot dostupných veličin.

Relativně šikovná funkce je, že lze data daného kola porovnávat s jiným zaznamenaným kolem.

Predikce dat Aplikace pouze zobrazuje zachycená data a neposkytuje žádné predikce.

Export a sdílení dat Export dat je možný, a to pouze do souboru CSV. Aplikace také umožňuje sdílení dat po síti na jiný počítač.



■ **Obrázek 3.8** Sim Racing Telemetry – zobrazení při záznamu dat

Lap	Pos	Lap Time	Sectors
1 /	→ 1	3:25.918	PIT 1:53.311 34.445 58.162
2 /	→ 1	2:06.975	36.675 36.602 55.298

■ **Obrázek 3.9** Sim Racing Telemetry – přehled kol

3.4 ACC Drive

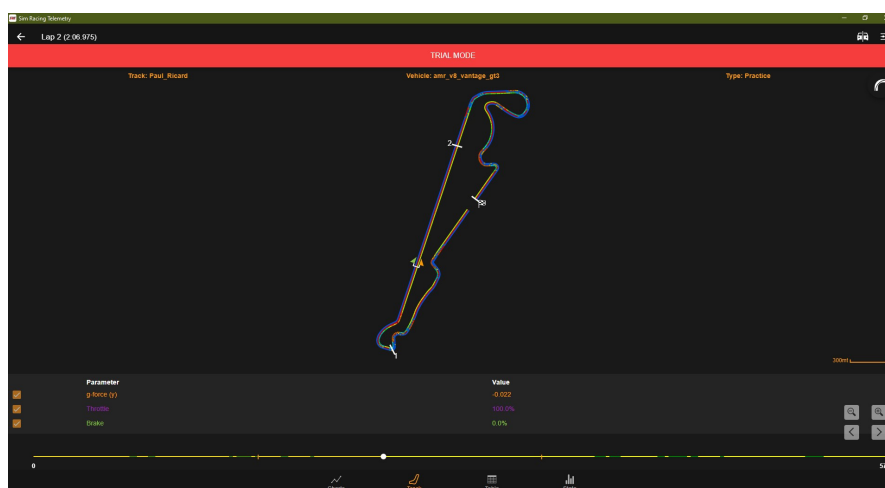
ACC Drive je relativně nová aplikace, jejímž autory jsou Niklas Krause a ACC-Solutions. Funkcionality jsou také představeny v YouTube videu [26] a řadí se mezi ně vizualizace realtime dat a ranking systém pro porovnání. Zajímavou funkcí je ovládání nastavení zastávky v boxech ve hře přímo z aplikace, a predikce pozice po návratu ze zastávky v boxech. Bohužel je aplikace placená a proto také nebyla přímo vyzkoušena. Fungování aplikace je tak shrnuto z volně dostupných informací. [27]

Základní vlastnosti Aplikace je dostupná přes službu STEAM pomocí odkazu na oficiální webové stránce aplikace s inzerovanou cenou 25 €. Podporovaným jazykem je angličtina, informace o podpoře dalších jazyků není známá. [27]

Připojení na hru Aplikace se připojuje jak pomocí sdílené paměti, tak pomocí UDP komunikace, a má tak k dispozici všechna dostupná data.



■ Obrázek 3.10 Sim Racing Telemetry – telemetrie v grafech



■ Obrázek 3.11 Sim Racing Telemetry – zobrazení na mapě

Uživatelské rozhraní a použitelnost Uživatelské rozhraní je zpracováno jednoduše a je použitelné pro zobrazení v malém okně. Využívá záložky, které někdy kombinuje s postranní nabídkou a stavovou lištu. Zobrazení dat je ale často ve formě prostého textu, což snižuje přehlednost a může se stát, že se uživatel v zobrazení ztratí. Aplikace dokonce nabízí vlastní elementy pro překrytí obrazovky (HUD), které lze použít pro zobrazení dat přímo ve hře.

Zobrazovaná data a grafy Poskytován je především přehled nastavení zastávky v boxu s možností její přenastavení. Z telemetrie se aplikace zaměřuje hlavně na tlaky pneumatik, stav paliva a poškození. Ze závodních situací ukazuje přehled dosažených časů, a také informace o pozicích a časech soupeřů a o jejich zastávkách v boxech. V grafech jsou pak zobrazeny pouze tlaky pneumatik v závislosti na čase závodu.

Predikce dat Dle popisu funkcí by aplikace měla umět počítat zbývající čas paliva a predikovat pozici a provoz při výjezdu ze zastávky v boxu. Vzhledem k tomu, že tyto funkce nebyly nikde ukázány, nebylo je možné ověřit.

Export a sdílení dat Sdílení dat je umožněno skrze vlastní mechanismus spojení přes veřejnou síť, kdy se v rámci aplikace tvoří vlastní sezení a jiný uživatel se na ně může připojit, pokud obdrží jeho identifikátor a heslo.

Další funkce Dodatečnou funkcí, která je poměrně unikátní, je možnost úpravy nastavení úkonů pro zastávky v boxu (např. změna množství paliva pro tankování, úpravy tlaků pneumatik pro přezutí, ...). Tuto možnost ale hra nenabízí, proto je tato funkce implementována pomocí simulování řady příkazů pomocí klávesnice, čímž provede úpravu v menu hry. Ve výsledku tedy provede stisky těch kláves v potřebném množství a pořadí, jako kdyby uživatel tyto zásahy provedl sám.



■ Obrázek 3.12 Acc Drive - koláž zobrazení [27]

3.5 Race Element

Aplikace Race Element je dílo vývojáře Reiniera Klarenberga a je určena jak pro ACC, tak pro další simulátory Asseto Corsa a iRacing. Testována byla její verze 1.0.5.4.

Základní vlastnosti Aplikace je dostupná ke stažení, stejně jako Telemetry Tool for ACC (viz 3.2) ze stránky Overtake Race Department [28]. U této aplikace je možnost obejít nutnost mít registrované členství pro umožnění stažení díky tomu, že její repozitář na GitHubu je veřejně dostupný [29] a aplikace volně k použití. Zde je možné stáhnout přímo spustitelný soubor, který stačí po stažení spustit a není tak potřeba aplikaci instalovat. Podporovaným jazykem je pouze angličtina.

Připojení na hru Připojení je realizováno pouze pomocí sdílené paměti a funguje bez potřeby další konfigurace.

Uživatelské rozhraní a použitelnost Uživatelské rozhraní na první pohled vypadá dobře a přehledně. Využívá se vswislá navigace pomocí níž se přepíná mezi jednotlivými obrazovkami s příslušnými funkcemi, které aplikace nabízí. Zároveň je na každé obrazovce další navigace, které jsou jednou vodorovně dole, jednou vodorovně nahoře, jednou jako další řada vswisle, a navíc pokaždé vypadají mírně odlišně. Někde i poněkud splývají se zbytkem zobrazení a uživatel se tak o jejich existenci může dozvědět až později.

Zobrazovaná data a grafy Tato aplikace podporuje, podobně jako ACC Tools (viz 3.1), zobrazení uložených vzhledů vozů a jejich předdefinovaných nastavení, která jsou uložena mezi uživatelskými soubory hry. V rámci aplikace nejsou telemetrická data zobrazována vůbec a to jak čistě hodnoty, tak ani pomocí grafů. Zaměřena je především na vytvoření řady prvků, které lze použít pro překrytí obrazovky hry a které tato data poskytují. Toto má společné s aplikací ACC Drive (viz 3.4), ale u této aplikace je nabídka možných prvků větší.

Při testování a snaze připravit pár prvků pro představu jak fungují, docházelo k tomu, že přepínání oken tuto funkcionalitu dokázalo rozhodit. Výše zmíněné prvky tak skončily například jako překrytí pro na pozadí spuštěný webový prohlížeč, kde se jejich využití limitně blíží nule. Po opravě se tato situace mohla kdykoliv opakovat.

Predikce dat Aplikace neposkytuje žádnou možnost predikcí, pouze s úpravami zobrazuje přijatá data.

Export a sdílení dat Aplikace je určena především k zobrazení aktuálních dat, proto je neukládá a ani neumožňuje jejich export. Sdílení dat také není podporováno, naopak je podporováno streamování vlastního obrazu.



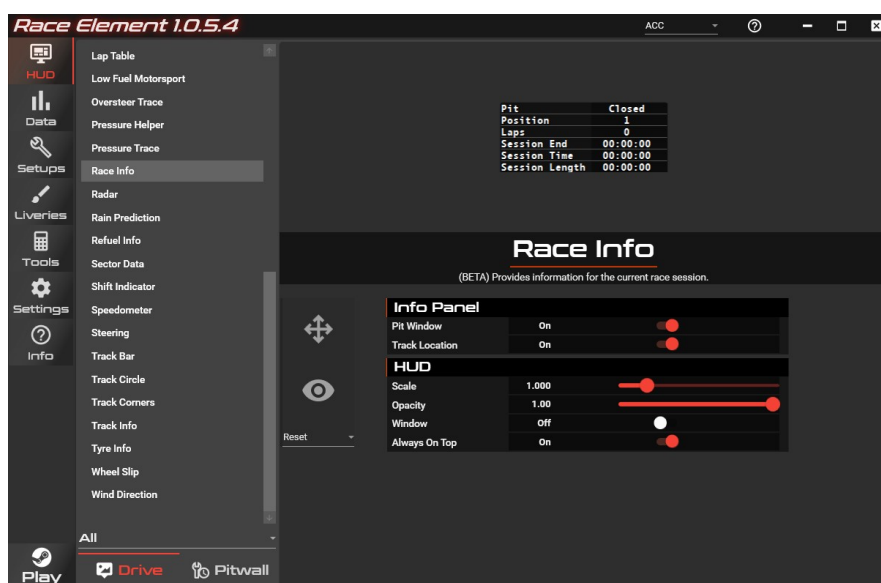
■ Obrázek 3.13 Race Element – přehled nastavení

3.6 Srovnání a zhodnocení

3.6.1 Finální srovnání

Finální srovnání představených aplikací je provedeno podle stejných zkoumaných hledisek.

Základní vlastnosti Všechny představené aplikace jsou dostupné. Telemetry Tool for ACC a Race Element jsou výsledkem osobních projektů samostatných vývojářů, o ACC Tools, ACC Drive a Sim Racing Telemetry se starají celé skupiny lidí. Nicméně pouze ACC Tools byla plně zdarma, Race Element pak bylo možné zdarma získat z repozitáře, ale na oficiálním webu bylo pro stažení nutné mít předplatné. Ostatní aplikace byly zpoplatněny, přičemž cena ACC Drive



■ Obrázek 3.14 Race Element – výběr HUD elementu

(viz 3.4) už může odpovídat ceně samotné hry ACC v některé z častých slev (běžná cena 39,99 €, slevy bývají i 40%).

Připojení na hru Pouze aplikace Sim Racing Telemetry a Race Element získávají data pouze pomocí sdílené paměti. Všechny ostatní aplikace využívají oba způsoby, tedy navíc i UDP připojení. ACC Tools navíc obsahuje chybu v ukončování spojení, což může působit potíže při používání.

Uživatelské rozhraní a použitelnost U žádné aplikace nelze říci, že by její uživatelské rozhraní bylo nepoužitelné. Nicméně se zpracování u každé aplikace výrazně liší a všechny jsou svým způsobem originální. U některých aplikací byly sice vytknuty některé konkrétní nedostatky, ale hodnocení zde bude záviset spíše na osobních preferencích konkrétních uživatelů.

Zobrazovaná data a grafy Zde se zkoumané aplikace liší nejvíce. ACC Tools a Sim Racing Telemetry zobrazují data pouze zpětně ve formě jejich analýzy, kdežto Telemetry Tool for ACC, ACC Drive zobrazují data i v průběhu závodu, Race Element pak dokonce jenom ve formě vlastních HUD elementů a nikoliv v aplikaci. ACC Tools a Race Element navíc umožňují zobrazovat uživatelské soubory, ve kterých se nachází uložená nastavení a vlastní vzhledy vozů.

Predikce dat Žádná z aplikací vyloženě predikce neposkytuje. Jistá, ač limitovaná možnost je zde pouze u ACC Drive, která avizuje predikce zbývajících času paliva a pozice návratu po návratu ze zastávky v boxu. S ohledem na nejvyšší cenu ze všech zkoumaných aplikací ale nebylo ověřeno zda a jakým způsobem predikce poskytuje.

Export a sdílení dat Pouze ACC Tools, Sim Racing Telemetry a Telemetry Tool for ACC umožňují export dat a to do formátu CSV. Ostatní aplikace žádnou možnost exportu dat neposkytují. ACC Tools a Race Element nepodporují žádnou možnost sdílení dat na jiný počítač. Telemetry Tool for ACC a Sim Racing Telemetry pak umožňují sdílení po síti a ACC Drive má na sdílení vlastní mechanismus.

3.6.2 Vyhodnocení řešerše

Srovnání aplikací z řešerše je v tabulce 3.1 přehledně zobrazeno vůči požadavkům tak, jak jsou definovány později v částech 5.1 a 5.2

■ **Tabulka 3.1** Plnění požadavků existujícími řešeními

Požadavek	ACC Tools	Telemetry Tool for ACC	Sim Racing Telemetry	ACC Drive	Race Element
F1 - Zobrazení realtime dat	NE	ANO	NE	ANO	ANO
F2 - Zobrazení grafů	ANO	ANO	ANO	ANO	NE
F3 - Predikce dat	NE	NE	NE	Omezené	NE
F4 - Export dat	ANO	ANO	ANO	NE	NE
F5 - Sdílení dat	NE	ANO	ANO	ANO	NE
N1 - Využití prostředků	ANO	ANO	ANO	ANO	ANO
N2 - Podpora Windows	ANO	ANO	ANO	ANO	ANO
N3 - Windows Store distribuce	NE	NE	Steam	Steam	NE
N4 - Uživatelské rozhraní	Výtka	Výtka	ANO	Výtka	NE
N5 - Licencování (zdarma)	ANO	NE	NE	NE	ANO
N6 - Podpora češtiny	NE	NE	NE	NE	NE

Ze srovnání existujících řešení je patrné, že pokud bychom chtěli kvalitní aplikaci, která by byla rozumně použitelná a se subjektivně lepším grafickým prostředím, je třeba využít některé z placených. Cena je ale jedním z požadavků, navíc pokud má o aplikaci zájem celá komunita, byla by výsledná cena značná.

Z tabulky je rovněž patrné, že žádné z již existujících popsanych řešení nesplňuje všechny definované požadavky na aplikaci. Proto lze konstatovat, že vývoj nové aplikace je na místě a bude, minimálně pro komunitu, přínosem.

Výpočty a predikce

V této kapitole jsou krátce shrnuty možnosti a způsoby predikcí. Dále jsou popsány zvolené predikce, jejich význam a způsoby výpočtu.

4.1 Stručný úvod k predikcím

Predikce jako pojem je předpověď nebo prognóza budoucí události nebo konkrétního výsledku. Není založena na náhodě, ale naopak vychází ze znalosti současného stavu a/nebo znalosti získané studiem trendů a vzorů vývoje v minulosti. Existuje mnoho typů predikcí, například bodová predikce (tj. odhad jednoho výsledku), krátkodobé či dlouhodobé predikce pravděpodobnosti výskytu nějaké události či nějakého trendu hodnot. Mezi typy predikcí ale patří i deterministické predikce, které jsou založené na jednoznačném a určitém výsledku na základě známé informace, například výpočtem z matematické rovnice. Současně existuje i množství metod pro tvorbu těchto predikcí. Může se jednat o dohad experta na danou problematiku, metody využívající statistiku historických dat, metody simulující budoucí vývoj, metody strojového učení, metody využívající neuronové sítě a další. [30]

Interpretací dat s cílem získat informované predikce pomocí statistiky se zabývá oblast datové vědy s názvem Prediktivní analýza. Ta využívá řadu statistických technik ke sběru a vytěžování dat (tj. hledání vzorů a závislostí ve velkém souboru dat) a k prediktivnímu modelování, které produkuje výsledný statistický model používaný k vytváření predikcí. Základní typy modelů se dají shrnout do několika skupin:

- Regrese – zjišťují závislosti mezi proměnnými (např. přímková - lineární regrese)
- Klasifikace – stanovují charakteristiky množiny dat a podle nich kategorizují nová data
- Seskupování – seskupují data podle podobností do skupin úzce souvisejících dat
- Kombinace výše zmíněných typů – potřebné pro komplexní problémy

Pro konkrétní situaci je potřeba vybrat správný typ modelu a připravit historická data pro jejich analýzu a jeho vybudování. Především se pro vybudování takového modelu využívá strojového učení. [31, 32, 33]

Hlavním problémem při vytváření modelů je získání kompletní sady potřebných dat, dále pak případná chybovost některých dat v získané sadě. Zároveň je nutné dbát na bezpečnost dat, zvláště pokud obsahují citlivé informace. [33]

4.2 Výběr predikcí

Predikce jsou jedním z hlavních přínosů této aplikace. Samotný rozsah možných predikcí je u závodů ale poměrně velký. Lze například předpovídat vývoj počasí, předpovídat vývoj dat telemetrie (například tlaku pneumatik, opotřebení brzd, a další) včetně případného zohlednění jejich vzájemného ovlivňování. Zároveň některé z těchto predikcí jsou užitečnější pro jezdce, jiné pro závodního inženýra.

Při výběru se mimo jiné musí se brát na zřetel, že dostupný výpočetní výkon je velmi omezený a i když není vyloženo nutné přepočítávat predikce v reálném čase, výpočet by neměl trvat více než pár sekund. Pokud by výpočet trval déle, mohlo by se stát, že jeho výsledky už budou v okamžiku doručení uživateli zastaralé a nepotřebné.

Pro účely práce byly proto vybrány spíše jednodušší predikce, které jsou užitečné především pro samotného jezdce v závodě. Jedná se především o deterministické predikce využívající k nim definované výpočty, případně jednoduché regrese, jako je například lineární regrese proložením přímkou. Jejich výčet je uveden níže, konkrétnější popis pak dále, kdy je každé z nich věnována pozornost samostatně.

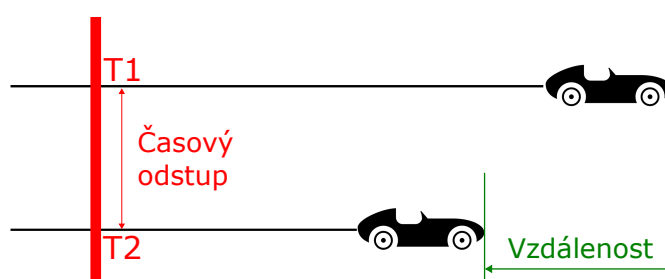
- Časový odhad zbývajících paliva
- Předpověď souboje na trati
- Předpověď ztraceného času a pozic při zastávce v boxu
- Jednoduchá predikce časových řad různých telemetrických dat

4.2.1 Výpočet časových odstupů

Výpočet časových odstupů sice není prediktivní výpočet, ale řada dále uvedených výpočtů odstupů potřebuje. Simulátor tyto údaje ve svém vlastním uživatelském rozhraní zobrazuje, ale poskytuje je pouze pro online závody a to navíc pouze pro nejbližší oponenty před a za jezdce. Z tohoto důvodu musí být časové odstupy počítány v aplikaci a i proto je jejich výpočet také uveden.

4.2.1.1 Význam

Časový odstup (někdy také interval) představuje časovou mezeru mezi dvěma jezdci. Oproti prosté vzdálenosti má tu výhodu, že změna rychlosti vozů v průběhu kola tento odstup neovlivňuje. Navíc časový údaj je právě to, co je zajímavé (například dále u zastávky v boxu), samotná vzdálenost jezdci nic významného neřekne. Grafické znázornění časových odstupů je ukázáno na obrázku 4.1, kde T_1 a T_2 jsou časy, ve kterých se příslušné vozy nacházely na konkrétním místě trati. [34]



■ Obrázek 4.1 Ilustrace vzdálenosti a časového odstupu

4.2.1.2 Řešení

U všech vozů simulátor dodává informace o dokončených kolech `laps` (aktuální je teda další v pořadí) a aktuální poloze na trati `spline_location` jako hodnotu mezi 0–1 (začátek kola – konec kola). Pokud se k tomuto přidá informace o délce trati `track_length`, lze snadno spočítat odstup ve vzdálenosti podle vztahu 4.1.

$$l_{dif} = track_l * (laps_j + spline_j - laps_o - spline_o) \quad (4.1)$$

S časovým odstupem je to o něco složitější. Jak už bylo naznačeno výše, pro přesný údaj o časovém odstupě je nutné pravidelně zaznamenávat časy průjezdu jednotlivých vozů u nějakého pevného bodu na trati. Rozdíly v časech průjezdů dvou jezdců u posledního takového bodu pak dávají jejich aktuální časový odstup. Pevným bodem může být například konec kola (každý průjezd cílovou čarou), což ale znamená, že je nutno čekat celé kolo na další výpočet odstupů. Z toho důvodu bylo zvoleno rozdělení kola na 10 takových bodů – `mini-úsektor`, které umožní dostatečnou jemnost přepočtů časových odstupů v průběhu kola a nezatíží tím příliš aplikaci. Časy průjezdů všech vozů v mini-sektorech je nutné zaznamenat a odstupů následně spočítat odečtením časů prvního a druhého vozu.

$$t_{dif} = t_{vuz1} - t_{vuz2} \quad (4.2)$$

Pro snížení náročnosti stačí časové odstupů počítány pouze vzhledem k vozu uživatele aplikace, jiné nejsou aktuálně zajímavé.

4.2.1.3 Alternativa

Prakticky lze získat hrubý odhad časového odstupů také prostým podělením vzdálenostního odstupů aktuální rychlostí v_{act} . Takový výsledek ale příliš závisí na použité aktuální rychlosti. Správný výsledek by totiž byl získán při použití průměrné rychlosti v celém aktuálním úseku trati mezi jezdci. Na druhou stranu, tento hrubý přístup by mohl být vhodný u malých odstupů (například zhruba do 5 s), kdy se průměrná rychlost může blížit rychlosti aktuální. Výhodou tohoto přístupu je pak to, že je možné odstup počítat kdykoliv a není třeba čekat na další bod na trati, kde se měření časového odstupů provede.

$$t_{dif} = \frac{l_{dif}}{v_{act}} \quad (4.3)$$

4.2.2 Časový odhad zbývajících paliva

4.2.2.1 Význam

Mezi daty aplikace se mimo jiné posílá údaj o předpokládaném zbývajícím počtu kol, na které vystačí aktuální množství paliva. Protože jsou závody v simulátoru často ohraničené časově a nikoli počtem kol, jezdec tak sice zjistí, že mu palivo vystačí například na dalších 42 kol, ale zároveň vidí, že do konce závodu zbývá 2h 15min 43s. Stejně je to i v případě pevně stanovených oken pro zastávku v boxu – jejich otevření bývá také určeno nějakým časem v průběhu závodu.

Proto je jezdcům potřeba zbývajících palivo přepočítat do časového údaje. Následně jezdec může snadno porovnat zbývajících čas závodu a zbývajících čas paliva a vidět, jestli je vše v pořádku, nebo chybí pouze drobné množství a je možné s palivem trochu šetřit stylem jízdy, nebo paliva chybí příliš a je potřeba zajet do boxu dotankovat.

4.2.2.2 Řešení

Jedná se o relativně jednoduchý výpočet, který je potřeba s ohledem na prezentovanou časovou hodnotu pravidelně aktualizovat. Se znalostí údaje o zbývajícím počtu kol, který dodává přímo hra, je potřeba pro tato následující kola znát budoucí časy na kolo. Pokud by budoucí časy byly známé, stačilo by vzít predikovaný počet kol a pouze sečíst časy na kolo, které v nich budou dosaženy. Protože z logiky budoucí časy známé nejsou, je zde několik možností, které časy na kolo lze použít.

- Průměrný čas na kolo (kompletní / klouzavý)
- Extrapolace časů na kolo
- Komplexní predikce časů na kolo

Nejspíše lze hned vyloučit komplexní predikci, i když by měla být nejpřesnější, protože časy na kolo ovlivňuje spousta okolností. Časy totiž může ovlivnit výrazně například stav pneumatik, stav trati, déšť, teplota, hmotnost paliva v nádrži, ale také nečekané situace na trati jako souboje s ostatními závodníky, nehody nebo žluté vlajky. Taková simulace i s případným zohledněním variant nečekaných situací je sama o sobě složitá a náročná, jak z hlediska výpočetních prostředků, tak přípravy modelů. Nesmí se zapomínat ani na to, že aktuálně vytvářená aplikace má být jednoduchá a výkonově nenáročná.

Extrapolace i použití průměru je relativně jednoduchá varianta, jak předpovědět budoucí časy na kolo. V obou případech je na místě použití pouze určitého počtu posledních kol, aby předpovědi neovlivňovaly závodní kola ze vzdálenější historie zasetá za úplně odlišných podmínek. Takto také výrazně odlišná kola budou po chvíli vyloučena z výpočtu a nebudou zkreslovat výsledky dále v budoucnu.

V rámci zachování jednoduchosti je v tomto případě zvolena varianta s využitím průměrných časů na kolo z posledních 5 kol. Výpočet je pak poměrně jednoduchý a pokud k určíme jako zbývajícím počet kol a $t_1..n$ jako časy posledních n kol, je výsledný zbývajícím čas:

$$t = \frac{\sum_{i=1}^n t_i}{n} * k \quad (4.4)$$

Po ověření při uživatelském testování je možné, že výsledky nebudou úplně nejpřesnější. Výpočet by pak bylo možné upravit zavedením buď nějaké formy váženého průměru, nezahrnutím nejhoršího času, úpravou počtu průměrovaných kol, nebo případně opravdu použít nějakou formu extrapolace.

4.2.3 Predikce soubojů na trati

4.2.3.1 Význam

V závodě je důležité mít přehled o soupeřích, kteří jsou v pořadí bezprostředně před a za závodníkem. Jezdec se obvykle snaží dojet a předjet toho před sebou a naopak ubránit se útokům toho za sebou a pokud možno se mu vzdálit. Srovnáním s dalšími jezdci také může získat povědomí o tom, jestli je jeho závodní tempo (tj. dosahované časy na kolo) dostatečné. Pokud je tempo nedostatečné, znamená to většinou, že ho dojíždí soupeř za ním a nebo nedojíždí soupeře před sebou. Toto lze snadno zobrazit časovým vývojem hodnot časových odstupů k soupeřům před, respektive za jezdce.

Tato informace sice nemá příliš velký význam v případě dynamického vývoje situace (start závodu, aktuálně probíhající souboje, ...), ale pro ustálené pořadí s většími rozestupy (což bývá po většinu závodu) je užitečná. Predikce by měla nastoupit v případě, kdy jezdec potřebuje vědět za jak dlouho dojede soupeře před sebou, nebo kdy jej dojede soupeř za ním (případně se může stát oboje zároveň a nebo naopak nehrozí ani jedno). Díky tomu jezdec snadno uvidí, jestli tyto situace nastanou například do konce závodu nebo do naplánovaného pitstopu.

4.2.3.2 Řešení

Změny odstupů se mění podle rozdílu časů na kolo, kterých oba jezdci dosáhly. Časy jsou ale ovlivněny situací na trati, chybami které jezdci dělají, hmotností vozidla, případně jeho poškozením. Navíc samotné rozestupy se mění výrazněji i v průběhu kola, což by samotné časy na kolo nebyly schopné rozlišit. Základní premisou zde je, že se oba jezdci snaží jezdit stále přibližně stejné časy¹ a tedy se hodnota časových odstupů bude měnit především lineárně. V tomto případě se pak jako nejschůdnější jeví proložení hodnot časových odstupů přímkou a určení hodnot z ní. Výsledkem by měly být počet kol, za jak dlouho dosáhne časový odstup nulové hodnoty, tj proložená přímka protne 0. Případně je možné určit jako limit určitý čas odstopu, kdy už reálně začíná souboj mezi jezdci, například 0,3 s.

Časové odstupy se také mohou v jenom kole zmenšit a v dalším narůst, ale i v rámci kola se mohou různit podle aktuálního místa na trati. Při použití lineární extrapolace v případě přesného střídání růstu a zmenšení časového odstopu o stejnou hodnotu bude výsledkem přesně konstantní hodnota rozestupu a správná indikace, že k dojetí nedojde. Jako základ pro extrapolaci se budou brát hodnoty během posledních 5 kol podobně jako v případě výpočtu predikcí paliva (viz 4.4).

4.2.4 Predikce pitstopu

4.2.4.1 Význam

Zastávka v boxu je situace, která znamená významnou časovou ztrátu nutnou pro dotankování, výměnu kol, různé opravy apod., a proto často způsobuje změnu pořadí závodního pole.

Jezdci také vždy nejezdí do boxu naráz a proto se dva, kteří spolu soupeří, mohou vlivem zastávky absolvované jedním z jezdců dostat daleko od sebe. V takovém případě si jezdec, který má aktuálně náskok, musí hlídat časový odstup od soupeře, který už zastávku absolvoval, aby se po své vlastní zastávce vrátil na trať před ním.

Jezdec také může mít takový náskok, že by se v případě absolvování zastávky vrátil doprostřed nebo těsně za skupinu pomalejších jezdců, i kdyby třeba byli o kolo zpět. Takový návrat na trať pro jezdce znamená, že bude pomalejšími soupeři brzděn do té doby, než je zvládne předjet. Tím může ztratit drahocenný čas vůči svým aktuálním soupeřům. Zároveň hustší provoz může také znamenat větší riziko nehody.

Pokud by jezdec v okamžiku, kdy plánuje svoji zastávku v boxu, měl nějakým způsobem dostupné informace o pravděpodobné situaci při návratu na trať, mohl by okamžik zastávky a případně její parametry pozměnit. Například by mohl zastávku pozdržet a snažit se zvětšit svůj náskok aby se při návratu dostal před svého soupeře, případně aby se vyhnul hustší skupině pomalejších jezdců. Další možností je, že by mohl upravit parametry zastávky – tankovat méně paliva nebo odložit výměnu brzd na další zastávku pro rychlejší odbavení zastávky a tím pádem snížení časové ztráty.

4.2.4.2 Řešení

Řešení časové zastávky v popsaném rozsahu je komplexnější problém, u kterého lze využít pouze základní data ze simulátoru. Lze jej rozdělit do několika částí, které jsou popsány zvlášť a ve finále je stačí pouze zkombinovat. Pro snadnou orientaci je potřeba vypočítat časté varianty nastavení zastávky. Tyto varianty pak může jezdec vedle sebe snadno porovnat.

- Pouze průjezd
- Tankování
- Tankování + opravy

¹O to se skutečně ve vytrvalostních závodech jezdci snaží.

- Tankování + opravy + přezutí
- Tankování + opravy + přezutí + výměna brzd

Výpočet časové ztráty zastávky U tohoto výpočtu je problém v tom, že simulátor poskytuje pouze informace ohledně nastaveného množství paliva pro zastávku `mfdFuelToAdd` a velikosti poškození různých částí vozu. Veškeré další informace je potřeba získat a použít pouze jako statická data konkrétního vozidla nebo tratě. Takovými daty jsou:

- Časová ztráta průjezdu boxovou uličkou
- Časová ztráta na zvednutí a položení vozu ²
- Čas tankování na 1l paliva
- Čas opravy poškození na jednotku rozsahu poškození
- Čas výměny pneumatik
- Čas výměny brzd

Časy zvednutí vozu, výměny pneumatik t_{vp} a brzd t_{vb} jsou konstantní, proto je stačí pouze přičíst.

$$t_1 = t_{vp} + t_{vb} \quad (4.5)$$

U tankování a oprav poškození je známá informace o množství paliva m_{mp} k dotankování a rozsahu poškození m_{rpk} a zavěšení m_{rpz} . Rychlosti tankování r_{tp} i oprav poškození karoserie r_{ok} a zavěšení r_{oz} jsou konstantní, proto pro získání času tankování t_{tp} a oprav karoserie t_{ok} a zavěšení t_{oz} stačí jednoduché vynásobení.

$$t_2 = t_{tp} + t_{ok} + t_{oz} = m_{mp} * r_{tp} + m_{rpk} * r_{ok} + m_{rpz} * r_{oz} \quad (4.6)$$

Časová ztráta průjezdu boxovou uličkou je na druhou stranu poměrně komplikovaná. Sice jsou délka boxové uličky a maximální povolená rychlost při jejím průjezdu také konstantní (čímž by šlo snadno získat čas pro průjezd), ale pro získání časové ztráty je nutné čas průjezdu porovnat s referenčním časem, který by vůz zajel mezi vjezdem a výjezdem závodní rychlostí po trati. Tento referenční čas může být různý, pokud jsou na trati například žluté vlajky (obě znamenají povinné zpomalení), navíc by se mělo zohlednit aktuální závodní tempo jezdce. Jelikož jsou známy body na trati, kde se nachází vjezd a výjezd z boxu, jako řešení bude použito zaznamenávání několika posledních intervalů průjezdu mezi těmito body pro získání reference jako jejich průměru. Výsledná ztráta je pak rozdíl doby průjezdu boxem a referenčního průjezdu po trati.

$$t_3 = t_{pit} - t_{ref} = t_{pit} - \frac{\sum_{i=0}^n t_i}{n} \quad (4.7)$$

Výsledná časová ztráta pro každou z uvedených variant zastávek je jednoduše získána součtem příslušných ztrát. Například pro zastávku s plnou procedurou je výpočet následující:

$$t_{zt} = t_1 + t_2 + t_3 = \pm t_{vp} \pm t_{vb} + m_{mp} * r_{tp} + m_{rpk} * r_{ok} + t_{pit} - \frac{\sum_{i=0}^n t_i}{n} \quad (4.8)$$

Konstanty použité pro výpočet byly získány přímo ze hry a jsou uvedeny v příloze B.

Výpočet odstupů soupeřů Tato informace je známá díky zavedení průběžného výpočtu časových odstupů. Detailněji je vše popsáno v části 4.2.1

²Ve hře se zvednutí vozu provádí vždy, a proto je příslušný čas připočten i v případě, že se pouze tankuje palivo.

Výpočet odstupů soupeřů na trati Jedná se o obdobný výpočet jako v případě prostého výpočtu časových odstupů (viz 4.2.1). Výše uvedené časové odstupy mohou nabývat hodnot přes více kol (soupeř s náskokem přes dvě kola bude mít větší časový odstup než je čas dvou kol), zde je ale potřeba znát aktuální časové odstupy v rámci kola bez ohledu na to, ve kterém kole se daný soupeř nachází. Ve výsledku se porovnávají časy v mini-sektorech ne podle jejich indexu od začátku závodu, ale podle jejich indexu od začátku kola, tj. čas průjezdu daným mini-sektorem bez ohledu na to ve kterém kole kdo je.

Výsledný výpočet Výsledné výpočty jsou vlastně jednoduché a jedná se již pouze o porovnání hodnot. Pro predikci ztráty pozic vůči soupeřům stačí pro soupeře za jezdcem odečíst od aktuálního odstupů časovou ztrátu na pitstop. Pokud u některého soupeře vyjde hodnota záporná, po výjezdu z boxu bude jezdec až za tímto soupeřem a bude na něj ztrácet tento výsledný čas. Protože výpočty nejsou úplně přesné, a i soupeři nejezdí úplně přesně a mohou udělat cestou chybu a zpomalit, je dobré určit nějaký toleranční interval (např $\pm 3s$), kdy ztráta pozice bude ve výsledku jako *možná*.

$$res_t = t_{dif} - t_{zt} \quad (4.9)$$

Pro predikci provozu při návratu na trať pak opět stačí odečíst od aktuálního odstupů časovou ztrátu na pitstop, stejně jako v případě zjištění ztráty pozic. Ztráta v tomto případě ale není vůči soupeřům v pořadí ale vůči soupeřům na trati bez ohledu na kolo. Zde se pouze zjišťuje kolik vozů je v daném intervalu (např $\pm 1.5s$) a proto mohou tvořit na výjezdu provoz.

$$res_t = t_{dif_p} - t_{zt} \quad (4.10)$$

4.2.5 Predikce vývoje technických údajů

4.2.5.1 Význam

Technické údaje z telemetrie vozidla, které se v závodě vyplatí sledovat, jsou například tlaky a opotřebení pneumatik nebo teploty a opotřebení brzd. Pokud jsou známy aktuální hodnoty, jezdec má přehled jestli jsou v pořádku nebo ne. Co ale potřebuje vědět možná ještě více je to, kdy už v pořádku nebudou, případně jestli se k tomu stavu neblíží rychleji než je nutné. Pokud by věděl, že aktuálním stylem jízdy budou brzdy přehřáté a opotřebované, mohl by například přizpůsobit jízdní styl, brzdám ulevit a výrazně oddálit nutnost výměny.

4.2.5.2 Řešení

Řešené technické údaje jsou velmi různorodé. Mají různé fyzikální podstaty, působí na ně jiné vlivy, apod. Co mají ve velké většině společného je to, že prakticky všechny se zaznamenávají do grafu jako časové řady.

Díky skutečnosti, že technické údaje v grafech tvoří časové řady, lze pro predikování jejich budoucího vývoje použít regrese nějakým druhem křivek. Zde je pak potřeba pro každý jednotlivý technický údaj na základě historických dat určit typ křivky, který nejvíce odpovídá jejímu vývoji.

Další možností je použít strojové učení a s pomocí z historických dat vytvořit regresní prediktivní model. Pokud se použije model s dostatečným množstvím možných parametrů a bude tak obsahovat všechny technické o vozidle, ale k tomu například i údaje o počasí, výsledné predikce by mohly zohledňovat i jejich vzájemné ovlivňování. Vytvoření a ladění takového modelu je mimo rozsah této práce, proto by bylo potřeba použít nějaké již hotové řešení.

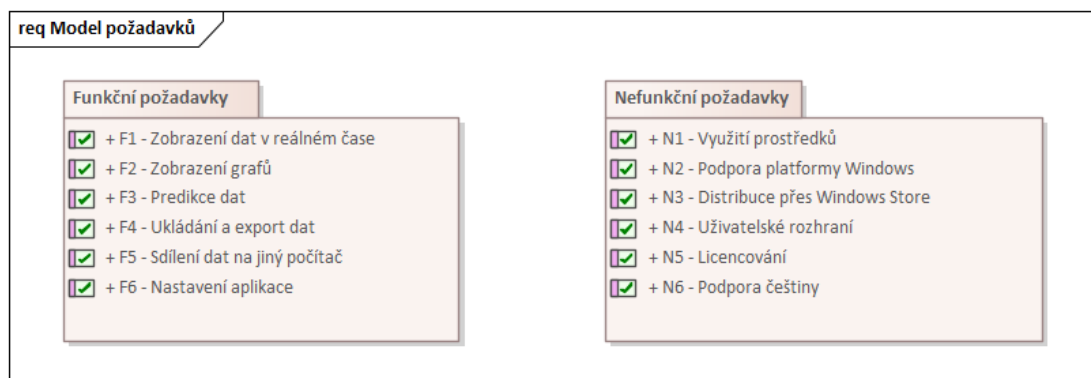
Ve výsledku bude rozhodnutí o použití explicitně definované regrese nebo strojového učení rozhodnuto v části věnované implementaci.

Kapitola 5

Analýza

V této kapitole jsou definovány konkrétní požadavky na aplikaci a její funkcionalitu. Rovněž jsou zde popsány základní případy užití aplikace realizující uvedené požadavky. Nakonec jsou na základě analýzy navrženy modely řešené domény.

Jak už bylo řečeno v úvodu, nápad na realizaci této aplikace vznikl přímo v okruhu komunity, od které byly požadavky a náměty zaznamenány a zpracovány. Samotný sběr požadavků byl dlouhodobější a neformální, proto je zde uvedeno až jejich finální zpracování do odpovídající podoby vhodné pro analýzu. Základní přehled požadavků je vidět na obrázku 5.1.



■ Obrázek 5.1 Model požadavků

5.1 Funkční požadavky

F1 – Zobrazení dat v reálném čase

Aplikace musí umět v reálném čase zobrazovat data získaná z připojené hry jak pomocí UDP připojení, tak čtením sdílené paměti. Zobrazení je možné členit do zvláštních obrazovek mezi kterými je možné přepínat navigací. Není nutné zobrazovat veškerá data, jejich minimální rozsah je ale následující:

- Data o sezení – typ a fáze (závod, ...), online status, název tratě, časy začátku a konce, informace o vlajkách a počasí, mapa okruhu s pozicemi protivníků

- Data o uživateli – jméno a příjmení ve hře, název týmu, typ vozu, závodní číslo, aktuální závodní pozice a pozice v kategorii, aktuálně signalizovaná závodní vlajka, informace o nejlepším a posledním čase na kolo
- Aktuální pořadí protivníků – u každého jméno, tým, závodní číslo, pořadí, závodní kategorii, typ vozu
- Srovnání s protivníky – zobrazení časového odstupe na protivníka před (ztráta) a za uživatelem (náskok)
- Informace k zastávce v boxu – čas otevření a zavření okna pro zastávku v boxu, stav a spotřeba paliva, poškození vozu, stav opotřebení brzd, nastavené množství paliva pro příští tankování
- Telemetrie vozu – vstupy uživatele (sešlápnutí plynu, brzdy, ...), rychlost, převodový stupeň, otáčky a teplota motoru, teplota vody, nastavení stupně ABS a kontroly trakce, teploty brzd a pneumatik, tlak pneumatik

F2 – Zobrazení grafů

Aplikace musí podporovat přehledné zobrazení vývoje získaných dat pomocí grafů. Graf musí být minimálně dostupný pro zobrazení vývoje teplot vzduchu a tratě, teploty chladící kapaliny, stavu paliva, spotřeby paliva, teplot pneumatik a brzd, tlak pneumatik a opotřebení brzdových kotoučů a destiček.

F3 – Predikce dat

Aplikace musí umět generovat predikce budoucích hodnot dat a jejich trendů. Jsou požadovány minimálně následující predikce:

- Predikce zbývajících času na který vystačí palivo aktuálně dostupné v nádrži
- Predikce vývoje časového odstupe pro vůz před a za uživatelem
- Predikce okamžiku, kdy dojde k dojetí protivníka uživatelem nebo uživatele protivníkem
- Predikce času ztraceného zastávkou v boxu, počtu ztracených pozic zastávkou a velikost provozu po výjezdu z boxu
- Predikce vývoje telemetrických dat zobrazovaných v grafu pro omezený budoucí časový úsek

U predikce vývoje v grafu se nepožaduje generovat predikce u všech dat zobrazovaných v grafu uvedených v požadavku F2. Aplikace ale v takovém případě musí umožnit generování predikcí zbylých dat z požadavku F2 pouze s malým zásahem do programu bez jeho významnější změny. Přepočty predikcí nemusí být počítány okamžitě v případě změny hodnot, ale mohou probíhat v předem definovaných intervalech.

F4 – Ukládání a Export dat

Aplikace musí průběžně ukládat data, která budou vhodně a srozumitelně odlišena pro různé sezení (závod, ...). Uživatel musí mít možnost uložená data po ukončení sezení smazat, nebo data konkrétního sezení exportovat ve formátech JSON a CSV pro pozdější použití nebo detailnější analýzu.

Pro ukládání a následný export jsou požadována pouze telemetrická data získaná sdílenou pamětí ve stejném minimálním rozsahu jako v požadavku F1. Větší rozsah dat je možný, ale není vyžadován. Rozsah ukládaných a exportovaných dat musí být snadno rozšířitelný s možným omezeným zásahem do kódu programu bez rozsáhlejších změn.

F5 – Sdílení dat na jiný počítač

Aplikace musí umožnit uživateli v reálném čase sdílet stejná a kompletní data, která aplikace zpracovává. Očekává se sdílení v rámci lokální sítě mezi dvěma instancemi této aplikace s cílem definovaným IP adresou a portem, jiný způsob sdílení není vyžadován. Uživatel tak musí mít možnost se místo připojení na hru připojit na sdílená data a vidět stejná data jako uživatel od kterého se data sdílí. Šifrování sdílených dat není vyžadováno.

F6 – Nastavení aplikace

Aplikace musí uživateli umožnit nastavit základní parametry fungování aplikace. Požadovaným nastavením jsou údaje pro komunikaci se hrou pomocí UDP (adresa, port, heslo pro připojení, heslo pro ovládání, interval aktualizace, časový limit vypršení spojení), a sdílené paměti (adresy paměťových souborů, interval čtení sdílené paměti). Nastavení parametrů pro sdílení (adresa, port), u příjemce i odesílatele je také vyžadováno. Aplikace musí umožnit nastavit i všechny volby které se týkají se výpočtu predikcí, například použité vstupní konstanty a tolerance.

5.2 Nefunkční požadavky

N1 – Využití prostředků

Aplikace musí využívat maximálně 1GB RAM pro svůj běh aby neomezovala simulátor na který je napojena. Využití CPU musí být maximálně takové, aby aplikace dohromady s běžící hrou (při závodě s 25 protivníky) využívala maximálně 90% výkonu CPU.

N2 – Podpora platformy Windows

Simulátor je v současnosti dostupný pouze pro platformu Windows a jeho představení na jiné platformě se v budoucnu neočekává. Aplikace musí běžet jako samostatná desktopová aplikace na stejném stroji a systému, a proto je zásadní její funkčnost pouze na platformě Windows. Předpokládá se podpora Windows od verze sestavení 10.0.20348.0. Jiné platformy nejsou vyžadovány.

N3 – Distribuce přes Windows Store

Aplikace musí být obecně dostupná přes Windows store a tím pádem i být podepsaná platným certifikátem. Nesmí být potřeba přidávat další důvěryhodnou entitu pro správnou funkčnost.

N4 – Uživatelské rozhraní

Uživatelské rozhraní musí být příjemné, moderní, přehledné a umožňovat volbu světlého a tmavého tématu. Ovládání musí být v co největší míře intuitivní a pohodlné. Hodnocení návrhu uživatelského rozhraní a jednotlivých obrazovek bude provedeno pomocí uživatelského testování. Nepředpokládá se možnost úpravy grafického rozhraní přímo uživatelem ani použití na tabletech a mobilních zařízeních. Není proto vyžadována responzivita.

N5 – Licencování

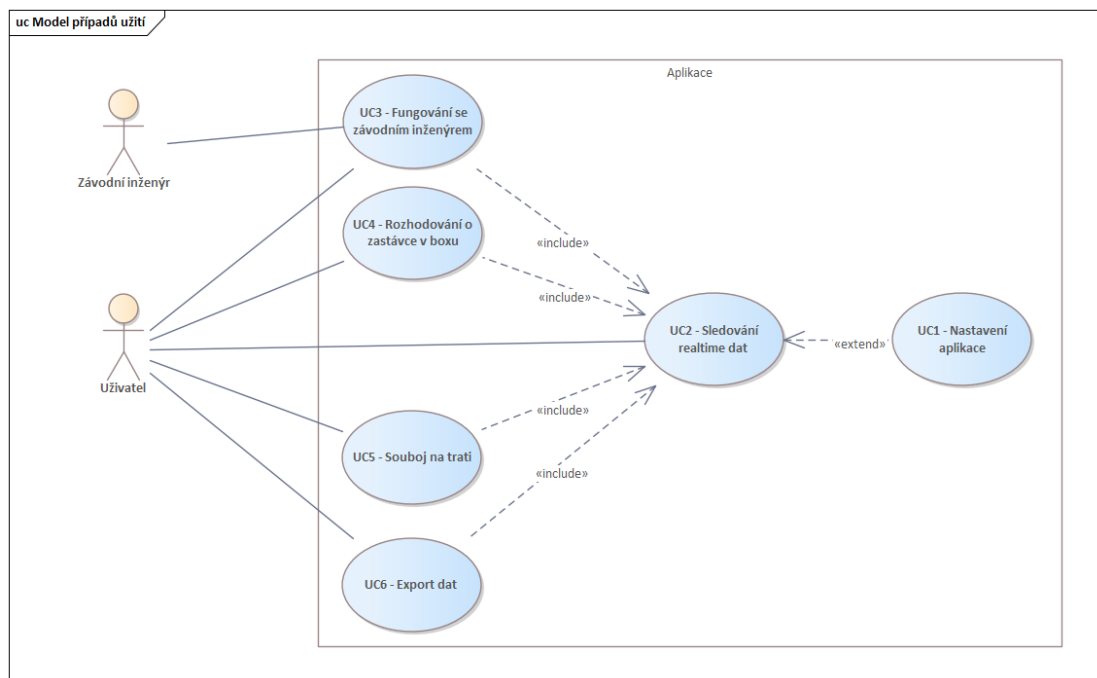
Aplikace musí být licencována s otevřeným kódem. Distribuce musí být zdarma bez jakékoliv monetizace.

N6 – Podpora češtiny

Aplikace musí být lokalizovaná v českém jazyce. Podpora angličtiny pro uživatele neovládající češtinu je vhodná.

5.3 Případy užití

Pro aplikaci byly popsány také základní případy užití. Jejich zobrazení v diagramu je na obrázku 5.2 a dále jsou popsány.



■ Obrázek 5.2 Model případů užití

UC1 – Nastavení aplikace

Umožňuje uživateli nastavit potřebné údaje pro správné chování aplikace.

1. Příklad užití začíná v okamžiku, kdy uživatel potřebuje provést prvotní nastavení nebo jej kdykoliv změnit.
2. Uživatel otevře nastavení použitím tlačítka pro přechod na stránku nastavení.
3. Aplikace ukáže přehled nastavení na samostatné stránce.
4. Uživatel provede potřebné změny a provede uložení příslušným tlačítkem.
5. Aplikace uloží změnu nastavení a aplikuje je kde je potřeba.
6. Uživatel se zavřením stránky nastavení vrátí na předchozí stránku.

UC2 – Sledování realtime dat

Umožňuje uživateli mít kompletní přehled o aktuální situaci a jejím vývoji díky prezentaci ucelených dat vhodně roztríděných a ve vhodně navrženém zobrazení.

1. Případ užití začíná v okamžiku, kdy uživatel chce spustit hru a připojit aplikaci.
2. Uživatel spustí simulátor a v něm závodní sezení.
3. Uživatel spustí aplikaci.
4. Aplikace zobrazí úvodní stránku s ovládacími prvky.
<UC1>
5. Uživatel příslušným tlačítkem připojí aplikaci na simulátor.
6. Aplikace naváže spojení a začne zobrazovat aktuální data hry.

UC3 – Fungování se závodním inženýrem

Umožňuje uživateli sdílet druhému uživateli s jinou instancí aplikace svoje aktuální data, aby mohl pomáhat prvnímu uživateli v jejich interpretaci.

1. Případ užití začíná v okamžiku, kdy uživatel chce začít přeposílat data druhému uživateli (závodnímu inženýrovi) s druhou instancí aplikace.
2. Include (UC2)
3. Uživatel na úvodní stránce nastaví požadovanou cílovou adresu a port pro sdílení dat a příslušným tlačítkem spustí sdílení. Volitelně může také vypnout zobrazování dat ve své instanci aplikace pro ušetření výkonu stanice.
4. Závodní inženýr ve své instanci aplikace na úvodní stránce nastaví jako zdroj příjem sdílení s odpovídajícím nastavením a spustí příjem.
5. Závodní inženýr ve své aplikaci vidí aktuální data ze závodního sezení prvního uživatele.

UC4 – Rozhodování o zastávce v boxu

Umožňuje uživateli mít na jedné obrazovce přehled všech dat a predikcí potřebných k rozhodování o vhodném okamžiku uskutečnění zastávky v boxech, jejích parametrech a pravděpodobné situaci na trati po jejím skončení.

1. Případ užití začíná v okamžiku, kdy uživatel zvažuje a chce prověřit možnost zastávky v boxech.
2. Include (UC2)
3. Uživatel přepne aplikaci na stránku s přehledem dat k zastávce v boxu.
4. Aplikace ukáže data o aktuálním stavu vozu, který má vliv na zastávku v boxech (především stav pneumatik, paliva a poškození).
5. Aplikace také ukáže zastávky s definovanými kombinacemi úkonů časové ztráty, pravděpodobný počet ztracených pozic a provoz při výjezdu.

UC5 – Souboj na trati

Umožňuje uživateli mít přehled o nejbližších soupeřích před a za sebou a mít pohromadě data potřebná k volbě vhodné útočné/obraně taktiky.

1. Případ užití začíná v okamžiku kdy uživatel potřebuje přehled o situaci před a za sebou.
2. Include (UC2)
3. Uživatel přepne aplikaci na stránku s přehledem dat o nejbližších soupeřích.
4. Aplikace ukáže data o soupeřích před a za uživatelem (především časy na kolo a po sektorech, rychlosti v určitých místech trati a další).
5. Aplikace také ukáže za jaký počet kol dojde k soboji s protivníkem před nebo za uživatelem.

UC6 – Export dat

Umožňuje uživateli zpracovat data po závodním sezení offline jiným způsobem.

1. Případ užití začíná v okamžiku, kdy uživatel potřebuje exportovat data z aplikace.
2. Include (UC2)
3. Uživatel dokončí závodní sezení a vypne připojení aplikace na simulátor.
4. Aplikace má data z této a minulých sezení uložena a označena pro snadné nalezení potřebných dat.
5. Uživatel na úvodní stránce vybere data požadovaného sezení a příslušným tlačítkem zažádá o export dat.
6. Aplikace zobrazí okno s možností volby požadovaného adresáře pro export a formátu souboru.
7. Uživatel provede výběr a export potvrdí tlačítkem.
8. Aplikace provede export.

5.4 Realizace požadavků případy užití

Následující tabulka zobrazuje realizaci konkrétních funkčních požadavků jednotlivými případy užití.

■ **Tabulka 5.1** Realizace funkčních požadavků

Požadavek	UC1	UC2	UC3	UC4	UC5	UC6
F1 – Zobrazení realtime dat						
F2 – Zobrazení grafů						
F3 – Predikce dat						
F4 – Ukládání a Export dat						
F5 – Sdílení dat na jiný počítač						
F6 – Nastavení aplikace						

5.5 Rozsah prototypu

Požadavky a případy užití jsou definovány tak, jak by měl vypadat finální produkt. V této práci by ale měl být zpracován pouze prototyp, proto je níže ke konkrétním požadavkům popsáno, co bude v jeho rámci zpracováno. Požadavky, na které je níže odkazováno, jsou popsány v částech 5.1 a 5.2.

F1 Prototyp bude umět číst data z obou možných API hry a v rozsahu jak jej popisuje požadavek. Výjimkou je mapa okruhu, která nebude zobrazena.

F2 Prototyp bude zobrazovat pouze grafy stavu paliva, teplot pneumatik a brzd, tlak pneumatik a opotřebení brzdových kotoučů a destiček.

F3 Prototyp bude umět generovat všechny uvedené predikce. Generované predikce pro vývoj dat v grafech budou implementovány pouze pro vývoj stavu paliva a teplot a tlaků pneumatik. Prediktivní modely pro generované predikce budou funkční, ale bude použito zjednodušených modelů a způsobů generování.

F4 Ukládání dat bude v prototypu realizováno dle požadavků. Export dat bude umožněn pro formáty CSV a JSON.

F5 Funkčnost sdílení na jiný počítač bude realizována dle požadavků. V rámci prototypu ale nebude realizována bezpečnostní analýza této funkcionality, ani implementováno šifrování posílaných dat.

F6 Prototyp umožní uživateli nastavit příslušné parametry připojení na hru a pro sdílení. Nastavení predikcí bude omezeno na možnost vypnutí/zapnutí, interval aktualizace, nastavení tolerancí pro predikci zastávky v boxu a velikost vstupních dat pro predikci časového odstupe a predikci časových řad. Všechny ostatní konstanty a nastavení budou u prototypu nastaveny interně bez možnosti konfigurace.

N1 V rámci prototypu nebude detailněji ověřováno využití HW prostředků.

N2 Prototyp bude dle požadavku podporovat systém Windows v dané verzi a bude realizován jako samostatná desktopová aplikace.

N3 Distribuce pomocí Windows store nebude pro prototyp realizována. Pro testování bude vytvořen jak instalační MSIX soubor s testovacím certifikátem, tak nezabalená varianta aplikace, kterou je možné přímo spustit.

N4 Uživatelské rozhraní bude v prototypu realizováno pomocí stejné zvolené technologie jako v případě finální aplikace. U prototypu nebude realizována responsivita. Výstupy z uživatelského hodnocení nebudou u prototypu zpětně zpracovány.

N5 Prototyp aplikace v podobě, která bude výsledkem této práce, nebude veřejně distribuován. Proto také nebude ani explicitně stanovena licence.

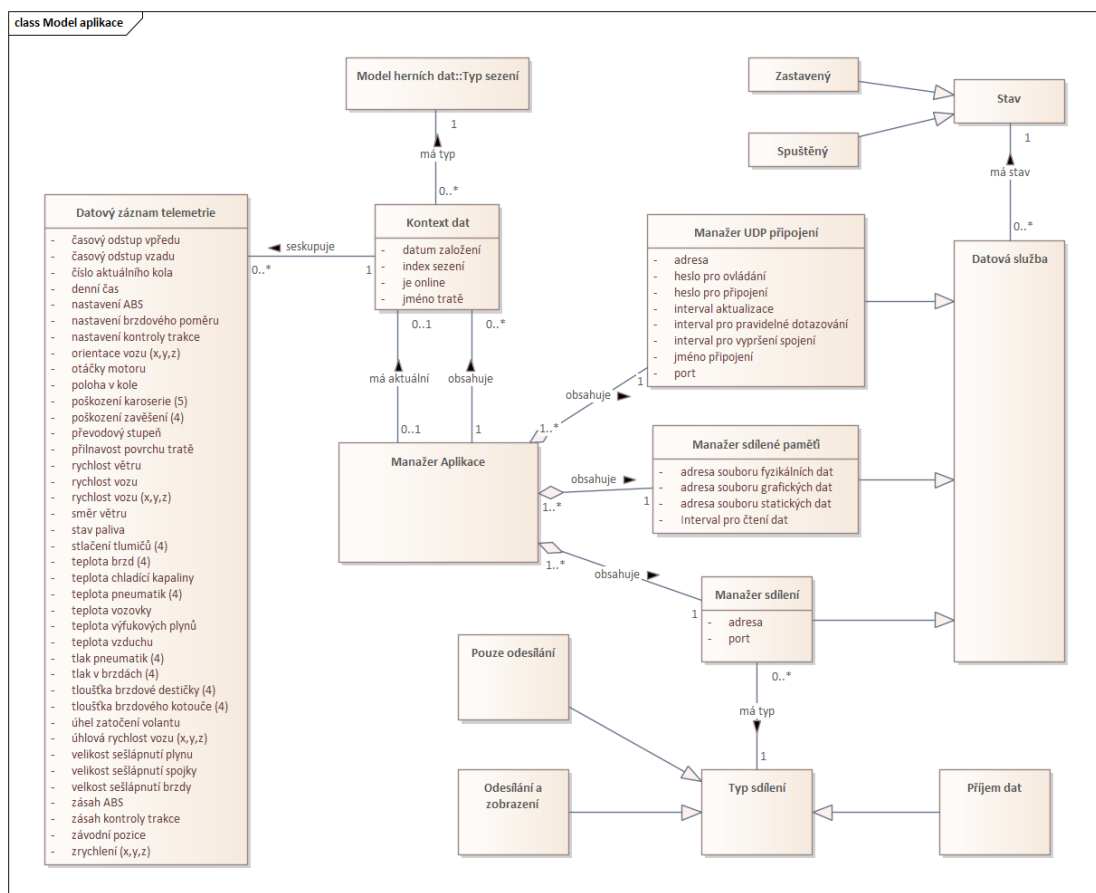
N6 Prototyp bude mít popisky v češtině, pro jiné jazykové nastavení systému Windows budou poskytovány v anglickém jazyce. Nicméně v případě prototypu bude datový obsah, který je získán ze hry, použit v původní podobě bez překládání a proto bude primárně angličtině. Rovněž některé implementační pojmenování, které se mohou uživatelům zobrazit, budou nutně v tuto chvíli pouze v angličtině.

5.6 Doménový model

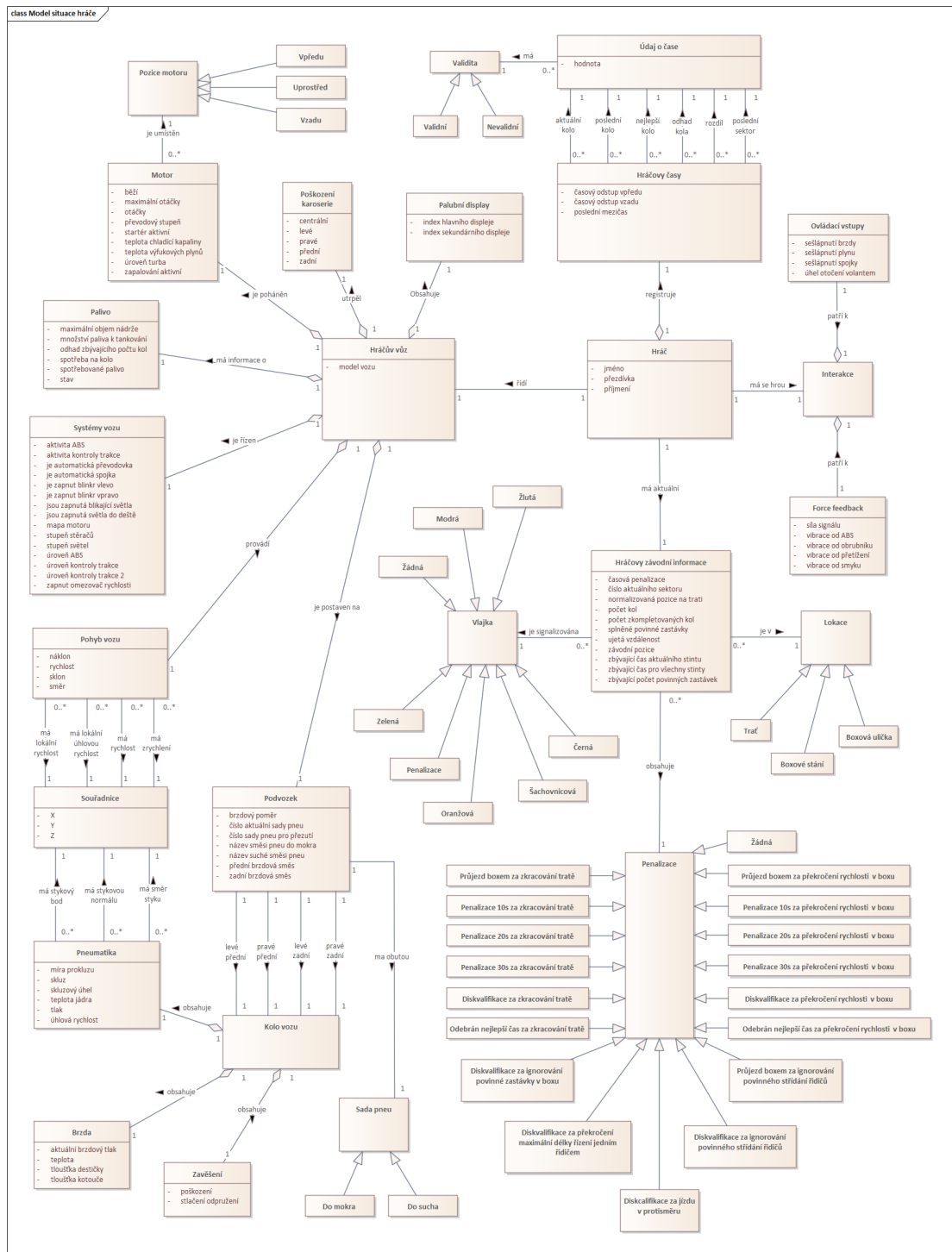
Pro aplikaci jsou navrženy dva relativně samostatné doménové modely. Jeden doménový model reprezentuje aktuální situaci ve hře pomocí průběžných aktualizací. Druhý model reprezentuje stav aplikace, jejích služeb, a referenci na uložená či aktuálně ukládaná data. Oba modely jsou odděleny záměrně, protože reprezentují odlišné domény, které jsou navzájem rozdílné a oddělené. Pokud dochází k jejich interakci, pak pouze z funkčního a nikoliv doménového hlediska.

5.6.1 Doménový model aplikace

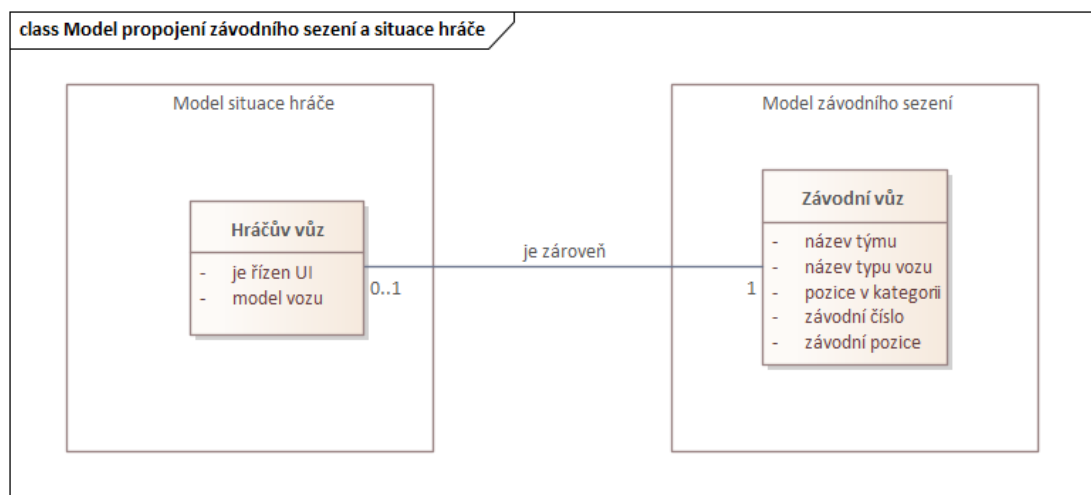
Model aplikace tvoří entity reprezentující její nastavení a stav připojení na hru a sdílení. Zároveň jsou zde entity reprezentující uložená data. Model je zobrazen na obrázku 5.3.



■ Obrázek 5.3 Doménový model aplikace



■ Obrázek 5.5 Doménový model situace hráče



■ **Obrázek 5.6** Propojení modelů hráče a závodního sezení

Návrh a realizace

V této kapitole je popsán návrh a realizace prototypu aplikace. Především pak výběr technologií, návrh grafického rozhraní, návrh architektury a základní ukázka způsobu implementace.

6.1 Výběr technologií

Zvolené technologie a architektura aplikace se ovlivňují, a proto je potřeba nejprve rozhodnout, které technologie budou použity. V přehledu níže jsou rovněž uvedeny technologie, jejichž potřeba vyplyne až později v popisu architektury, ale pro pořádek jsou uvedeny v kompletním přehledu.

Jelikož se nejedná o komerční projekt, nečeká se žádná monetizace ani zde nejsou požadavky na širokou dostupnost, inklinuje se obecně k používání pokud možno nejnovějších technologií. Toto umožní využít nejnovější architektury aplikací a sníží se rychlost zastarávání aplikace v případě, že jakýkoliv pozdější vývoj bude sporadický, či bude prováděna pouze rutinní aktualizace.

6.1.1 Cílová platforma

V požadavku na platformu (viz 5.2) je specifikována jako požadovaná platforma pouze Windows 10 a novější. Z toho důvodu není potřeba hledat technologie a frameworky, které jsou multiplatformní. Naopak by ale multiplatformnost neměla být překážkou výběru některé technologie či frameworku, pokud splní ostatní požadavky.

6.1.2 Jazyk a základní infrastruktura

Volbu jazyka ovlivňují především požadavky kladené na aplikaci a jeho vhodnosti pro jejich naplnění. Mimo to jeho výběr také mohou ovlivňovat další aspekty, jako je dostupnost nástrojů pro vývoj, dostupnost dokumentace nebo znalost vývojáře. Pokud se výběr omezí na nejpoužívanější vhodné jazyky, šlo by aplikaci napsat pomocí libovolného z jazyků Java, .NET, C++, Python či dokonce Javascript. Vždy by ale bylo potřeba použít nějaký framework pro desktopové aplikace.

Pro realizaci byl zvolen jazyk C# a platforma .NET od firmy Microsoft. V době, kdy se začala psát tato práce, byla nejnovější dostupnou verzí **.NET 7**, s k němu příslušnou verzí **C# 11**. Tato verze je sice označena pouze jako verze se standardní délkou podpory (STS), ale i tak byl konec podpory přibližně ve stejné době jako u předchozí verze **.NET 6**. Použití nejnovějších možností platformy navíc přináší nejnovější optimalizace jak z hlediska výkonu a možností knihoven, tak také znamená pozdější zastarání a nutnosti změny na novější verzi. V průběhu realizace je proto potřeba provést migraci na další LTS verzi **.NET 8** a **C# 12**, který má datum vydání až v průběhu tvorby práce. Samotný přechod by měl být bez problému. [35]

6.1.3 Framework uživatelského rozhraní

V rámci platformy .NET je nabízeno velké množství různých frameworků pro tvorbu uživatelského rozhraní, ať už z dílny Microsoftu, nebo třetích stran. Některé z nich jsou již označeny jako zastaralé a určeny k postupnému útlumu (ačkoli stále podporované). Část z nich je zatím stále v překotném vývoji, a ani u těch aktuálně preferovaných nemá sám Microsoft jasno, která by měla být hlavní volbou pro nové aplikace. Ačkoliv architektura by měla směřovat k tomu, že je možné použít libovolný framework, případně budoucí předělání do jiného frameworku by stálo příliš času a energie navíc. Pro všechny uvedené důvody je tedy třeba provést detailnější rozbor stavu aktuální nabídky s ohledem na stanovené požadavky a již zvolenou verzi .NET 7 (.NET 8). Následující rozbor vychází především z přehledu připraveného firmou Microsoft „An overview of Windows development options“ [36] a dále z magisterské práce „C#/.NET GUI toolkit comparison“, která se právě tomuto srovnání věnuje [37], a nakonec z knihy věnované WinUI 3 „Learn WinUI 3.0“ [38].

6.1.3.1 Windows Forms

Windows Forms je původní a nejstarší framework pro uživatelské rozhraní pro .NET přímo od Microsoftu. Je postavený na formulářích pro jednoduché aplikace určené pouze pro Windows a umožňuje rychlé prototypování a vývoj. Pro nový vývoj by se již neměl používat, ačkoliv je stále podporován i v .NET 7 a 8, protože je na něm stále postaveno velké množství starších aplikací.

6.1.3.2 WPF

Druhý framework Microsoftu, také určený pouze pro Windows, obsahuje řadu vylepšení oproti WinForms (například oddělení logiky a UI použitím XAML šablon). Aplikace vypadají výrazně lépe díky použití Direct3D pro vykreslování a tedy i pro vývojáře možnost použití výrazně komplexnějších a graficky intenzivních UI. V současnosti je WPF stále Microsoftem podporován s ohledem na ohromné množství aplikací, které jsou na WPF postavené. Pro nový vývoj tento framework již není doporučovaný a Microsoft se snaží přimět uživatele k migraci na některý z novějších frameworků, především WinUI 3. Několik let nazpět by pro vývoj UI aplikace pro windows a .NET nebyla prakticky jiná adekvátní volba než právě WPF.

6.1.3.3 UWP

První snahou o univerzální vývoj aplikací na desktopy a mobilní zařízení, i když stále pouze s Windows, byl framework UWP. Je velmi podobný WPF, ale kromě XAML umožňuje použití i HTML, knihovny WinUI nebo DirectX. Jeden čas byl preferovaným řešením, i když jeho velkou nevýhodou je silná vazba na konkrétní verzi Windows, a to pouze od některé z verzí Win10 (starší verze Windows nejsou podporované). Microsoft sám nakonec UWP zavrhnul právě pro jeho silnou vazbu na konkrétní verzi Windows a doporučuje přechod na WinUI 3.

6.1.3.4 WinUI 3

WinUI 3 je nejnovější framework Microsoftu pro uživatelské rozhraní postavený na Windows App SDK a Win32. Vznikl evolucí, kdy Microsoft vyřešil nevýhody UWP oddělením WinUI knihovny od Windows a tím vznikl Windows App SDK a postupně nový framework WinUI 3. Je podporován, stejně jako UWP, pouze od verze Windows 10. Microsoft ji aktuálně preferuje pro vývoj desktopových aplikací určených pouze pro Windows a sám se snaží některé aplikace do WinUI 3 přepsat, například Průzkumník souborů. Využití WinUI 3 je možné nejen pro .NET, ale také pro aplikace psané přímo v C++.

6.1.3.5 MAUI

Jedná se o nejnovější počín Microsoftu na poli multiplatformního vývoje aplikací s jedním společným kódem. Prakticky jde o nástupce Xamarin.forms a pro běh na platformě Windows pak využívá právě Windows App SDK. Je směřován primárně spíše na mobilní zařízení, i když v posledních verzích se klade důraz i na podporu Windows desktopů.

6.1.3.6 Blazor hybrid

Blazor hybrid je řešení Microsoftu, které umožňuje kombinovat webovou technologii s vývojem desktopových aplikací. Řešena je jako MAUI aplikace spouštějící nativně webové komponenty Razor, díky čemuž je Blazor hybrid také multiplatformní. Výhodou je možné sdílet komponenty jak pro webovou, tak i desktopovou aplikaci.

6.1.3.7 Avalonia

Avalonia je open-source framework vyvinutý Avalonia UI Community jako vlastní řešení. Velmi připomíná WPF s dílčími vylepšeními a především je multiplatformní.

6.1.3.8 UNO Platform

Uno je další open-source multiplatformní UI framework. Umožňuje také běh UWP a WinUI aplikaci i na jiných platformách než Windows.

6.1.3.9 Finální Výběr

Autor práce s žádným z těchto frameworků nikdy nepracoval a bude se muset jeden z nich naučit v průběhu vývoje uživatelského rozhraní. Dle požadavků není potřeba řešit použití pro více platform, což zde může být naopak na obtíž z hlediska složitosti a velikosti aplikace. Z výše uvedených důvodů nedává smysl pro nový projekt používat starší a nedoporučované frameworky typu WinForms, WPF a UWP. Multiplatformní frameworky MAUI a Avalonia také nebudou potřeba. Výhodu Blazor hybrid ve sdílení komponent s webovou aplikací také není jak využít, a navíc se jedná vlastně opět o již zavrženou MAUI aplikaci. Výsledně tedy pro vývoj bude využita platforma **WinUI 3** a k ní doporučenou MVVM architekturu. [39]¹

6.1.4 Knihovny

Při vývoji desktopové aplikace je potřeba použít řadu knihoven, které by řešily běžné a opakující se problémy jež u takových aplikací vyvstávají. Přímo od Microsoftu jsou dostupné nástroje (toolkits), které obsahují znovupoužitelné implementace řešení pro vývoj desktop aplikací. Z nich budou použity knihovny obsahující často potřebné komponenty pro MVVM architekturu (MVVM Toolkit) a pro WinUI 3 framework (Windows Community Toolkit). [38, 40].

Zároveň bude s výhodou použit doplněk pro VisualStudio s názvem Template Studio for WinUI, který na základě parametrů vybraných v průvodci vygeneruje základní WinUI projekt. Ten obsahuje pouze prázdnou, ale fungující WinUI 3 aplikaci s navigací a přepínáním stránek. Jedná se tak o dobrý výchozí bod pro vývoj ostatních částí aplikace. [41].

6.1.5 Logování

Pro správné logování je vhodné použít externí knihovnu, která je snadno použitelná a podporuje zaznamenávání informací z běhu aplikace a jejich ukládání do souboru nebo jiných cílů (např.

¹Zde je sice MVVM architektura popsána s implementací pro .NET MAUI, ale princip je stejný.

do databáze). Ačkoliv existuje několik možností jako starý log4net nebo novější NLog, z důvodu snadného použití bylo rozhodnuto o použití moderního nástroje Serilog. [42, 43]

6.1.6 Zobrazení grafů

Zobrazení grafů je požadovaná funkčnost, u které je žádoucí použít nějakou z volně dostupných knihoven pro .NET a WinUI 3. Mezi dostupné volby patří například OxyPlot nebo LiveCharts2 [44], který je v základu zdarma a kromě mnoha typů grafů a stylů jejich zobrazení umožňuje i animace a interaktivní ovládání. Je také určen pro většinu .NET UI frameworků, mezi kterými je i WinUI 3.

6.1.7 Perzistence dat

Ukládání množství dat, která by šla při běhu aplikace využít a později exportovat, je každopádně vhodné řešit databází. Vzhledem k omezení desktopové aplikace nemůže být použit plnohodnotný databázový server, ale pouze samostatná knihovna s ukládáním do souboru. S ohledem na podobu dat, která přicházejí v reálném čase jako ucelené zprávy, se zdá být vhodnější použití nějaké NoSQL databáze ukládající data jako řadu souborů, například volně použitelná databáze LiteDB. Na druhou stranu lze očekávat velké množství dat, které je potřeba uložit a přistupovat k nim, což u NoSQL databáze může být problém s ohledem na podobu uložených dat v JSON souborech. Proto by se nabízela i relační databáze SQLite, která také ukládá data do jednoho souboru na disku. Úplně nejlepší by bylo použít specializovanou vestavnou databázi pro .NET uzpůsobenou a optimalizovanou pro použití s časovými řadami, ale zdá se, že žádná taková není aktuálně dostupná. [45, 46, 47]

Pro použití v rámci .NET architektury bylo otestováno řešení LiteDB, které by pro případ ukládání všech dat s frekvencí 20/s a očekávaným během cca 24h generovalo necelé 2 miliony záznamů. Takové množství může být omezující, ale pokud se budou ukládat pouze data definovaná v požadavcích pro ukládání a export, zároveň s použitím binární serializace a komprese, odhadovaná velikost je okolo 1,5 GB, což je pro toto použití odpovídající. Proto bylo finálně rozhodnuto o použití právě LiteDB databáze pro její jednoduchost nasazení a používání ve spojení s možným vkládáním záznamů po skupinách. [45]

6.1.8 Serializace

Pro případy sdílení dat je třeba provést vhodnou serializaci dat pro vytvoření zprávy a deserializaci po přijetí. Serializace pro JSON je však relativně pomalá a výsledkem jsou datově příliš velké zprávy. Pro přenos dat v reálném čase je tak mnohem výhodnější použít binární serializaci. Bohužel z bezpečnostních důvodů byla v posledních verzích .NET nativní binární serializace zavržena. Ačkoliv je stále možné původní serializaci použít použitím vhodného pragma, bude pro serializaci použita knihovna třetí strany. Nakonec bylo zvoleno použití MessagePack-CSharp, protože vykazuje vhodné parametry rychlosti a velikosti serializace/deserializace a umí řešit serializaci s ohledem na strukturu a dědičnost objektů. [48]

6.1.9 Matematické výpočty

Pro zjednodušení některých matematických výpočtů bylo zvoleno použití knihovny Math.net, konkrétně její části pro numerické výpočty Math.NET Numerics. Tato knihovna poskytuje nástroje pro počítání s maticemi a vektory, pro řešení rovnic, statistiky a řady dalších problémů. Mimo to knihovna obsahuje nástroje pro lineární interpolace křivek, které jsou v této práci použity. [49]

6.1.10 Strojové učení

Pro výpočty predikcí je také vhodné použít nějakou z forem strojového učení. Microsoft pro tyto účely vytvořil vlastní knihovnu, která je dostupná pro snadné použití v .NET aplikacích bez nutnosti explicitního naprogramování vlastních modelů. Obsahuje řadu různých komponent, mimo jiné i komponentu pro predikci časových řad a sekvenčních dat. Právě ta je v této práci využita pro předpověď průběhu grafů zaznamenaných dat. [50]

6.1.11 Shrnutí vybraných technologií

Zde je uveden přehledný souhrn technologií představených a vybraných v této kapitole.

■ **Tabulka 6.1** Výběr technologií

Platforma	Windows
Jazyk	C# + .NET 7
GUI framework	WinUI 3
Logování	Serilog
Zobrazení grafů	LiveCharts2
Ukládání dat	LiteDB
Binární serializace	MessagePack-CSharp
Matematické výpočty	Math.Net Numerics
Strojové učení	ML.NET

6.2 Návrh grafického rozhraní

Pro návrh a implementaci uživatelského rozhraní byly převážně použity zdroje uvedené v přehledu na stránce Microsoft [51]. Jedná se především o službu Figma pro návrh podoby obrazovek, Template Studio pro rychlý začátek vývoje, a také WinUI 3 Gallery a Windows Community Toolkit Gallery aplikace.

Návrhy obrazovek byly provedeny s pomocí služby Figma [52]. V této službě má MS přímo umístěny oficiální zdroje obsahující připravené XAML elementy z knihovny WinUI 3, které je zde možné přímo použít do svého vlastního návrhu uživatelského rozhraní. Odkaz na Figma zdroj má Microsoft přímo uveden na zmíněné stránce v dokumentaci [51], ale je dostupný pouze po přihlášení ke službě Figma. Díky existenci tohoto zdroje byl návrh obrazovek proveden přímo s těmito komponenty a vypadá tak již velmi podobně jako finální uživatelské rozhraní. Lze to považovat za poměrně velkou výhodu, protože tvorba a úprava návrhu ve službě Figma byla výrazně rychlejší než jakékoliv pokusy ladit UI přímo pomocí XAML i s použitím funkce „HotReload“. Bohužel ale neexistují žádné vhodné Figma zdroje k použitému frameworku pro zobrazování grafů (viz 6.1.6), proto v návrhu zůstávají grafy pouze naznačené, což je pro účely návrhu naprosto dostatečné.

Návrh aplikace je vytvořen ve stylu běžných desktopových aplikací vytvořených s pomocí WinUI 3. Sice je možné připravit design aplikace v jakékoliv podobě, ale samotné základní rozvržení vypadá dostatečně čistě a je známé a funkční, proto nebyl důvod snažit se jej nějak hluboce upravovat. Byly také v co největší míře využity základní designové prvky a doporučení Microsoft pro vývoj takovýchto aplikací.

6.2.1 Základní rozvržení

Aplikace bude rozdělena do několika obrazovek, mezi kterými se dá přepínat pomocí postranní navigace. Výchozí obrazovkou po spuštění aplikace bude domovská obrazovka, kde budou umístěny ovládací prvky a informace o stavu celé aplikace. Dalšími obrazovkami bude nastavení, a několik obrazovek, které budou obsahovat seskupená data a představovat vlastní smysl existence aplikace.

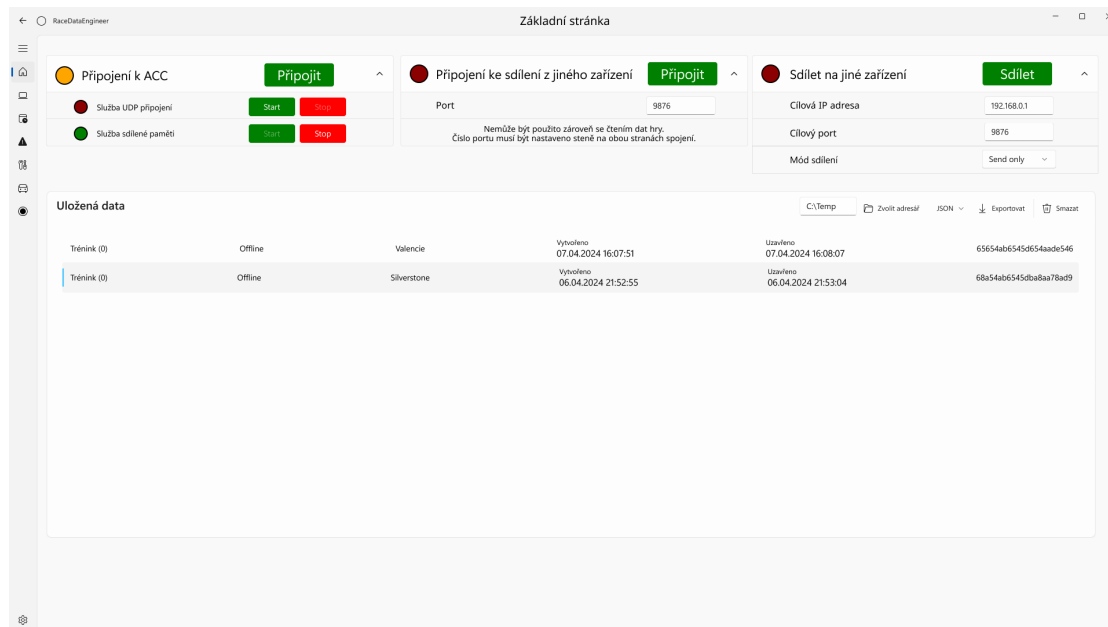
S ohledem na to, že některé obrazovky budou sledované uživatelem přímo v procesech řízení, budou data zobrazena s většími fonty a se snahou o přehlednost a čistotu zobrazení. Z tohoto hlediska pak nedává příliš smysl okno aplikace zmenšovat, došlo by buď ke ztrátě čitelnosti, nebo schování informací za hranu okna. Proto bude stanovena minimální velikost okna rovna FullHD rozlišení, tedy 1920x1080, a grafický návrh bude této velikosti přizpůsoben.

Níže jsou představeny návrhy jednotlivých obrazovek s popisem co které jejich části obsahují za informace a případně k čemu jsou vhodné.

6.2.2 Domovská obrazovka

Domovská obrazovka se otevře po spuštění aplikace, její navržená podoba je ukázána na obrázku 6.1. Bude obsahovat ovládací prvky pro připojení na simulátor, sdílení na jiné zařízení nebo z jiného zařízení, a bude indikovat stav aplikace, tedy jestli je připojení ke hře nebo sdílení aktivní. Rovněž bude obsahovat informace o uložených datech a možnost jejich exportu do souboru a případně jejich smazání.

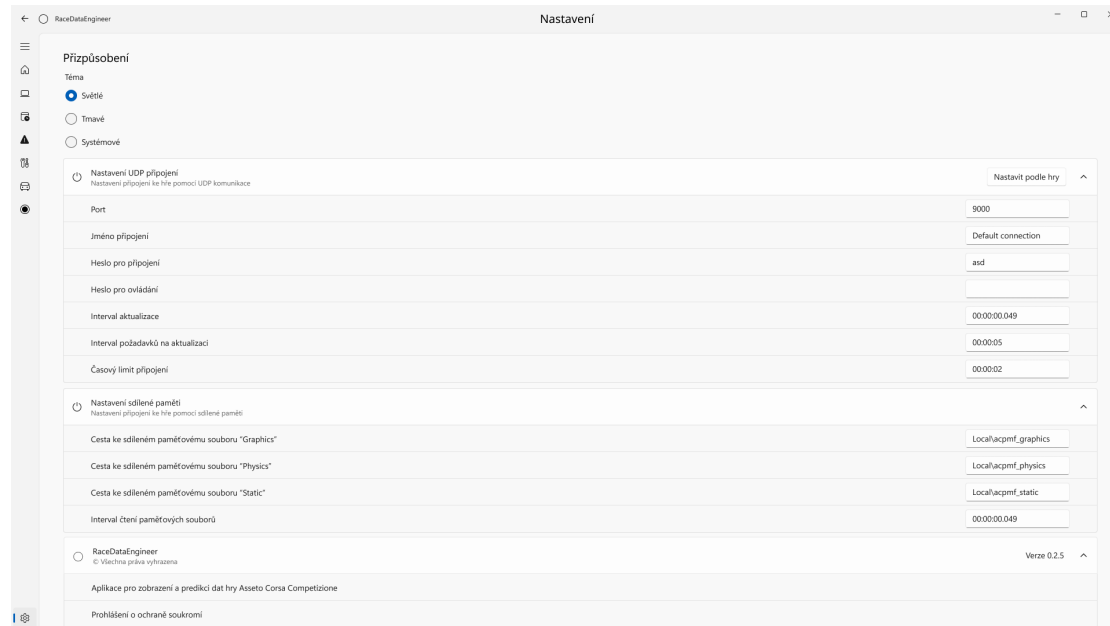
Ovládací prvky budou seskupeny do jednotlivých rozbalovacích karet, kdy bez rozbalení lze vidět tlačítko pro ovládání, označení čeho se týká a indikace stavu. V rozbalovací nabídce bude umístěna možnost detailnějšího nastavení příslušné funkce u níž je předpoklad častější změny. Proto některé volby nebudou umístěny až na stránce nastavení, čímž se minimalizuje pátrání uživatele po změně parametrů a nutnost příliš často přecházet do nastavení celé aplikace.



■ Obrázek 6.1 Návrh domovské obrazovky

6.2.3 Nastavení

Obrazovka nastavení bude obsahovat základní nastavení aplikace, u kterých se nepředpokládá častá změna (například pouze jednou při prvním spuštění aplikace). Bude zde možné nastavit téma (světlé/tmavé/systémové), parametry připojení hry pomocí UDP přenosu i sdílené paměti a další připravené možnosti konfigurace. Nastavení bude seskupeno do rozbalovacích karet, které budou sdružovat jednotlivé parametry a celé nastavení zřehledňovat. Nakonec se zde bude vyškvtovat informace o aplikaci, její verzi a možná právní ustanovení. Návrh této obrazovky je na obrázku 6.2.



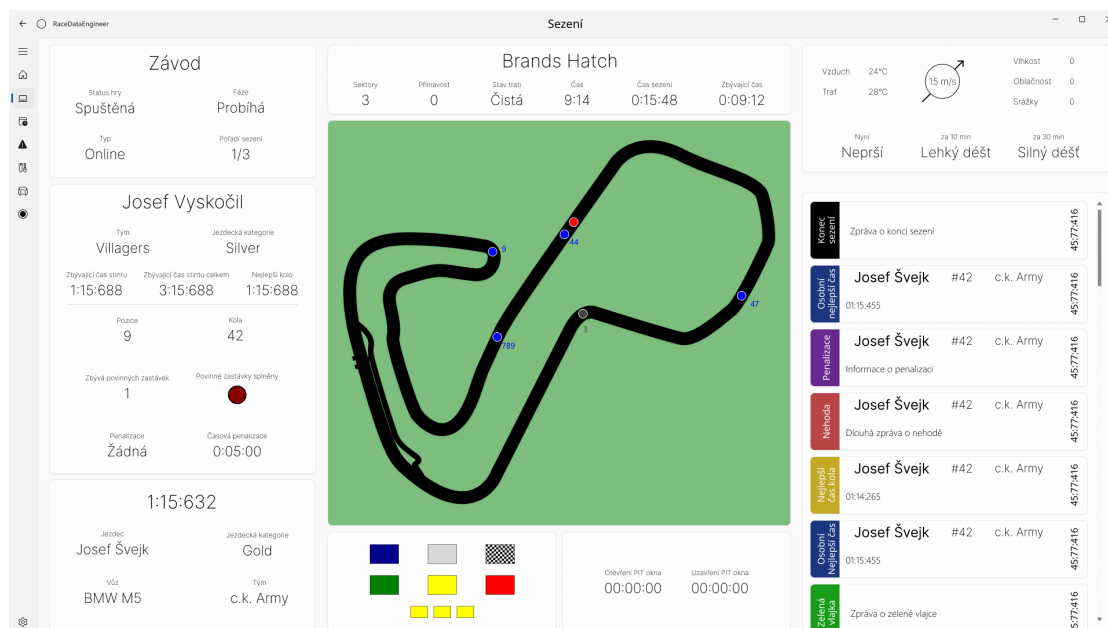
■ **Obrázek 6.2** Návrh obrazovky nastavení

6.2.4 Závodní session

Obrazovka závodní session (sezení) bude ukazovat obecná data o hře. Informace budou seskupeny do viditelně ohraničených bloků/karet, které je budou logicky sdružovat. Návrh lze vidět na obrázku 6.3.

Bude zde uvedeno na jaké trati se závod odehrává, mapa okruhu, informace o session (online status, status hry, fáze závodu, kolikátá je tato session a z kolika celkem) a informace o počasí (teploty vzduchu a trati, vítr, déšť). Pro přehled bude důležité uvést informace o jezdcích - uživatelích (jméno, tým, kategorie, pozice, penalizace, vlajka), nejrychlejším kole session (čas kola a kdo ho zajel), přehled aktivních vlajek a v neposlední řadě přehled důležitých událostí (nehody, nejrychlejší kola, konec závodu).

Jedná se o základní přehled o session, který uživatel potřebuje mít dostupný. Pravděpodobně tato stránka nebude příliš často sledována v průběhu závodu. Spíše se očekává použití u jiné osoby konzumující data ze sdílení nebo ověření některých údajů před začátkem závodu. Nejdůležitější informace, jako počasí, aktivní vlajky, stav povrchu trati, hodiny a zbývající čas session, budou formou dodatečného horní banneru umístěny na většině ostatních obrazovek, což by mělo zvýšit informační přehled uživatelů bez nutnosti obrazovky často přepínat.



■ Obrázek 6.3 Návrh obrazovky závodní session

6.2.5 Závodní pořadí

Na obrazovce závodního pořadí bude pro přehlednost uvedena tabulka všech vozů v aktuálním pořadí (návrh viz obrázek 6.4). U vozů bude uvedeno jejich pořadí, jméno, závodní číslo, tým, typ vozu, poloha (např. na trati / v boxech), aktuální počet zajetých kol a přehled jejich časů na kolo. Jedná se o kompletní pořadí, které si může uživatel zkontrolovat v případě zájmu i v průběhu závodu, ale i zde se počítá spíše s menším využitím v průběhu závodu. V jiných místech této aplikace se plánují zobrazovat pouze nejbližší soupeři, kteří jsou adekvátní závodní situaci.

6.2.6 Závodní situace

Obrazovka závodní situace bude zobrazovat závodní situaci z pohledu uživatele (návrh viz obrázek 6.5). Jedná se o poměrně důležitou obrazovku, jejímž účelem bude pomáhat uživateli v nastavování tempa, a ukázat jestli se blíží k soupeři před sebou nebo jej naopak dojíždí soupeř za ním. Bude zobrazovat nejbližší 3 soupeře před a za jezdcem s podobnými informacemi jako v kompletním pořadí. Navíc bude přehledně ukazovat porovnání uživatelova výkonu s těmito ostatními jezdci, jako je např. čas posledního, aktuálního a nejlepšího kola.

Pro jezdce před a za uživatelem bude zobrazovat také aktuální časový odstup, graf vývoje odstupu během posledních kol, a předpovídat za jak dlouho dojde k dojetí soupeře vpředu nebo bude dojet soupeřem vzadu. Uživatel tak jasně na první pohled uvidí jak si aktuálně stojí s tempem, tedy jestli má přidat, aby nebyl dojet nebo jestli opravdu postupně dojíždí soupeře před sebou. Zároveň bude zobrazovat další údaje o uživateli, které potřebuje mít dostupné, např. jestli je mu signalizována vlajka, obdržel penalizaci apod.

6.2.7 Informace o zastávce v boxech

Informace o zastávce v boxech bude také jedna z nejdůležitějších obrazovek (návrh viz obrázek 6.6). Uživatel na ní uvidí aktuální stav svého vozu, který má vliv na plánování zastávky, např. stav paliva, opotřebení brzd, teploty pneumatik, poškození vozidla a u vybraných hodnot

Číslo	Jméno	Číslo	Tým	Kategorie	Vozydlo	Stav	Kola	Nejlepší kolo	Poslední kolo	Aktuální kolo
1	Josef Švejk	#42	c.k. Army	Overall	BMW M5	Na trati	42	1:15:632	1:16:745	0:44:756
2	Josef Švejk	#42	c.k. Army	Overall	BMW M5	Na trati	42	1:15:632	1:16:745	0:44:756
3	Josef Švejk	#42	c.k. Army	Overall	BMW M5	Na trati	42	1:15:632	1:16:745	0:44:756
4	Josef Švejk	#42	c.k. Army	Overall	BMW M5	Na trati	42	1:15:632	1:16:745	0:44:756
5	Josef Švejk	#42	c.k. Army	Gold	BMW M5	Na trati	42	1:15:632	1:16:745	0:44:756
6	Josef Švejk	#42	c.k. Army	Overall	BMW M5	Na trati	42	1:15:632	1:16:745	0:44:756
7	Josef Švejk	#42	c.k. Army	Overall	BMW M5	Na trati	42	1:15:632	1:16:745	0:44:756
8	Josef Švejk	#42	c.k. Army	Overall	BMW M5	Na trati	42	1:15:632	1:16:745	0:44:756
9	Josef Vyskočil	#609	c.k. Army	Silver	BMW M5	Na trati	42	1:15:632	1:16:745	0:44:756
10	Josef Švejk	#42	c.k. Army	Overall	BMW M5	Na trati	42	1:15:632	1:16:745	0:44:756
11	Josef Švejk	#42	c.k. Army	Overall	BMW M5	Na trati	42	1:15:632	1:16:745	0:44:756
12	Josef Švejk	#42	c.k. Army	Overall	BMW M5	Na trati	42	1:15:632	1:16:745	0:44:756
13	Josef Švejk	#42	c.k. Army	Overall	BMW M5	Na trati	42	1:15:632	1:16:745	0:44:756
14	Josef Švejk	#42	c.k. Army	Overall	BMW M5	Na trati	42	1:15:632	1:16:745	0:44:756
15	Josef Švejk	#42	c.k. Army	AM	BMW M5	Na trati	42	1:15:632	1:16:745	0:44:756

■ Obrázek 6.4 Návrh obrazovky závodního pořadí

Závodní situace

Cas: 9:14 | Cas sezeni: 0:15:48 | Zbyvajici cas: 0:09:12 | Přínavost: 0 | Status přínavosti: Čistá | Větrůch: 24°C | Trať: 28°C | Nyří: Bez deště | za 10 min: Lehký déšť | za 30 min: Silný déšť

Číslo	Jméno	Číslo	Tým	Kategorie	Vozydlo	Stav	Kola	Dokončení kola	Nejlepší kolo	Poslední kolo	Aktuální kolo	Rozdíl	Rychlost
6	Josef Švejk	#42	c.k. Army	Overall	BMW M5 GT3	Na trati	42	59%	1:15:632	1:15:632	1:15:632	+0:02:456	153 km/h
7	Josef Švejk	#42	c.k. Army	Silver	BMW M5 GT3	Na trati	42	59%	1:15:632	1:15:632	1:15:632	+0:02:456	153 km/h
8	Josef Švejk	#42	c.k. Army	Overall	BMW M5 GT3	Na trati	42	59%	1:15:632	1:15:632	1:15:632	+0:02:456	153 km/h

Josef Vyskočil

Tým: Villagers | Závodní číslo: #609

Kategorie: Silver

Časová penalizace: 00:05:00

Spřínání zastávky: Zbyvá zastávek: 2

Časová penalizace: 00:05:00

Časová penalizace: 00:05:00

Časová penalizace: 00:05:00

Ztráta

00:05:486

Kol do souboje: 5

Náskok

00:14:464

Číslo	Jméno	Číslo	Tým	Kategorie	Vozydlo	Stav	Kola	Dokončení kola	Nejlepší kolo	Poslední kolo	Aktuální kolo	Rozdíl	Rychlost
10	Josef Švejk	#42	c.k. Army	Overall	BMW M5 GT3	Na trati	42	59%	1:15:632	1:15:632	1:15:632	+0:02:456	153 km/h
11	Josef Švejk	#42	c.k. Army	Silver	BMW M5 GT3	Na trati	42	59%	1:15:632	1:15:632	1:15:632	+0:02:456	153 km/h
12	Josef Švejk	#42	c.k. Army	AM	BMW M5 GT3	Na trati	42	59%	1:15:632	1:15:632	1:15:632	+0:02:456	153 km/h

■ Obrázek 6.5 Návrh obrazovky závodní situace

i jejich vývoj v grafu a případně i jejich predikce. Zároveň mu aplikace bude predikovat, kolik času pravděpodobně ztratí pro definované rozsahy úkonů při zastávce v boxech, kolik pozic na trati jej zastávka bude stát pokud by se k ní v daném okamžiku rozhodnul, a jak silný může být provoz po návratu. V přehledu souperů pak uvidí jejich pořadí za sebou, z čehož může zjistit, mezi kým se pravděpodobně vrátí na trať. Uživatel díky tomu jasně uvidí, jestli může v případě nepříznivého výhledu zastávky, nebo menšího spotřeby/opotřebení odložit zastávku na vhodnější dobu, nebo zastávku v případě výhodnější situace uspišit.



■ Obrázek 6.6 Návrh obrazovky zastávky v boxech

6.2.8 Telemetrie vozu

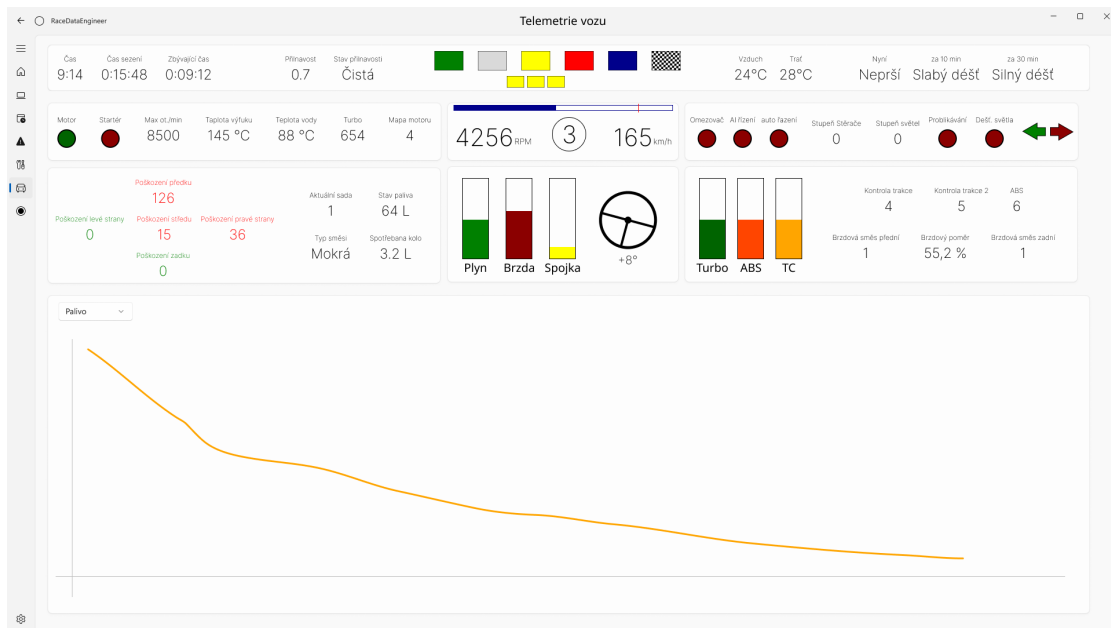
Telemetrie vozu zobrazí veškerá fyzikální a technická data o závodním voze uživatele kromě dat týkajících se kol (návrh viz obrázek 6.7). Bude zde možné najít např. údaje o motoru (převod, otáčky, teploty, turbo, mapování, ..), zapnutých světlech a stěračích, omezovači rychlosti pro jízdu v boxech, nastavení asistentů (kontroly trakce, ABS) a vyvážení brzd (poměr přední vs zadní brzdy). Zobrazeny zde budou také vstupy od uživatele jako je sešlápnutí plynu, brzdy, spojky a otočení volantu. Samozřejmě nebude chybět zobrazení v grafu, které bude dostupné pro palivo, teploty chladicí kapaliny, nastavení asistentů a vyvážení brzd.

6.2.9 Telemetrie kol

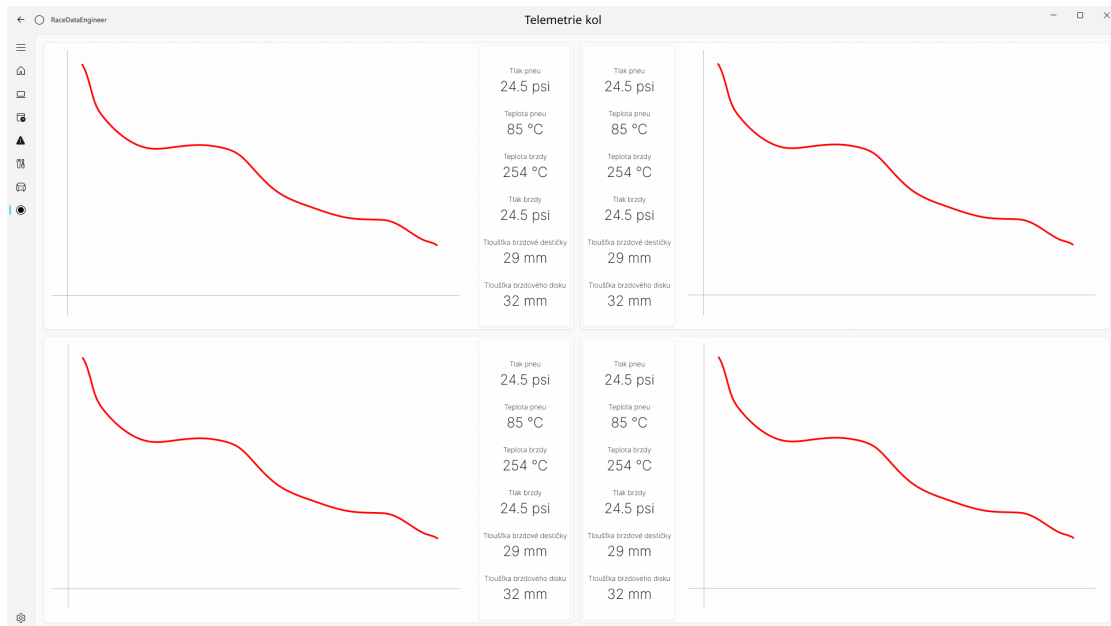
Telemetrie kol bude vyčleněna z důvodu větší přehlednosti a možnosti zobrazovat grafy pro všechna kola zvlášť (návrh viz obrázek 6.8). Obrazovka proto bude rozdělená do 4 částí odpovídající kolům vozidla s údaji o tlacích a teplotách pneumatik a brzd, opotřebeních brzdových kotoučů a destiček. Grafy pak budou zobrazovat vývoj těchto hodnot v čase včetně predikce dalšího vývoje.

6.2.10 Implementace uživatelského rozhraní

Během implementace bylo potřeba některé prvky částečně upravit, například kvůli možnostem samotného WinUI 3, nebo po implementaci výsledek vypadá mírně odlišně. Větší změny byly ověřeny nejprve ve Figma návrhu a až poté dodatečně implementovány, ale menší změny nebyly zpětně do návrhu promítnuty, pokud nedošlo později k větší změně, která tyto menší změny zahrnuje. Proto se může stát, že výsledná podoba uživatelského rozhraní je mírně odlišná od zde prezentovaného návrhu.



■ Obrázek 6.7 Návrh obrazovky telemetrie vozu



■ Obrázek 6.8 Návrh obrazovky telemetrie kol

6.3 Základní architektura

Jakým způsobem je aplikace navržena z velké části odpovídá způsobu, jakým aplikace funguje. Hlavním problémem je skutečnost, že aplikace přijímá pravidelné aktualizace ze dvou různých zdrojů, UDP připojení a sdílení paměti, které nejsou jinak synchronizované, a musí je zobrazit dohromady v jednom grafickém zobrazení. Je proto nutné umožnit asynchronní aktualizaci dat v aplikaci, která je inicializovaná přijetím zprávy, a tuto změnu dále propagovat do uživatelského

rozhraní, kde se musí v reálném čase uživateli příslušná data také aktualizovat. Zároveň je potřeba tato data ukládat, případně i sdílet dále po síti nebo přijímat ze sítě jako z náhradního zdroje.

Při návrhu se mimo jiné vycházelo z publikace edice Microsoftu věnující se architektuřám desktopových aplikací [39]. I když jsou zde konkrétní příklady pouze pro .NET MAUI, řada principů je obecně platných a bylo možné je využít bez ohledu na použitý UI framework. Popsány jsou zde adoptované principy jak pro aplikační logiku, tak pro komunikaci s UI.

6.3.1 Vnitřní fungování aplikace

S ohledem na výše popsané skutečnosti je vhodné aplikaci členit do více samostatných komponent s nějakou formou komunikace mezi nimi. Jednou z možností implementace by bylo použití architektury microservice, což je v případě takovéto desktopové aplikace nevhodné. Některé výhody microservice architektury lze částečně přenést i do monolitní architektury použitím tzv. moduluárního monolitu. Jeho základní myšlenkou je vytvoření několik nezávislých modulů², každý se svou oblastí zodpovědnosti, které spolu mohou komunikovat buď voláním speciálně definovaných rozhraní, či zasíláním zpráv přes centrální uzel zpráv. Díky možnosti komunikovat přes centrální uzel mohou tyto moduly mít nízkou provázanost a vnitřní implementaci nevěřejnou pro ostatní moduly.

Jádrem aplikace je uživatelské rozhraní, doména data a služby pro ukládání a export dat, pro výpočty predikcí a pro ovládání ostatních modulů. Kromě jádra aplikace byly navrženy samostatné moduly pro UDP komunikaci se hrou, čtení sdílené paměti a sdílení dat na jiné zařízení. Původně se uvažovalo o vytvoření samostatných modulů i pro ukládání, export a predikce, ale v současném stavu by takové striktní oddělení pouze působilo nadbytečnou zátěž pro předávání informací mezi nimi. V budoucnu je například možné oddělit predikce do samostatného modulu, pokud se zvětší jejich složitost a bude potřeba je více oddělit od zbytku aplikace.

Dělení na samostatné moduly má mimo jiné velkou výhodu pro testování, kdy díky jasně oddělané zodpovědnosti a jasnému rozhraní je moduly možné testovat naprosto samostatně. Navržené rozdělení na moduly a jejich vzájemná komunikace je přehledně znázorněna na obrázku 6.9, popis jejich implementace je podrobněji popsán dále.

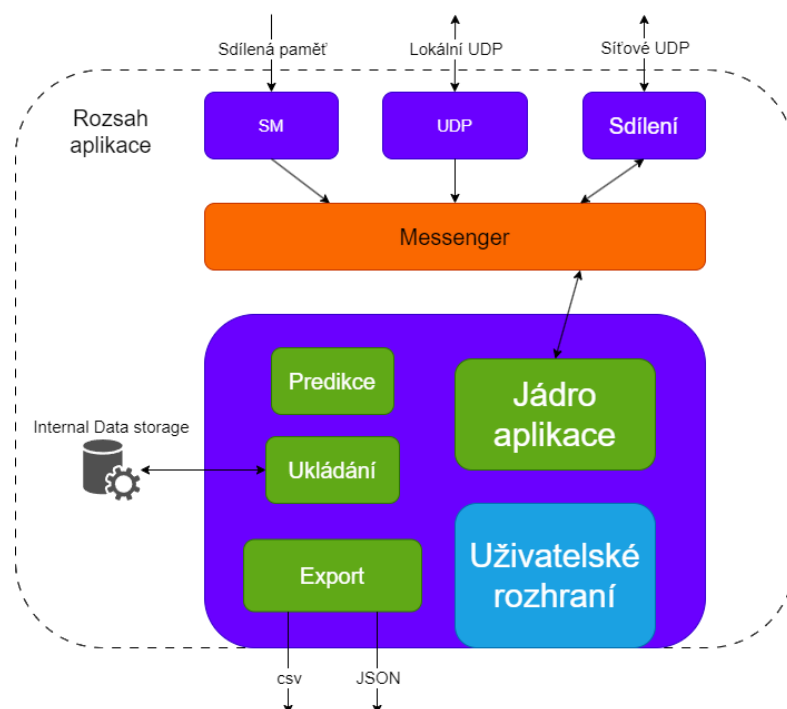
Ačkoliv by moduly mezi sebou mohly komunikovat voláním metod svých rozhraní, pro jasné dělení modulů a zachování nízké provázanosti je důsledně používáno zasílání zpráv. Díky tomu je možné závislost modulů mezi sebou plně oddělit, čímž se předejde nechtěným cyklickým závislostem apod. Právě zasílání informací s aktualizací dat bez nutné závislosti je hlavní účel, pro který byl tento přístup pomocí Messengeru vytvořen. V aplikaci je už využíván .NET Community toolkit, jehož část MVVM Toolkit obsahuje vlastní jednoduchý Messenger, který pro účely této aplikace postačuje. [53]

Zasílání zpráv funguje na základě návrhového vzoru Pozorovatel, což znamená že každý, kdo chce přijímat zprávy určitého typu, musí se u Messengeru zaregistrovat. Messenger zprávu po jejím obdržení přepoše všem příjemcům, který se pro příjem daného typu zprávy u něj zaregistrovali. Při posílání zprávy je také možné specifikovat jméno konkrétní skupiny pro kterou je zpráva určena. Příjemce se pak musí výslovně registrovat k příjmu se stejným jménem skupiny, jinak zprávu neobdrží. S výhodou lze jako názvy skupin použít jméno modulu nebo třídy, pro který je zpráva určena a který ji bude přijímat. Vhodné je to například pro situace, kdy jeden z možných příjemců zprávy rozhoduje, jestli ji má přeposlat dalšímu. V případě, že k přeposlání má opravdu dojít, lze totožnou zprávu poslat pouze se změnou názvu cílové skupiny. [53]

Zprávy mohou být jednosměrné, tzn. že odesílatel zprávu pošle a nestará se o její další osud. To je sice zajištěno synchronně, ale příjemce si může sám rozhodnout, jestli bude zpracovávat zprávu synchronně nebo asynchronně. Druhou možností jsou požadavky (Request zprávy), u kterých odesílatel čeká na odpověď. Čekání může být synchronní či asynchronní (odpověď je Task) a je možné přijímat odpovědi jak od jedné, tak od více protistran. [53]

²Jedná se o logické moduly, nikoliv funkční moduly jak je definuje .NET

Takto nadefinované zprávy jsou více než dostačující pro komunikaci mezi moduly. Pro každý modul je vhodné mít nadefinovaný kontrolní bod (controller) pro příjem zpráv, ale jinak komunikace může probíhat prakticky z libovolného místa v modulu. Zprávy mohou obsahovat základní informaci, například hodnotu *int* nebo text v typu *string*, ale pro potřeby aplikace se posílají strukturované typy. K daným modulům je tak potřeba přidat definici zpráv a v nich zasílaných datových typů, které jsou potřeba ke komunikaci s tímto modulem. Ty představují komplexní obsah, který modul očekává nebo poskytuje, který sám definuje tak, aby jej bylo možné použít při zasílání nebo příjmu zprávy. Messenger a zprávy s obsaženými objekty jsou pak jediné části, které jsou nutně sdílené a na kterých mají ostatní moduly definovanou závislost.

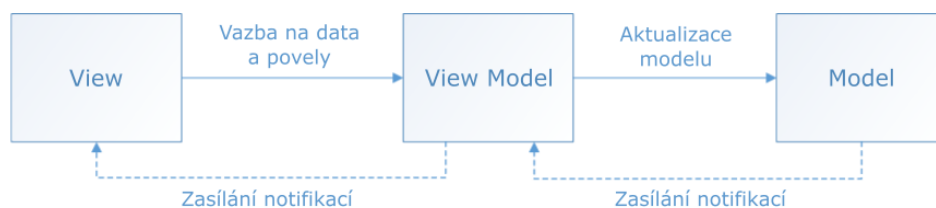


■ **Obrázek 6.9** Architektura modulů s naznačením jejich komunikace

6.3.2 Komunikace s uživatelským rozhraním

Pro komunikaci aplikace a uživatelského rozhraní WinUI 3 je využit MVVM (Model-View-ViewModel) vzor široce používaný pro vývoj desktopových aplikací pro některou z technologií na platformě .NET. V tomto přístupu je aplikační model (Model) a definice zobrazení stránky v XAML (View) odděleno jednou nebo více dalších tříd (ViewModel), které rozšiřují model o další funkce potřebné pouze pro daný View. Ve MVVM tak View zná pouze příslušný ViewModel (nebo ViewModely), provádí navázání (binding) na jeho data a volá definované příkazy (Commands). ViewModel pak podle situace provádí aktualizace dat získaných aktualizacemi od modelu aktualizacími zprávami, nebo zprostředkovává data modelu a případně kontroluje jejich aktualizace přihlášením k události. Příkazy ViewModel převádí na volání metod na modelu nebo přímo využívá dostupných služeb.

Díky tomuto přístupu zůstává aplikace a její model neovlivněné použitím konkrétního typu UI, ViewModel pak pouze připravuje data pro View a lze jej testovat jednotkovými testy. Samotný View pak pouze konzumuje data ViewModelu, a proto se může soustředit pouze na jejich zobrazení. Schéma provázání View-ViewModel-Model je znázorněno na obrázku 6.10.



■ Obrázek 6.10 Schéma MVVM [39]

Existuje několik rozdílných přístupů jak uchopit vztah Model-ViewModel. Jedním z nich je situace, kdy model prezentuje „doménový model“ a ViewModel reprezentuje pouze vrstvu poplatnou jedné zobrazované stránce a obsahující množinu modelů. Druhým přístupem je situace, kdy ViewModel pomocí aktualizáčních zpráv aktualizuje svůj obsah, přičemž informace poskytuje ViewModelu stránky k zobrazení nebo je použit přímo samostatně.

V aplikaci sice dochází k průběžným aktualizacím domény pomocí datových zpráv, ale doména je rozsáhlá, vzájemně velmi provázaná a navíc obsahuje další logiku potřebnou pro fungování aplikace. Proto je zachován přístup samostatných doménových modelů, které jsou přímo použity ve ViewModelu příslušné stránky. S ohledem na aktualizace je většina dat v modelech vytvořena jako pozorovatelná (observable), a tedy při změně vyvolají událost (event) s informací o této skutečnosti, které pak ViewModel nebo View může zachytávat. Dalo by se to považovat za mírné porušení MVVM vzoru, ale skládání domény pouze z ViewModelů by znamenalo míchání další logiky a zobrazení, což je vážnějším porušením MVVM principu. ViewModely jsou použity pro jednotlivé stránky a jiné modely, které nejsou aktualizovány v reálném čase.

6.4 Popis samostatných modulů

Návrh podoby aplikace rozdělené pomocí modulů je znázorněn na obrázku 6.9. Zde jsou popsány pouze moduly mimo jádra aplikace, které s nimi komunikuje prostřednictvím zpráv. Pro každý modul je proto krátce popsáno jeho fungování a popis jejich rozhraní pomocí přijímaných a posílaných zpráv.

6.4.1 Modul UDP

Zodpovědností modulu UDP je lokální síťová UDP komunikace s hrou ACC přes definované zprávy (popis UDP zpráv v příloze A). Modul navazuje a končí spojení s hrou ACC. Pokud je ve spuštěném stavu, ve smyčce přijímá vysílané UDP zprávy vyslané hrou a posílá je dále v definovaném interním formátu.

UDP modul provede start komunikace po přijetí `StartUdpServiceMessage` zprávy obsahující `UdpConnectionDataDto` s informacemi ohledně připojení (adresa, port, hesla, interval). Služba zkontroluje, že komunikace dosud neběží a pokusí se registrovat u hry a navázat spojení s danými parametry. Pokud k navázání spojení došlo, služba začíná číst příchozí zprávy, vytváří z nich vhodné objekty pro aplikaci a zasílá je jako obsah zprávy `StreamDataSourceMessage`.

V případě potřeby je možné se zprávou `CheckStateUdpServiceMessage` dotázat na aktuální stav. Služba sama dává také vědět zprávou `UdpServiceStateChangedMessage` o změně svého stavu, například při ukončení hry která tak přestane vysílat. UDP komunikace pak může být ukončena zprávou `StopUdpServiceMessage`. Služba v takovém případě provede de-registraci komunikace u hry, ukončí spojení, uvolní veškeré dočasně potřebné prostředky a potvrdí zastavení.

6.4.1.1 Zprávy

Modul komunikuje se zbytkem aplikace pouze prostřednictvím interních zpráv popsanych níže.

■ **Tabulka 6.2** Přijímané zprávy UDP modulem

Zpráva	Posílaný objekt	Vracený objekt
StartUdpServiceMessage	UdpConnectionDataDto	UdpServiceStateInfoDto
StopUdpServiceMessage		UdpServiceStateInfoDto
CheckStateUdpServiceMessage		UdpServiceStateInfoDto
CurrentUdpConnectionDataRequest		UdpConnectionDataDto
RequestDataMessage	string	
RequestHudPageChangeMessage	string	
RequestReplayChangeMessage	InstantReplaySettingDto	
RequestFocusChangeMessage	FocusSettingDto	

■ **Tabulka 6.3** Zasílané zprávy od UDP modulu

Zpráva	Posílaný objekt
UdpServiceStateChangedMessage	UdpServiceStateInfoDto
StreamDataSourceMessage	StreamDataDto

Všechny přijímané zprávy musí mít definovaného příjemce `MODULE_UDP`. Všechny zprávy od UDP modulu jsou odesílány jádru aplikace s příjemcem `MODULE_CORE`.

6.4.1.2 Komunikace tříd

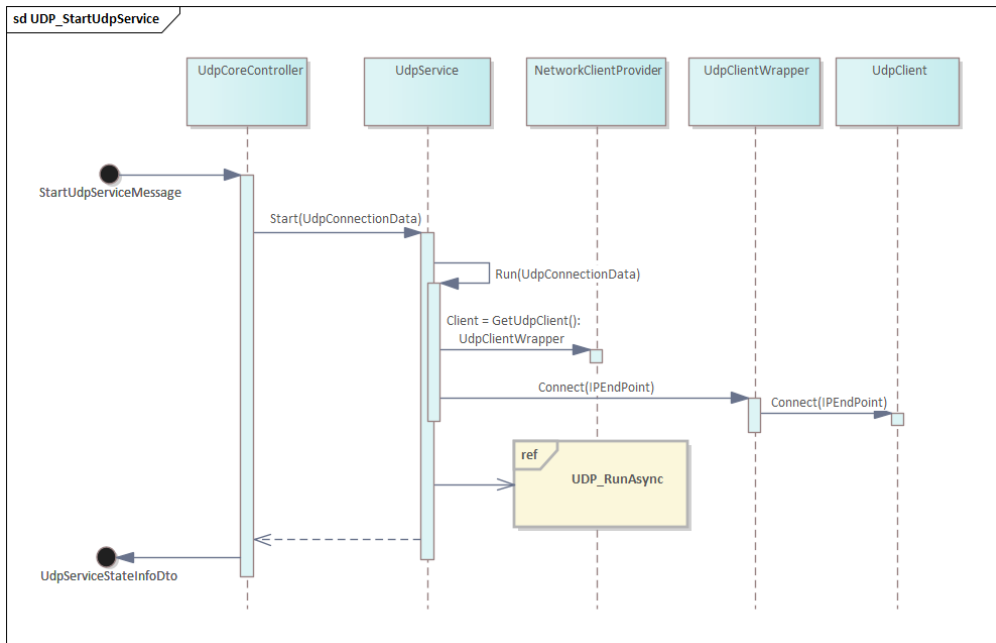
Komunikace tříd je ukázána na situaci, kdy po přijetí `StartUdpServiceMessage` zprávy modul započne UDP komunikaci a začne číst data. Diagram komunikace je znázorněn na obrázku 6.11 a jeho odkaz na obrázku 6.12.

6.4.2 Modul SM

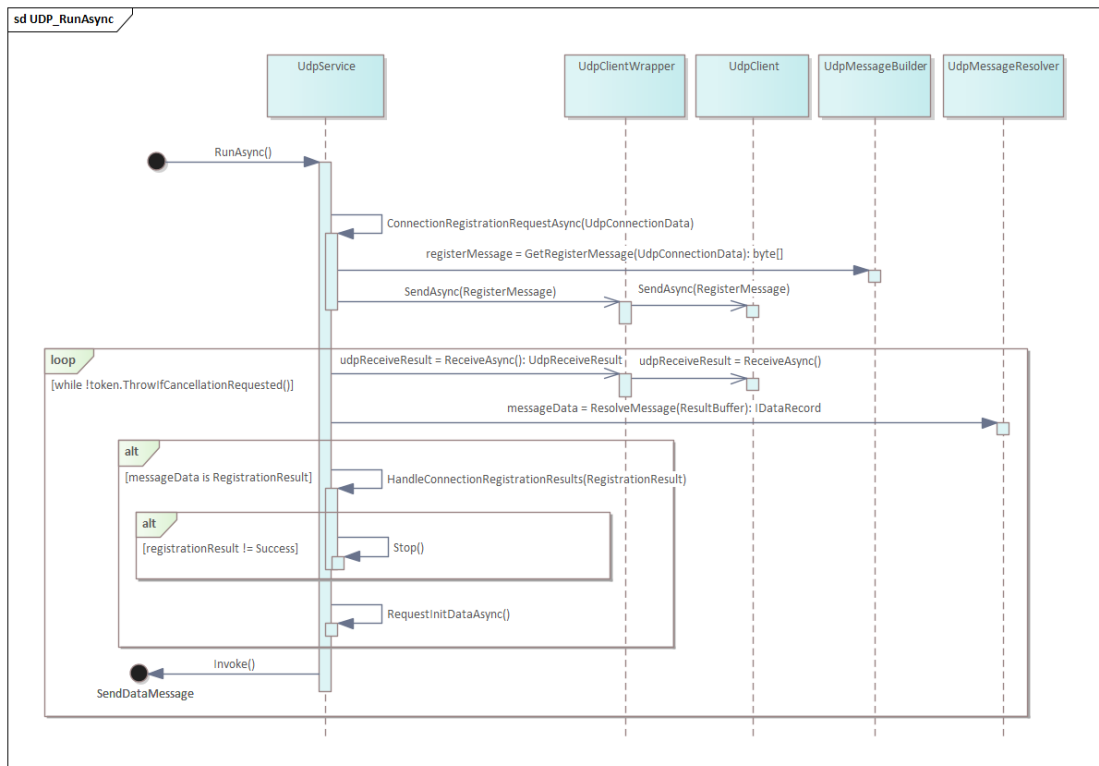
Zodpovědností SM modulu je čtení sdílené paměti. Pokud je modul spuštěn, ve smyčce v pravidelných intervalech čte sdílenou paměť a překládá přečtená data do interních zpráv zasílaných dále aplikaci.

Čtení sdílené paměti začíná po přijetí `StartSmServiceMessage` zprávy, která obsahuje objekt `SmConnectionDataDto` s nastavením pro čtení paměti (adresy souborů, interval). Služba zkontroluje, jestli už čtení neprobíhá a následně se pokusí začít číst soubory sdílené paměti s danými parametry a pošle zpět informaci o výsledku. Pokud je čtení bez problému, služba ve smyčce s daným intervalem čte soubory sdílené paměti, vytváří z nich vhodné objekty pro aplikaci a zasílá je jako obsah zprávy `StreamDataSourceMessage`.

V případě potřeby je možné se zprávou `CheckStateSmServiceMessage` dotázat na aktuální stav. Služba ale dává sama vědět o změně svého stavu zprávou `SmServiceStateChangedMessage`. Čtení sdílené paměti může být ukončeno zprávou `StopSmServiceMessage`. Služba v takovém případě ukončí smyčku čtení paměti, uvolní veškeré dočasně alokované prostředky a potvrdí zastavení.



■ Obrázek 6.11 Komunikace tříd při přijetí zprávy spouštěcí UDP přenos



■ Obrázek 6.12 Komunikace tříd při asynchronním čtení příchozích aktualizací

6.4.2.1 Zprávy

Modul komunikuje se zbytkem aplikace pouze prostřednictvím interních zpráv popsaných v tabulkách 6.4 a 6.5.

■ **Tabulka 6.4** Přijímané zprávy SharedMemory modulem

Zpráva	Posílaný objekt	Vracený objekt
StartSmServiceMessage	SmConnectionDataDto	SmServiceStateInfoDto
StopSmServiceMessage		SmServiceStateInfoDto
CheckStateSmServiceMessage		SmServiceStateInfoDto
CurrentSmConnectionDataRequest		SmServiceStateInfoDto

■ **Tabulka 6.5** Zasílané zprávy od SharedMemory modulu

Zpráva	Posílaný objekt
SmServiceStateChangedMessage	SmServiceStateInfoDto
StreamDataSourceMessage	SharedMemoryFilesDto

Všechny přijímané zprávy musí mít definovaného příjemce `MODULE_SM`. Všechny zprávy odesílané modulem jsou odesílány příjemci `MODULE_CORE`.

6.4.2.2 Komunikace tříd

Modul funguje z velké části obdobně jako modul s UDP komunikací (viz část 6.4.1.2). Hlavní rozdíl je v tom, že místo komunikace po síti využívá třídy z `kernel32.dll` pro čtení sdílené paměti. [16, str. 971]

```
[DllImport("kernel32.dll", SetLastError = true, CharSet = CharSet.Unicode)]
private static extern nint CreateFileMapping(
    nint hFile,
    int lpAttributes,
    FileProtection flProtect,
    uint dwMaximumSizeHigh,
    uint dwMinimumSizeLow,
    string lpName);
```

```
[DllImport("kernel32.dll", SetLastError = true)]
private static extern nint MapViewOfFile(
    nint hFileMappingObject,
    FileRights dwDesiredAccess,
    uint dwFileOffsetHigh,
    uint dwFileOffsetLow,
    uint dwNumberOfBytesToMap);
```

```
[DllImport("kernel32.dll", SetLastError = true)]
private static extern bool UnmapViewOfFile(nint map);
```

```
[DllImport("kernel32.dll", SetLastError = true)]
private static extern int CloseHandle(nint hObject);
```

Tyto třídy jsou použity k mapování předdefinované struktury na konkrétní část paměti. Při běhu modulu dochází pouze ke čtení paměti pomocí těchto struktur a zkopírování potřebných dat do interních objektů. Čtení nutně probíhá v „unsafe“ části kódu, protože je nutné pracovat s ukazateli, jak je ukázáno je na příkladu níže. Zde je vidět, že se dokonce kontroluje stav hry, aby se neposílala data například při pauze, kdy je obsah sdílené paměti vynulován.

```

private unsafe void ReadFiles()
{
    var graphicsPageFilePointer = (GraphicsMemoryFileMapping*)GraphicsPageHandler!
        .FileMap.ToPointer();
    var physicsPageFilePointer = (PhysicsMemoryFileMapping*)PhysicsPageHandler!
        .FileMap.ToPointer();
    var staticsPageFilePointer = (StaticsMemoryFileMapping*)StaticPageHandler!
        .FileMap.ToPointer();

    var gameInfo = SharedMemoryDataResolver.ResolveGameInfo(*graphicsPageFilePointer,
        *physicsPageFilePointer,
        *staticsPageFilePointer);

    SendDataMessage?.Invoke(gameInfo);

    if (gameInfo.GameStatus == GameStatus.Live)
    {
        var memoryData = SharedMemoryDataResolver.ResolveData(*graphicsPageFilePointer,
            *physicsPageFilePointer,
            *staticsPageFilePointer);

        SendDataMessage?.Invoke(memoryData);
    }
}

```

6.4.3 Modul Sdílení

Modul je zodpovědný za sdílení - odesílání i příjem - dat mezi více instancemi aplikace přenosem po síti pomocí UDP protokolu. V případě, že je modul nastaven na posílání, čeká na data z aplikace, která následně ihned odešle na cílovou IP adresu a port. Pokud je nastaven na příjem, naslouchá na definovaném portu a přijatá data okamžitě přijme a pošle aplikaci. Fungování modulu je velmi obdobné už popsaným modulům pro UDP komunikaci, a proto jsou popsány pouze zprávy pomocí kterých modul komunikuje.

6.4.3.1 Zprávy

Modul komunikuje se zbytkem aplikace pouze prostřednictvím interních zpráv. Začátek a konec sdílení je řízen příslušnými zprávami uvedenými v tabulkách 6.6 a 6.7. Modul také sám posílá informace o změně svého stavu, či je možné se na něj příslušnou zprávou také dotázat.

■ **Tabulka 6.6** Přijímané zprávy Sharing modulem

Zpráva	Posílaný objekt	Vracený objekt
StartShareServiceMessage	SharingSettingDto	SharingServiceStateInfoDto
StopShareServiceMessage		SharingServiceStateInfoDto
CheckStateShareServiceMessage		SharingServiceStateInfoDto
ShareServiceStateChangedMessage	ShareServiceStateInfoDto	
StreamDataSourceMessage	StreamDataDto	

■ **Tabulka 6.7** Zasílané zprávy od Sharing modulu

Zpráva	Posílaný objekt
ShareServiceStateChangedMessage	SharingServiceStateInfoDto
StreamDataSourceMessage	StreamDataDto

Všechny přijímané zprávy musí mít definovaného příjemce `MODULE_SHARING`. Všechny zprávy zasílané modulem jsou odesílány příjemci `MODULE_CORE`.

6.5 Model tříd

Modely tříd byly vytvořeny na základě doménových modelů a jsou zobrazeny níže na obr. 6.13, 6.14, a 6.15. U modelu aplikace bylo oproti doménovému modelu potřeba provést změnu z důvodu umožnění změn (a ukládání) parametrů UDP připojení a sdílené paměti přímo v nastavení aplikace. Proto jsou stavy datových služeb uvedeny přímo jako výčtové hodnoty v třídě aplikace a jejich nastavení jsou vedeny jako samostatné třídy.

Zároveň bylo u ukládání telemetrických dat provedeno vložení jedné třídy obsahující informaci o kontextu a o telemetrických datech. To bylo vynuceno nutností serializace telemetrických dat při ukládání, aniž by se ztratila vazba na daný kontext.

6.6 Model komunikace

V aplikaci je mnoho míst, kde mezi sebou komunikují různé třídy, a proto je pro ilustraci vybrán jeden příklad. Model komunikace tříd v aplikaci je ukázán na příkladu připojení aplikace ke hře a začátku čtení dat. Na začátku se uživatel rozhodne připojit hru na aplikaci, což lze vidět na obrázku 6.16. Jeho odkazy jsou postupně na obrázcích 6.17, 6.18 a 6.19. Zde je informace o tomto záměru předána od uživatele přes vrstvu uživatelského rozhraní až dovnitř aplikace, která postupně:

- Zkontroluje, jestli může připojení provést.
- Uzavře a uklidí případný pozůstalý datový kontext.
- Připraví nový kontext v jehož rámci se budou ukládat data.
- Načte nastavení UDP připojení, požádá o spuštění UDP přenosu a aktualizuje stav.
- Načte nastavení sdílené paměti, požádá o spuštění jejího čtení a aktualizuje stav.
- Načte nastavení predikcí a případně požádá o jejich nastartování.

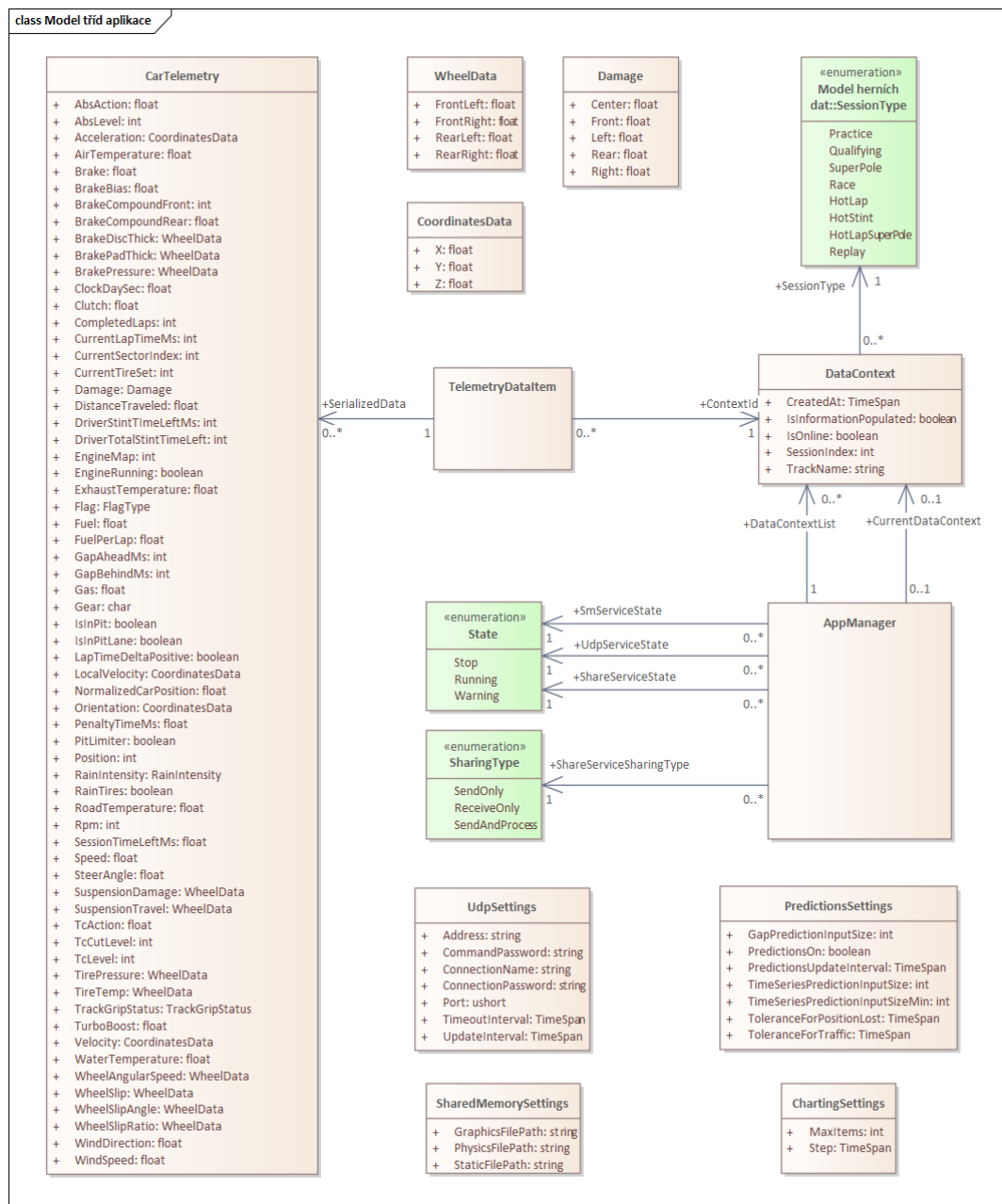
6.7 Příklady implementace predikcí

V této části jsou uvedeny příklady konkrétních implementací některých predikcí. K popisu byly vybrány predikce časových odstupů a predikce časových řad. Obě jsou, stejně jako všechny ostatní, volány skrze `PredictionService`, která spustí výpočty asynchronně, aby nedocházelo k blokování aplikace. Příklad volání pro predikci časových odstupů.

```
public Task<GapPredictionResult> PredictGaps(GapPredictionInput input)
{
    var provider = new GapPredictionProvider();
    return Task.Run(() => provider.Predict(input));
}
```

6.7.1 Implementace predikce časových odstupů

Časové odstupy počítá samostatná třída `GapPredictionProvider`, která přijímá jako vstupní definici `GapPredictionInput` který obsahuje množství prvků `GapRecord`. Každý z nich obsahuje historickou informaci o nedávné poloze na trati (jako podíl křivky okruhu) a o konkrétním časovém odstupu v tom místě. Jako výstup pak poskytuje `GapPredictionResult` obsahující množství stejných prvků `GapRecord`, nyní představující predikovaná data, které je omezeno definovaným maximálním množstvím prvků. Protože toto množství nemusí být dostatečné pro zjištění vzdálenosti,



■ Obrázek 6.13 Model tříd aplikace

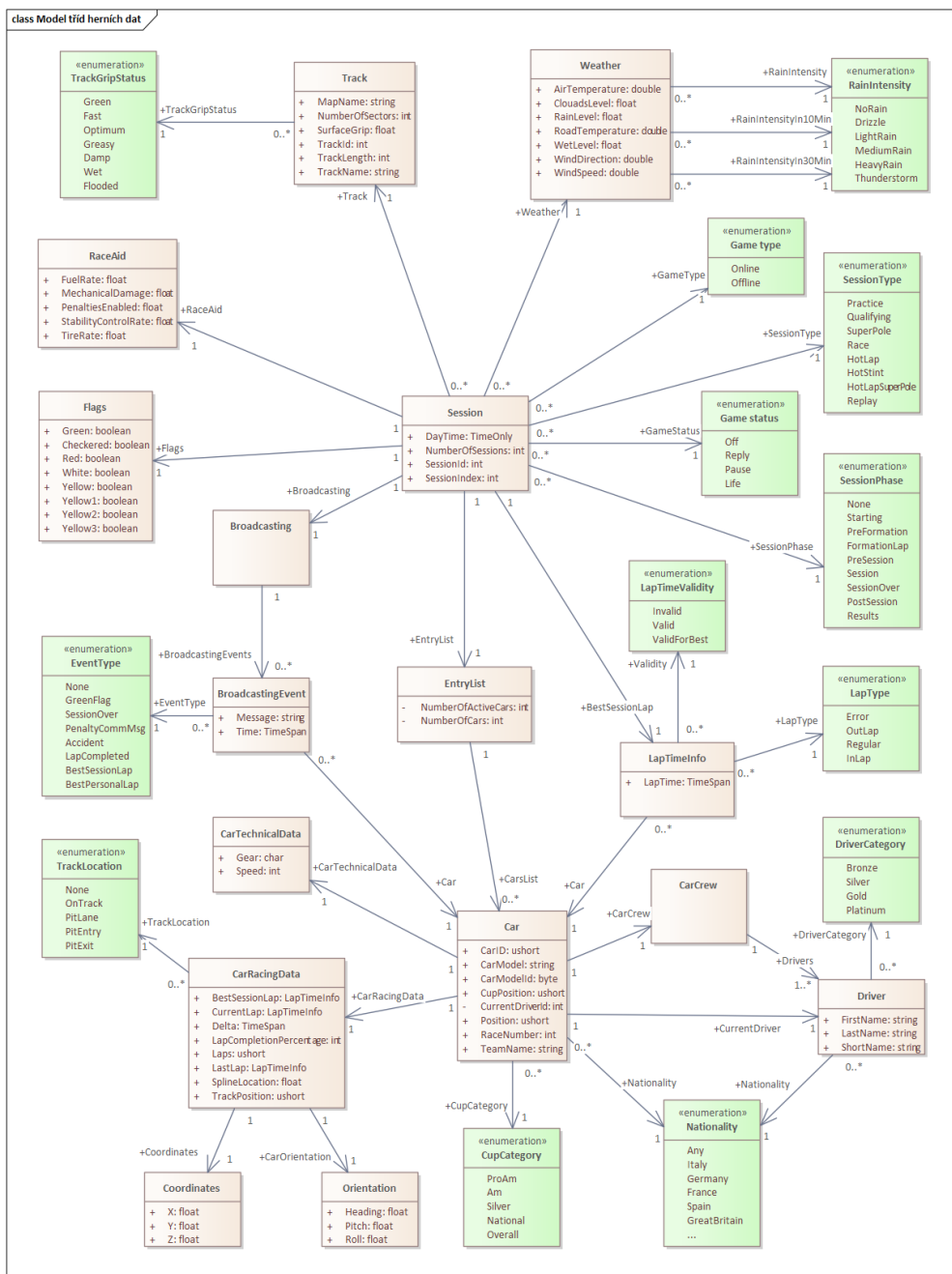
kdy odstup klesne na nulu (a dojde k dojetí), obsahuje výstup také informace o vzdálenosti k nulovému odstupu `LapsToZeroGap`, případně k hodnotě kdy už dojde k souboji `LapsToBattleStart`.

```

public record GapRecord(double TrackPosition, TimeSpan Gap);

public record GapPredictionInput(IEnumerable<GapRecord> GapRecords) : IPredictionInput;

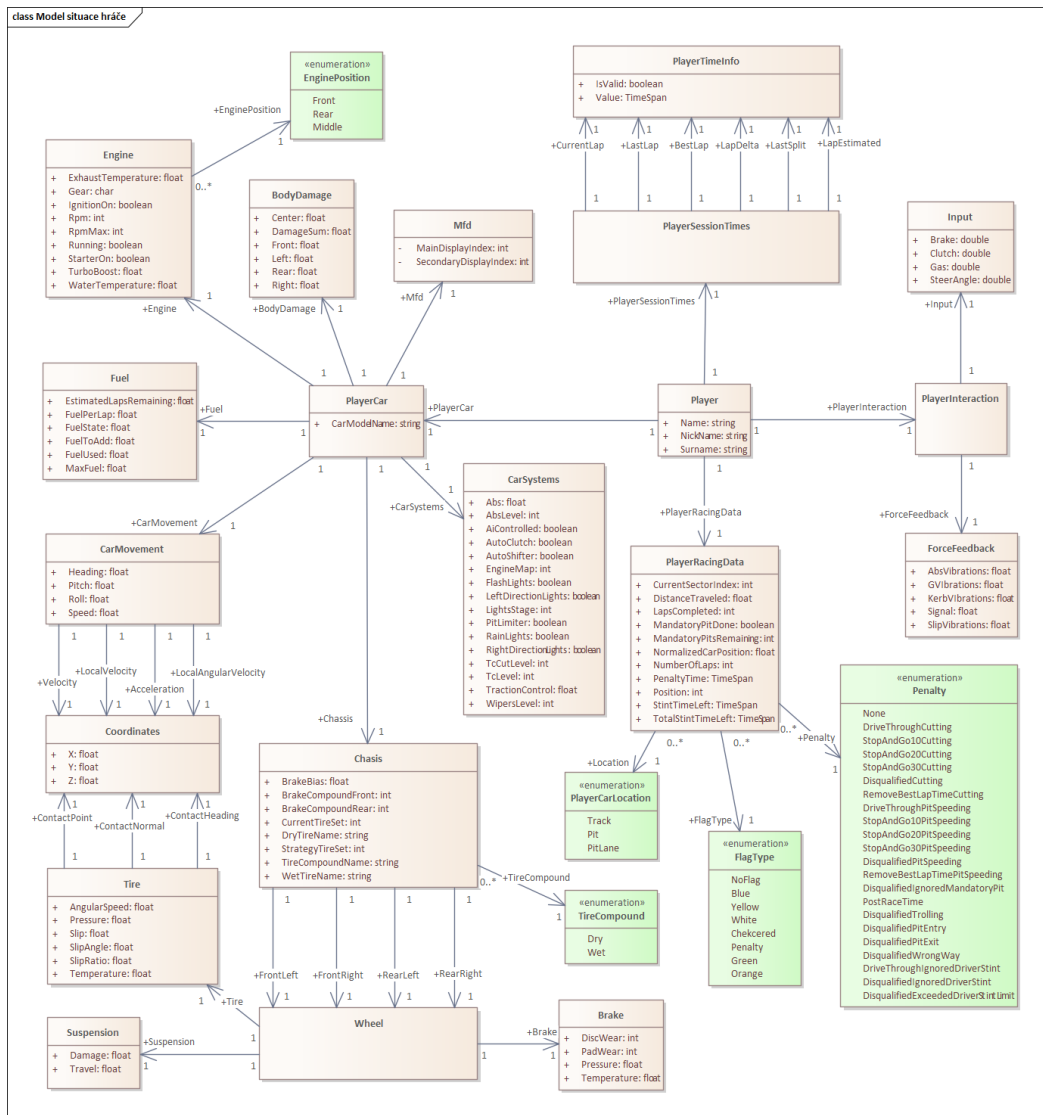
public record GapPredictionResult(List<GapRecord> GapPredictions,
    double? LapsToZeroGap,
    double? LapsToBattleStart
) : IPredictionResult;
  
```



■ Obrázek 6.14 Model tříd závodního sezení

Jak už bylo zmíněno v teorii predikce odstupů v části 4.2.1, předpokládá se lineární vývoj, a proto byla k výpočtu použita lineární regrese. Zároveň lze očekávat malé množství údajů, protože časový odstup se mění pouze v rámci mini-sektorů, což znamená 10 hodnot na kolo.

Pro usnadnění výpočtu byla použita knihovna Math.NET Numerics, která obsahuje připravené funkce pro lineární regrese, jejichž použití je velmi jednoduché. Následující příklad dvou funkcí používá její funkce `Fit.Line` a `Fit.LineFunc` tak, jak jsou uvedeny v dokumentaci, a vrací koeficienty lineární funkce či přímo lineární funkci, kterou lze použít. [54].

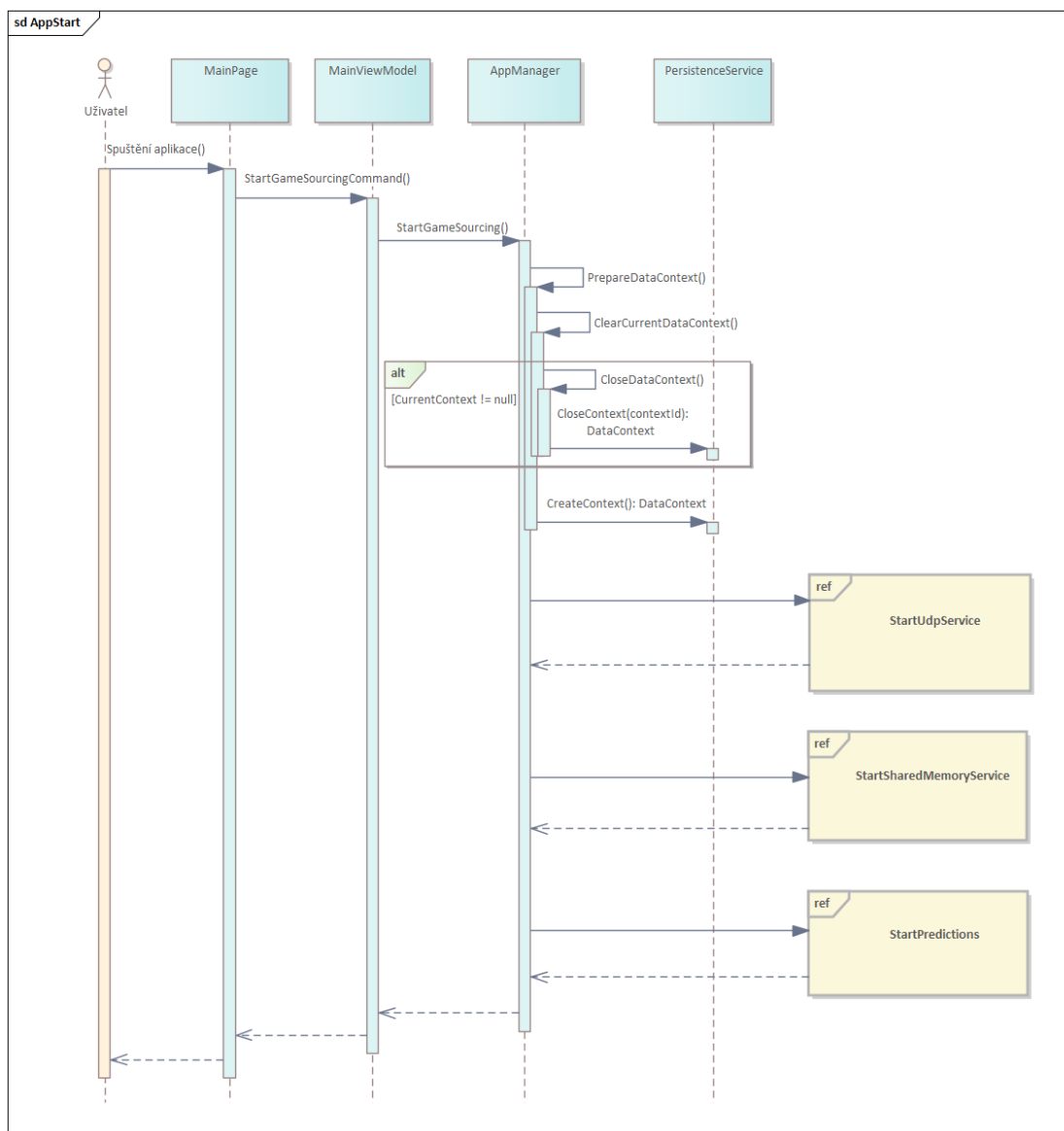


■ Obrázek 6.15 Model tříd situace hráče

```

/**
 * Vrácí koeficienty rovnice  $y = ax + b$ 
 */
private (double b, double a) GetLinerRegressionCoefficients(ICollection<GapRecord> gapRecords)
{
    var points = gapRecords.Select(x => x.TrackPosition).ToArray();
    var values = gapRecords.Select(x => x.Gap.TotalMilliseconds).ToArray();
    return Fit.Line(points, values);
}

/**
 * Vrácí první funkci  $x \rightarrow y$ 
 */
private Func<double, double> GetLinerRegression(ICollection<GapRecord> gapRecords)
{
    var points = gapRecords.Select(x => x.TrackPosition).ToArray();
    var values = gapRecords.Select(x => x.Gap.TotalMilliseconds).ToArray();
    return Fit.LineFunc(points, values);
}
    
```

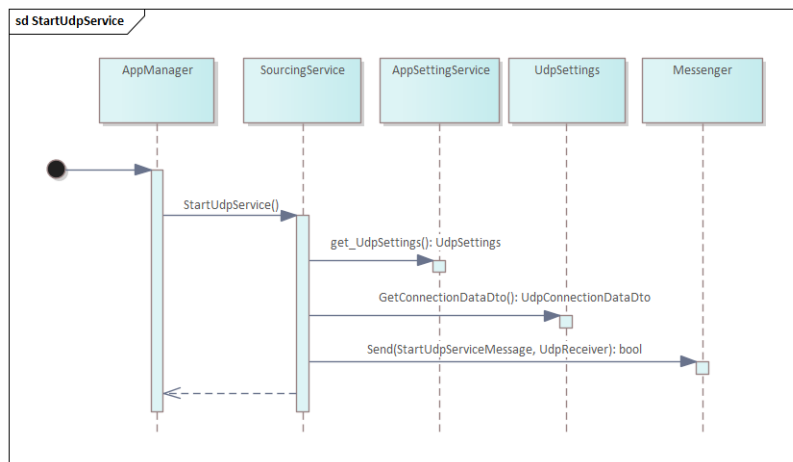



■ Obrázek 6.16 Komunikace tříd při spuštění aplikace uživatelem

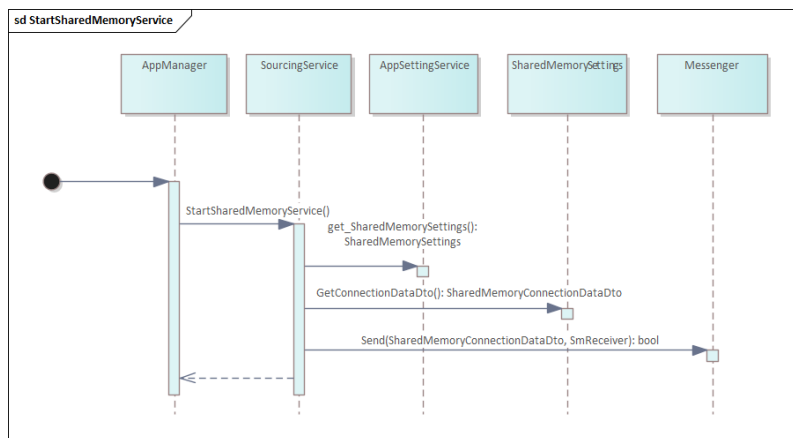
Pokud se na začátku výpočtu zkontroluje, že vstupem jsou alespoň dva body, je s použitím těchto funkcí snadné získat potřebnou lineární závislost. Na základě toho už je jednoduché připravit několik predikcí se stejným časovým krokem jako jsou vstupní data, či zjistit požadované vzdálenosti.

6.7.2 Implementace predikce časových řad

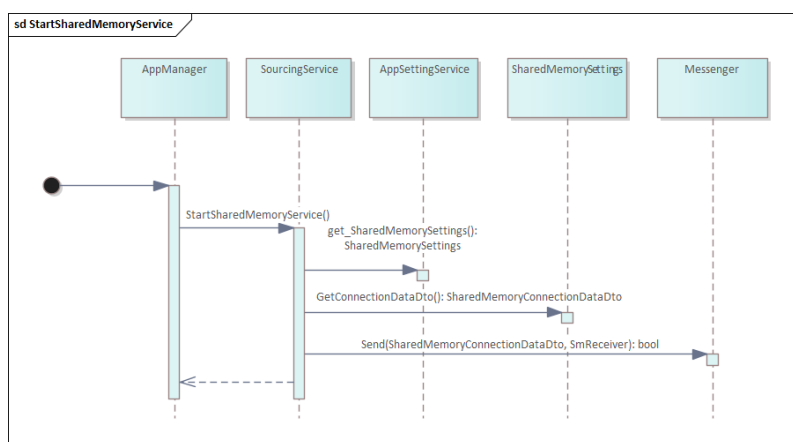
Predikce časových řad zajišťuje samostatná třída `TimeSeriesPredictionProvider`. Jako vstup přijímá objekt `TimeSeriesPredictionInputMerged` obsahující pole `TimeSeriesInput` objektů představujících dvojice klíč-hodnota, a možné údaje o velikosti horizontu predikce a velikosti okna pro výpočet. Jako výsledek výpočtu vrací `TimeSeriesPredictionInputMerged` objekt s řadou výsledných hodnot `TsRecord`.



■ Obrázek 6.17 Komunikace tříd při spuštění UDP



■ Obrázek 6.18 Komunikace tříd při spuštění sdílené paměti



■ Obrázek 6.19 Komunikace tříd při spuštění predikcí

```

public class TimeSeriesInput
{
    public float Key { get; set; }
    public float Value { get; set; }
}

public record TimeSeriesPredictionInputMerged(TimeSeriesInput [] InputArray,
                                             int? PredictionHorizon = null,
                                             float? WindowFraction = null)
    : IPredictionInput;

public record TsRecord(float Key, float Value);

public record TimeSeriesPredictionResultMerged(TsRecord [] ResultArray) : IPredictionResult;

```

Protože výpočet časových řad je náročnější, bylo vyzkoušeno použití knihovny ML.NET pro strojové učení přímo určené pro jednoduché použití v rámci aplikací. Na základě dostupných zdrojů a vzorových použití byla implementována jednoduchá funkce, která celý proces od přípravy modelu a vygenerování výstupu provede ve velmi krátkém čase. Právě použitá knihovna vyžaduje vstupy v uvedeném formátu `TimeSeriesInput`. [50, 55]

```

private const float DefaultPredictWindowFraction = 0.25f;

public TimeSeriesPredictionResultMerged Predict(TimeSeriesPredictionInputMerged input)
{
    var context = new MLContext { FallbackToCpu = true };
    var data = context.Data.LoadFromEnumerable(input.InputArray,
                                              SchemaDefinition.Create(typeof(TimeSeriesInput)));

    var inputSize = input.InputArray.Length;
    var horizon = input.PredictionHorizon ?? input.InputArray.Length;
    var windowFraction = input.WindowFraction ?? DefaultPredictWindowFraction;
    var windowSize = (int)Math.Floor(windowFraction * inputSize);
    var pipeline = context.Forecasting.ForecastBySsa(nameof(TimeSeriesResult.ForecastValues),
                                                    nameof(TimeSeriesInput.Value),
                                                    windowSize: windowSize,
                                                    seriesLength: inputSize,
                                                    trainSize: inputSize,
                                                    horizon: horizon);

    var model = pipeline.Fit(data);
    var engine = model.CreateTimeSeriesEngine<TimeSeriesInput, TimeSeriesResult>(context);
    var forecast = engine.Predict();

    var predictionKeys = GetPredictionKeys(input.InputArray, forecast.ForecastValues?.Length ?? 0);
    return forecast.ForecastValues is null || predictionKeys.Length == 0
        ? new TimeSeriesPredictionResultMerged([])
        : new TimeSeriesPredictionResultMerged(
            predictionKeys.Zip(forecast.ForecastValues)
                .Select(x => new TsRecord(x.First, x.Second))
                .ToArray());
}

```

Odkazovaná funkce `GetPredictionKeys` má za úkol pouze připravit klíče pro získané výsledky na základě klíčů vstupních dat a definovaného množství predikovaných dat. Ve výsledku jsou zde tato místa kde je využito knihovnických funkcí strojového učení:

1. Vytvoření kontextu: `new MLContext { FallbackToCpu = true }`
2. Načtení dat do kontextu: `context.Data.LoadFromEnumerable()`
3. Definování pipeline pro analýzu: `context.Forecasting.ForecastBySsa()`
4. Trénink modelu: `pipeline.Fit(data)`
5. Vyt. modelu: `model.CreateTimeSeriesEngine<TimeSeriesInput, TimeSeriesResult>(context)`
6. Vygenerování predikcí: `engine.Predict(input.PredictionHorizon)`

Ve funkci se prochází celým cyklem od přípravy modelu, přes trénování, až po vygenerování výsledku. Implementace byla provedena tímto způsobem, protože získání potřebných dat, které by mohla posloužit pro vytrénování a otestování předpřipraveného modelu, je v tomto případě mimo rozsah práce. U řešené domény se navíc budou průběhy u stejné veličiny pravděpodobně lišit pro různé závodní trati, typy vozů, počasí a dalších možností, proto by nejspíše bylo potřeba připravit takových modelů více. Zvláště by pak narostl počet různých modelů z toho důvodu, že použitá knihovna nedovoluje kombinovat více proměnných v jenom modelu.

Na druhou stranu je tento přístup také poměrně rychlý (i když více o něco více zatěžuje procesor) a výsledek odpovídá aktuální situaci a podmínkám a je naprosto univerzální. V budoucnu rozhodně bude muset být varianta s připravenými modely alespoň prozkoumána. Případně může být použita i jiná knihovna, která kombinaci různých veličin v jednom modelu umožní. Pro přípravu potřebných dat, která by mohla být použita k vytrénování modelu, bude také potřeba odjet množství modelových závodů, jejichž data bude možné použít.

6.8 Jednotkové testy

Při vývoji byly využity jednotkové testy, kterými bylo ověřeno správné fungování samostatných částí aplikace. Zvláště byly otestovány samostatné moduly a části s důležitou logikou. Nebylo testováno fungování uživatelského rozhraní, ani nebylo testování rozšířeno na integrační testy, například nahrazením hry náhradní službou.

Díky tomu, že bylo při návrhu aplikace využito vkládání závislostí (Dependency Injection) pomocí kontejneru z knihovny `Microsoft.Extensions.DependencyInjection`, šlo testovat samostatné části poměrně dobře nahrazením závislostí vytvořenými tzv. `mocky`. Oddělení jednotlivých modulů pomocí zpráv také umožnilo otestovat jejich kompletní fungování s jasným ohraničením definovaných zpráv modulu a kontrolovaných odpovědí. [56]

Pro účely testování nicméně bylo potřeba vytvořit a používat třídu, která by obalila `.NET` třídu `UdpClient`. Tato třída poskytuje funkce nezbytné pro UDP komunikaci, ale z důvodu chybějícího definovaného rozhraní nebylo možné vytvářet její `mocky`. Vznikla tak vlastní třída `UdpClientWrapper`, která implementuje rozhraní umožňující testování. Ta uvnitř obsahuje zabalený `UdpClient`, jehož metody volá, a navíc obsahuje další konstruktory usnadňující své vytváření, které je specifické v navrhované aplikaci.

Výsledný prototyp

7.1 Instalace

Prototyp je distribuován ve dvou variantách, které představují totožnou verzi aplikace, a je tedy možné si zvolit libovolnou z nich. První variantou je publikování ve formě nezabalené (unpacked) aplikace, kterou lze přímo spustit. Druhou variantou je MSIX balíček, který poskytuje jednoduchou instalaci aplikace pomocí instalátoru.

U obou variant jsou dodatečné soubory, které si aplikace potřebuje ukládat na disk, ukládány do složky `C:\Temp\RDE`. Konkrétně ve složce `C:\Temp\RDE\DB` je umístěn soubor databáze a ve složce `C:\Temp\RDE\Logs` lze najít textové soubory s uloženými logy.

7.1.1 Nezabalená aplikace

Nezabalená aplikace je dodávána ve formě archivu obsahující veškeré potřebné soubory aplikace. Archiv stačí rozbalit do zvolené složky. Aplikaci lze spustit dvojklikem na soubor `RDE.exe`, který je dostupný v rozbalené složce.

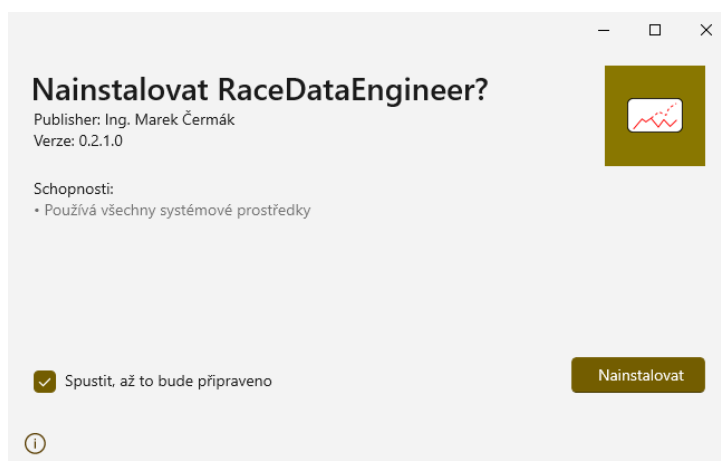
Archiv s se spustitelnou variantou aplikace je dostupný v elektronické příloze, avšak pouze ve verzi x64.

7.1.2 MSIX balíček

Nevýhodou instalace prostřednictvím MSIX balíčku je to, že musí být podepsaný certifikátem. Pro účely prototypu zatím nebyl zařízen důvěryhodný certifikát (který bude potřeba až později při nasazení do MS Store), ale byl vytvořen pouze samo-podepsaný (self-signed) certifikát. Z toho důvodu je potřeba nainstalovat testovací certifikát zároveň s první instalací aplikace. Instalace certifikátu je jednoduchá a je dobře popsána například v blogovém příspěvku „*Installing a test certificate directly from an MSIX package*“ [57]. Detailní popis instalace certifikátu je také součástí uživatelského návodu k aplikaci, který je dostupný v elektronické příloze.

Instalace aplikace se provádí pouze dvojklikem na soubor MSIX balíčku. Příklad, jak vypadá instalační okno MSIX instalátoru, je uveden na obrázku 7.1. Po nainstalování je aplikace dostupná v přehledu ostatních programů v systému windows, kde se může odinstalovat jako jakákoliv jiná aplikace.

Instalační MSIX balíček je přiložen k práci jako součást elektronické přílohy. Vygenerována byla pouze verze x64.



■ Obrázek 7.1 Instalace aplikace pomocí MSIX instalátoru

7.2 Výsledná podoba

Výsledná podoba aplikace vychází z navržených obrazovek a požadovaných funkcí tak, jak byly pro prototyp vygenerovány. Pro ilustraci je ukázána základní obrazovka, obrazovka nastavení a zastávky v boxu v průběhu závodu. Protože si uživatel může zvolit tmavou nebo světlou variantu aplikace, jsou zde pro ilustraci použity obrázky z obou variant.

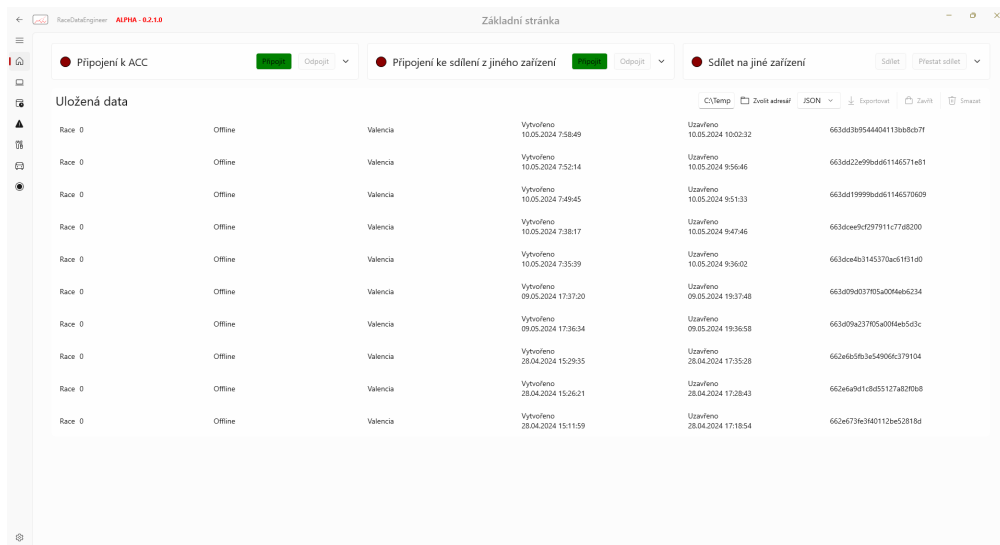
Na základní stránce (viz obrázek 7.2) je v horní části vidět ovládání připojení včetně možných nastavení. Funkčnost je indikována zelenou diodou u konkrétní služby. Ovládací tlačítka jsou příslušně aktivována podle toho, kterou funkci je možné v aktuálním případě vyvolat. Uvedená nastavení jsou v základu sbalena, ale pro sdílení umožňují nastavit potřebné údaje, které se pro každé sdílení mohou lišit.

Pod ovládáním služeb je přehled všech proběhlých sezení, jejichž data jsou uložena a lze je exportovat nebo smazat pomocí definovaného ovládání. U sezení jsou doplněny další informace, aby je šlo identifikovat, například datum a čas, konkrétní trať, a typ sezení. Nastavení je třeba po změně uložit tlačítkem, případně je možné nastavení resetovat do výchozích hodnot. Není potřeba po změně restartovat aplikaci, pouze se změny projeví až pro nové připojení ke hře, pokud je při změně zrovna nějaké aktivní.

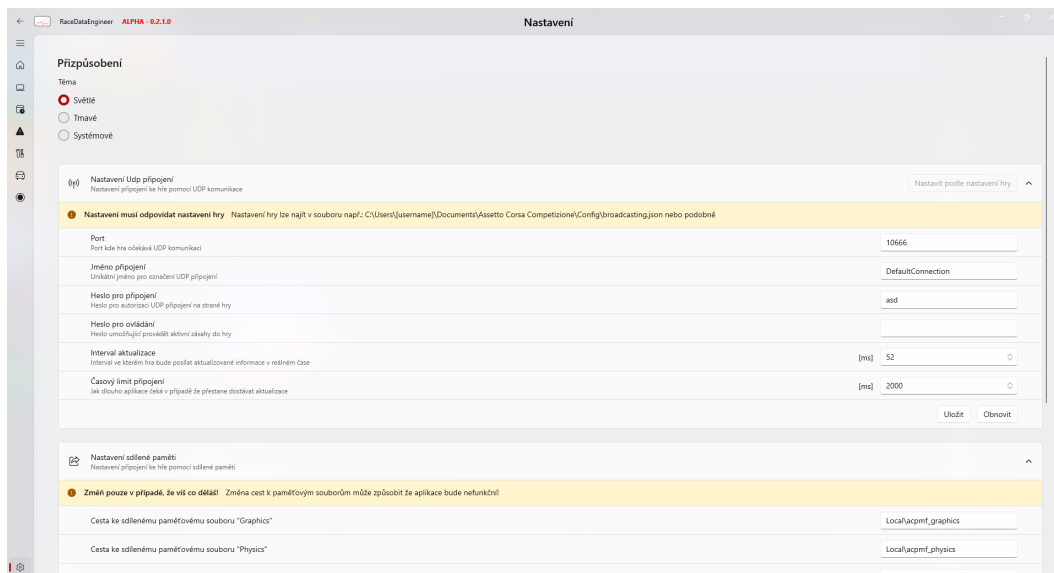
Na stránce nastavení (viz obrázek 7.3) je uveden příklad možných nastavení. Lze nastavovat jak parametry připojení UPD a sdílené paměti, tak třeba i krok a maximální počet prvků v grafech.

U predikcí, s ohledem na popsanou možnost větší náročnosti na výkon, je navíc doplněno více možností jejich nastavení (viz obrázek 7.4). Lze je kompletně vypnout, změnit interval přepočtu, změnit některé tolerance použité u predikce zastávky v boxu, nebo upravit počty bodů ze kterých se predikují časové řady.

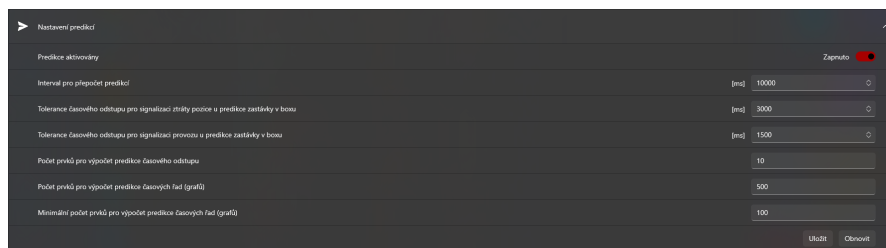
Stránka zastávky v boxu, jak vypadá v průběhu závodu, je ukázána na obrázku 7.5. Zobrazeny jsou potřebné informace a ukazují se grafy s vývoji důležitých hodnot (paliva, tlaků pneumatik, teplot brzd). Zároveň jsou ukázány predikce vývoje hodnot v grafu (viz obrázek 7.6) pomocí čárkované čáry. Predikce zastávky v boxu je ukázána velmi zjednodušeně z důvodu šetření místem, ale význam je jasný „Časová ztráta / počet určitě ztracených pozic / počet pravděpodobně ztracených pozic / velikost provozu po návratu“.



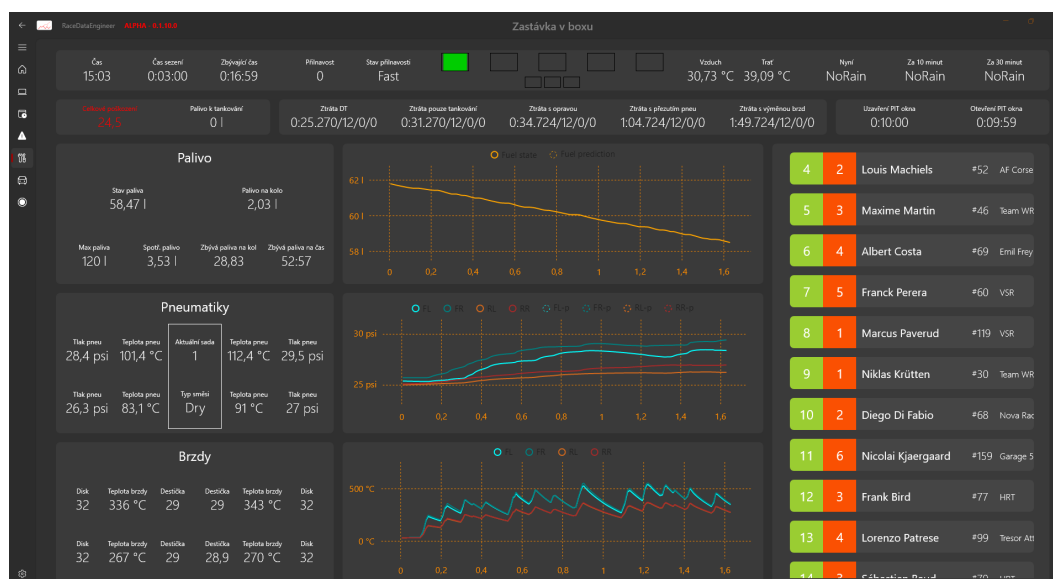
■ Obrázek 7.2 Základní stránka výsledné aplikace – světlá varianta



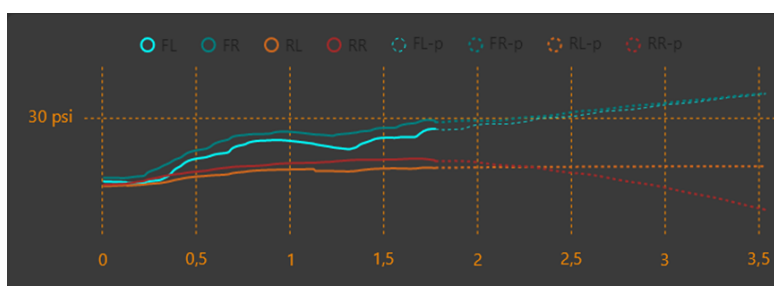
■ Obrázek 7.3 Výsledná stránka nastavení – světlá varianta



■ Obrázek 7.4 Nastavení predikcí – tmavá varianta



■ Obrázek 7.5 Výsledná stránka zastávky v boxu – tmavá varianta



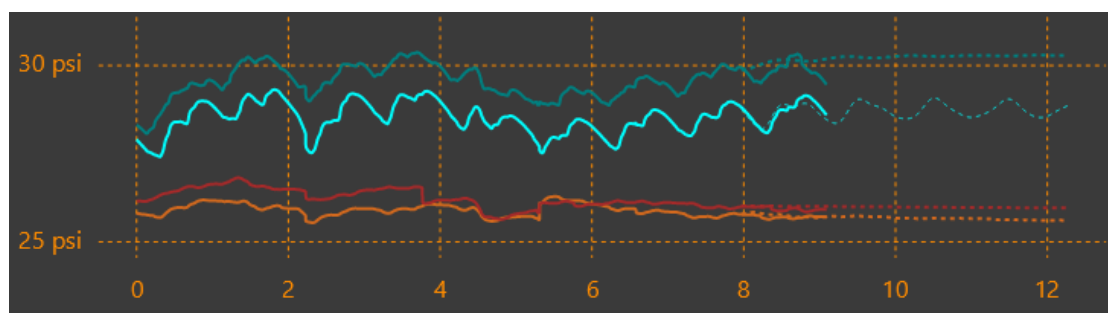
■ Obrázek 7.6 Výsledné zobrazení predikcí v grafu – tmavá varianta

7.3 Ověření predikcí

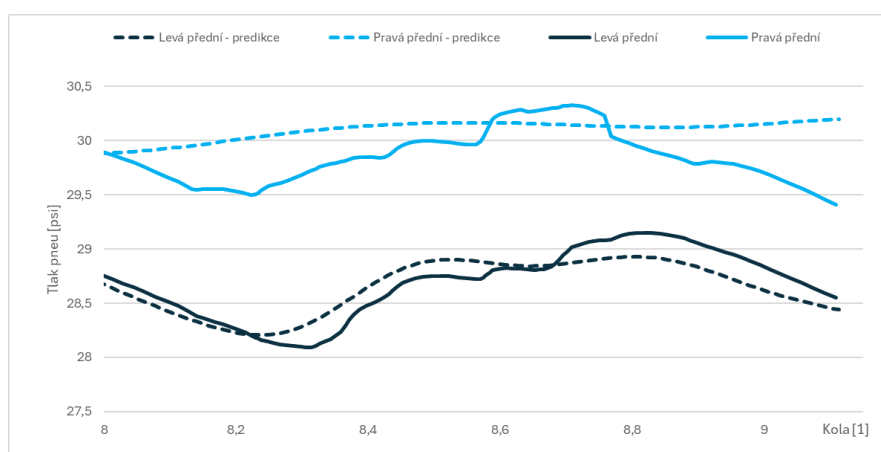
Predikce, které jsou definované výpočtem, byly ověřeny v rámci jednotkových testů. Proto lze predikce časových odstupů, ztráty času a pozic při zastávce v boxu a časový odhad zbývajcího paliva považovat za přesné. Limitací jsou u nich pouze definované předpoklady a způsoby výpočtu.

Jiná situace je u predikcí vývoje hodnot v grafech. Ty jsou generovány strojovým učení a proto je u nich porovnán vygenerovaný průběh s tím skutečným. Pro účely tohoto porovnání se proto v nastavení zvětšil interval přepočtu predikcí na 2 minuty (obvykle stačí na jedno kolo a kousek), aby bylo možné porovnat obě křivky, skutečnou i předpovídanou. Pro porovnání byl zvolen průběh tlaků pneumatik. Na obrázku 7.7 je vidět predikce tlaků pneumatik po několika odjetých kolech v grafu získaném přímo z aplikace. Čáry nejsou příliš vidět, proto byla číselná data obou předních pneumatik (která mají proměnný průběh) extrahována a provedeno názornější srovnání na obrázku 7.8. Zobrazeny jsou pouze ty části grafu, které mají dostupné obě hodnoty – skutečné i predikované.

Čáry predikce jsou u levého předního kola na první pohled velmi blízké skutečnému průběhu, u pravého předního už tak přesně průběhy neodpovídají. Vypočítaná korelace predikce u levého předního kola je 0,95, u pravého předního kola 0,57. Pro hodnocení se musí vzít do úvahy, že řidič nejedí kola vždy naprosto stejně a může se stát, že každé kolo projede jinak (včetně



■ **Obrázek 7.7** Srovnání predikcí a reálných hodnot tlaků pneu - graf z aplikace



■ **Obrázek 7.8** Srovnání predikcí a reálných hodnot tlaků pneu - graf extrahovaných hodnot

nějaké chyby nebo i vyjetí mimo trať). Tato změna se může projevit jako nepřesnost predikce, protože ta s takovou změnou nepočítala. Proto jsou uvedené predikce u prototypu hodnoceny jako dostatečně přesné pro předpokládané využití.

Pro další ověření přesnosti byli uživatelé, kteří testovali aplikaci, požádáni, aby se sledování přesnosti predikcí také zaměřili. Jejich výstup je uveden jako součást testování v části 7.4.

7.4 Uživatelské testování

Pro testování uživatelů byl připraven dotazník a řada scénářů, oboje uvedené v příloze C. Tyto testovací scénáře vycházejí z požadavků a případů použití a reflektují očekávaný způsob využívání aplikace.

7.4.1 Výběr uživatelů k testování

Možnosti použití aplikace jsou velmi specifické, což výrazným způsobem omezuje okruh možných uživatelů pro testování. Vzhledem k povaze aplikace je totiž u testujících uživatelů nutné, aby měli k dispozici hru, na kterou se aplikace připojuje. Případně je možné, aby jim bylo poskytnuto odpovídající a fungující prostředí pro testování autorem práce nebo jiným testujícím uživatelem.

Z těchto důvodů nešlo provést široké uživatelské testování a tak byli osloveni členové komunity, která o vytvoření aplikace měla zájem. Ti představují potenciální budoucí uživatele, kteří vlastní hru Asseto Corsa Competizione a měli by mít, mimo jiné, znalosti závodní problematiky. Nakonec se tak testování zúčastnilo 5 uživatelů.

7.4.2 Průběh testování

Každý testující uživatel obdržel balíček aplikace určený k testování, krátký manuál popisující jak má aplikace fungovat, scénáře které jsou potřeba vyzkoušet a dotazník pro výsledné hodnocení. Dotazníky a scénáře jsou uvedeny v příloze C a manuál pro použití aplikace je přiložen v elektronické příloze.

Testování probíhalo z větší části samostatně, pouze u dvou uživatelů bylo provedeno za účasti a podpory autora aplikace. Uživatelé testující samostatně vyzkoušeli sdílení za pomoci svého vlastního druhého počítače, uživatelé provádějící testování s asistencí ji využili i pro testování sdílení. Během testování nedošlo k žádným problémům, které by znamenaly, že není možné testování provést, nebo není možné provést hodnocení. Od všech uživatelů tak byla získána kompletní zpětná vazba ve formě vyplněných dotazníků a dalších doplňujících komentářů a námětů, které jsou shrnuty v části 7.4.5.

Příklad, jak vypadala konfigurace kokpitu pro testování aplikace u jednoho uživatele, je ukázán na obrázku 7.9.



■ Obrázek 7.9 Ukázka konfigurace simracing kokpitu pro testování

7.4.3 Identifikované problémy při testování

Během testování nedošlo k identifikaci výraznějších problémů s funkčností aplikace. Jeden z problémů, který stojí za zmínku je, že jeden z uživatelů si ztěžoval na přílišnou náročnost výpočtu predikcí na počítač, která způsobovala znatelné záseky hry při jejich přepočtu. Tento problém se u ostatních uživatelů nevyskytoval. Po krátké kontrole bylo zjištěno, že u daného uživatele jen samotná hra vytěžovala CPU na 85–90%, a proto při přepočtu predikcí byl vytížen zcela. Problém je proto ve výkonnosti počítače na straně uživatele, kdy lze pouze doporučit vypnutí predikcí v nastavení a pro jejich výpočet používat druhý počítač, případně používat pouze funkci sdílení a používat aplikaci na jiném počítači úplně.

7.4.4 Zpětná vazba z testování

Ač to nebylo zprvu v plánu s ohledem na jednoduchost ovládání, bylo potřeba vytvořit manuál pro instalaci a používání aplikace, který byl následně doplněn také do elektronické přílohy. Uživatelům totiž nebylo zprvu zřejmé, co přesně od aplikace očekávat, jaké predikce jsou počítány a jak provést jejich nastavení. Po krátkém vysvětlení či přečtení jednoduchého manuálu se ale

uživatelé rychle zorientovali a neměli s pochopením a ovládním aplikace žádný problém. Je tak ale vidět, že by mohlo být vhodné do uživatelského rozhraní aplikace zakomponovat prvky, které při prvním spuštění budou uživateli pomáhat v pochopení aplikace a fungovat jako průvodce po jejích možnostech.

Mezi testujícími uživateli také vznikl spor ohledně jazyka aplikace. Někteří preferovali češtinu, pro jiné byl překlad zbytečný a dali by přednost angličtině. Současný stav, kdy je jazyk aplikace řízen dle nastavení počítače uživatele (česky pro české nastavení, anglicky pro jiné), proto bude potřeba vylepšit a přidáním možnosti přepnout jazyk přímo v nastavení aplikace.

Při posuzování přesnosti predikcí uživatelé sledovali predikovaná data během závodu a porovnávali je později se skutečnými. Uživatelé shodně uvedli, že na začátku závodu, kdy je situace neustálená, soupeři jsou blízko sebe a případně se často předjíždí, a zatím nejsou získaná data v dostatečném počtu, byly predikce nepřesné. V takových situacích predikce časového odstupu neměla význam, predikce v grafech dávaly rozporuplné výsledky a predikce zastávky boxu z podstaty znamenala ztrátu všech pozic. Naopak jiná situace podle uživatelů byla později v průběhu delšího závodu, kdy se uživatelé drželi na ustálené pozici a rozestupy mezi vozy byly výrazně větší. V takových chvílích predikce ukazovaly hodnověrné údaje, které dle nich byly ověřeny, že jsou poměrně přesné.

Všechny další připomínky získané od uživatelů jsou validní a dávají smysl jak z hlediska uživatele, tak z hlediska vývojáře aplikace. Jejich výčet je uveden v přehledu níže.

- Vylepšit zobrazení vlajek, aby byla změna signalizované vlajky postřehnutelná periferně.
- Barevně rozlišit kategorií jezdců / vozidel u pořadí v kategorii v přehledu Aktuální pořadí.
- Přidat graf a predikce vývoje teplot vzduchu a trati.
- Vyznačit interval ideálních tlaků v grafu (nastavitelné uživatelem).
- Automaticky upravovat nastavení UPD v aplikaci také v konfiguraci hry.
- Přepínat obrazovky klávesovou zkratkou, kterou by šlo namapovat na tlačítko na volantu.
- Měnit barvu časového odstupu k soupeři před sebou podle trendu kdy červená je pro narůstající a zelená pro zmenšující se odstup – u soupeře za sebou obráceně.

Uvedené připomínky nebyly zapracovány v rámci této práce, ale byly zařazeny pro implementaci v některé z dalších verzí aplikace.

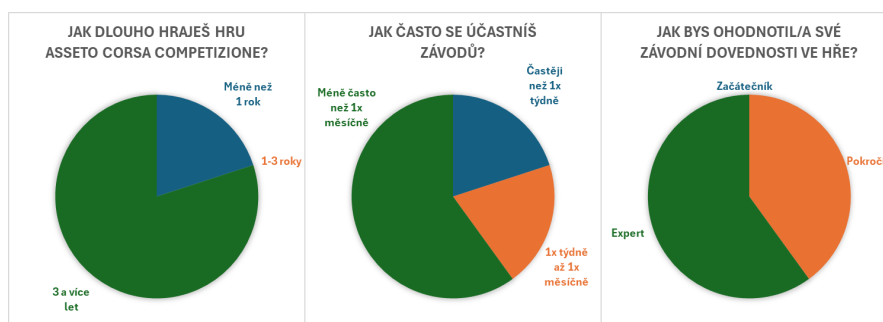
7.4.5 Vyhodnocení testování

Ačkoliv počet testovacích subjektů byl malý (5), jednalo se o část budoucích uživatelů mající potřebné znalosti řešené problematiky, tedy o uživatele znalé. Díky tomu bylo uživatelské testování hodnotné a výstupy z něj dobrým vstupem pro další pokračování ve vývoji této aplikace.

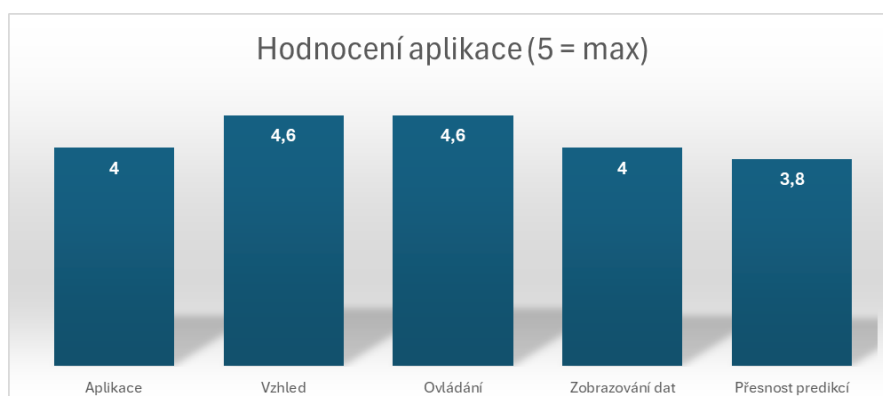
Informace o zkušenostech uživatelů se závoděním a úroveň jejich závodních dovedností, podle odpovědí v dotaznících, jsou znázorněny v grafech na obrázku 7.10. Je vidět, že testující uživatelé se všichni považují za alespoň pokročilé, kdy pouze jeden má méně než 3 roky závodních zkušeností, i když se teď už většinou účastní závodů méně často.

Na obrázku 7.11 jsou zobrazeny průměrně dosažené body k příslušným otázkám v dotazníku. Uživatelé bodově, ale i komentáři, nejvíce oceňovali vzhled aplikace a snadnost ovládní. Naopak menší počet bodů získalo zobrazování dat a přesnost predikcí. Hodnocení aplikace jako celku odpovídá dosaženému hodnocení dílčích částí.

Podle výsledků je potřeba se do budoucna zaměřit na zlepšení zobrazení dat, případně umožnit uživatelům vlastní úpravu zobrazení. U predikcí je hodnocení ve své podstatě zdůvodněno v části 7.4.4 věnující se zpětné vazbě.



■ **Obrázek 7.10** Znázornění zkušeností a dovedností uživatelů



■ **Obrázek 7.11** Průměr udělených bodů z testování

Uživatelské testování ověřilo, že aplikace je funkční a lze ji použít při ostrém nasazení. Nicméně byly identifikovány některé aspekty, které je potřeba do budoucna vylepšit nebo upravit. Testující uživatelé o používání aplikace projevili zájem a rádi by přispěli průběžnými připomínkami k jejímu dalšímu směřování, aby odpovídala jejich představám. To lze považovat za velmi dobrou zprávu a příslib pro budoucí vývoj.

7.5 Budoucnost aplikace

Po dokončení této práce je v plánu na aplikaci dále pracovat a rozvíjet její možnosti a průběžně vylepšovat již implementované funkce a především zobrazení. Proto je níže v několika základních bodech naznačen směr dalšího vývoje. Jedná se o funkcionality, které byly pro prototyp záměrně vypuštěny, dále pak o nedostatky a některé náměty vzešlé ze zpětné vazby uživatelů jako výsledek uživatelského testování. Nakonec jsou to i další nápady, jak aplikaci dále rozvíjet o úplně nové funkce, nebo rozšíření stávajících.

- Umožnit větší konfigurovatelnost aplikace. V současnosti jsou některé věci napevno zabudované do aplikace.
- Synchronizovat nastavení UDP komunikace s konfigurací hry, aby uživatel nemusel provádět ruční změny.
- Umožnit přepínání jazyka zobrazení přímo v nastavení aplikace.
- Přidat do grafů optimální rozsahy dat, které budou možné nastavit uživatelem.

- Zobrazení vlajek udělat výraznější pro lepší detekci změny uživatelem během závodu, a to především periferně.
- Zobrazovat směr větru vůči vozu, ne globálním souřadnicím.
- Vylepšit zobrazení mapy se znázorněním poloh všech vozů včetně jejich identifikace.
- Provést analýzu bezpečnosti sdílení dat a případně implementovat šifrování komunikace.
- Vytvořit lepší výpočetní modely pro predikce, které zvýší přesnosti výsledků a sníží náročnost výpočtů.
- Přidat predikce pro další veličiny a umožnit nastavení všech parametrů predikcí uživatelem.
- Přidat možnost zpětného zobrazení a analýzy uložených dat, včetně možnosti zpětného nahrání exportovaných pro účely zobrazení.
- Umožnit uživatelům, aby si sami mohli změnit rozmístění a velikost zobrazovaných prvků.
- Optimalizovat řízení připojování UDP, aby aplikace sama detekovala, jestli ve hře běží závod, a sama prováděla připojování ke hře ve správném okamžiku.

Závěr

Výsledným cílem práce byla implementace spustitelného prototypu aplikace, která by v reálném čase zobrazovala data ze simulátoru Asseto Corsa Competizione, ukládala je pro pozdější export a pro zvolená data generovala predikce jejich budoucí hodnoty.

V rámci teoretické části byl poskytnut krátký úvod do světa závodů, který čtenáři přiblížil řešenou problematiku, pokud ji dosud neznal, a poskytl náhled na to, jaká data telemetrická a závodní data jsou důležitá. Rešerše podobných aplikací ukázala, že jiné aplikace sice existují, ale žádná z nich přímo nesplňuje očekávání, která byla na tuto aplikaci kladena. Tím bylo ověřeno, že zde prostor pro novou aplikaci existuje. Při analýze API simulátoru byly ukázány obě možnosti, jakým způsobem lze data ze hry získat – UDP přenos a sdílená paměť. Byly krátce vysvětleny obecné principy jejich fungování a následně popsáno, jakým způsobem je lze použít pro získání dat ze hry.

V praktické části byl proveden výběr predikcí vhodných pro implementaci v prototypu a byly popsány výpočty pro jejich získání. Následně byly specifikovány požadavky na aplikaci a očekávané případy užití aplikace s tím, že pro prototyp byla uvedena mírně omezená specifikace toho, co se mělo implementovat.

Na tomto základě byly vybrány potřebné technologie a navržen prototyp aplikace a jeho uživatelské rozhraní. V dané podobě byl prototyp implementován a následně byl také podroben uživatelskému testování na několika osobách ze závodní komunity. Testující uživatelé ověřili funkčnost prototypu, přesnost generovaných predikcí, a poskytli zpětnou vazbu k některým aspektům jeho fungování. Zároveň uvedli řadu doporučení pro budoucí vývoj a vyjádřili zájem o používání aplikace.

Lze tak konstatovat, že cíle práce byly splněny a výstupem práce je skutečně funkční prototyp aplikace. S ohledem na zájem komunity bude aplikace vyvíjena i nadále, kdy budou zahrnuty veškeré zbylé požadavky, které nebyly v prototypu implementovány, a další úpravy vzešlé z uživatelského testování.

Z osobních zkušeností bych rád dodal, že nebylo úplně jednoduché získat podklady k fungování API hry. I ty získané navíc obsahovaly některé chyby, na které se přišlo až posléze. Zároveň použitá technologie WinUI 3 nebyla z mého pohledu ideální volbou, protože se při vývoji bylo často nutné potýkat s řadou nečekaných obtíží. Například podpora samostatného EXE souboru pro aplikaci byla pro tuto technologii přidána až krátce před termínem odevzdání práce. Generování samostatného EXE souboru bylo sice ještě otestováno, ale jelikož nebylo plně funkční, nemohl být samostatný EXE soubor použit jako výstup pro vytvořený prototyp.

..... Příloha A

Popis UDP zpráv

Přehledný popis UDP zpráv je uvedený tak, jak byl získaný ze vzorové aplikace přibalené k de-dikovanému serveru pro ACC [15].

A.1 UDP zprávy pro ACC

REGISTER

```
byte message_type = 0x01;
byte broadcasting_protocol_version = 0x04;
short connection_name_length;
byte [] connection_name;
short connection_password_length;
byte [] connection_password;
int update_interval_ms;
short command_password_length;
byte [] command_password;
```

DEREGISTER

```
byte message_type = 0x09;
int connection_ID;
```

REQUEST_TRACK_DATA

```
byte message_type = 0x11;
int connection_ID;
```

REQUEST_ENTRY_LIST

```
byte message_type = 0x10;
int connection_ID;
```

CHANGE_HUD_PAGE

```
byte message_type = 0x49;
int connection_ID;
short hud_name_length;
byte[] hud_name;
```

CHANGE_FOCUS

```
byte message_type = 0x50;
int connection_ID;
byte is_car_id_changed;
// present only if changed is true, otherwise skiped
unsigned short car_id;
byte is_camera_changed;
// present only if changed is true, otherwise skiped
short camera_set_name_length;
byte[] camera_set_name;
short camera_name_length;
byte[] camera_name;
```

INSTANT_REPLAY

```
byte message_type = 0x51;
int connection_ID;
float session_start_time_ms;
float duration_ms;
int focuser_car_id
short camera_set_name_length;
byte[] camera_set_name;
short camera_name_length;
byte[] camera_name;
```

A.2 UDP zprávy z ACC**REGISTRATION_RESULT**

```
byte message_type = 0x01;
int connection_ID;
byte is_success;
byte is_editable;
short error_message_length;
byte[] error_message;
```

ENTRY_LIST

```
byte message_type = 0x04;
int connection_ID;
unsigned short cars_count;
unsigned short[] car_ids;
```


TRACK_DATA

```
byte message_type = 0x05;
int connection_ID;
short track_name_length;
byte[] track_name;
int track_id;
int track_length;
byte camera_sets_count;
// for each camera set is here this schema
{
    short camera_set_name_length;
    byte[] camera_set_name;
    byte cameras_count;
    // for each camera is here this schema
    {
        short camera_name_length;
        byte[] camera_name;
    }
}
byte hud_pages_count;
// for each hud page is here this schema
{
    short hud_page_name_length;
    byte[] hud_page_name;
}
```

CAR_INFO_DATA

```
byte message_type = 0x06;
unsigned short car_ID;
short team_name_length;
byte[] team_name;
int race_number;
byte cup_category;
byte curretn_driver_index;
unsigned short nationality;
byte drivers_count;
// for each driver is here this schema
{
    short first_name_length;
    byte[] first_name;
    short last_name_length;
    byte[] last_name;
    short short_name_length;
    byte[] short_name;
    byte driver_category;
    unsigned short nationality;
}
```

REALTIME_UPDATE

```
byte message_type = 0x02;
unsigned short event_ID;
unsigned short session_ID;
byte session_type;
```

```

byte session_phase;
float session_time_ms;
float session_end_time_ms;
int focused_car_id;
short active_camera_set_name_length;
byte[] active_camera_set_name;
short active_camera_name_length;
byte[] active_camera_name;
short active_hud_page_name_length;
byte[] active_hud_page_name;
byte is_replay_playing;
// present only if replay playing is true, otherwise skipped
float replay_session_time_ms;
// present only if replay playing is true, otherwise skipped
float replay_remaining_time_ms;
float day_time_ms;
byte ambient_temp;
byte track_temp;
byte clouds_level;
byte rain_level;
byte wet_level;
{LAPTIME} best_session_lap; // defined subpart of message

```

REALTIME_CAR_UPDATE

```

byte message_type = 0x03;
unsigned short car_id;
unsigned short driver_id;
byte drivers_count;
byte gear_id;
float pos_x;
float pos_y;
float yaw;
byte track_location;
unsigned short car_speed;
unsigned short position;
unsigned short cup_position;
unsigned short track_position;
float spline_location;
unsigned short laps;
int delta
{LAPTIME} best_lap; // defined subpart of message
{LAPTIME} last_lap; // defined subpart of message
{LAPTIME} current_lap; // defined subpart of message

```

BROADCASTING_EVENT

```

byte message_type = 0x07;
byte event_type;
short message_length;
byte[] message;
unsigned int time_ms;
unsigned short car_id;

```

A.3 Opakující se části ve zprávách

LAPTIME

```
int lap_time_ms;
unsigned short car_id;
unsigned short driver_id;
byte splits_count;
int[] split_times_ms;
byte is_not_valid;
byte is_valid_for_best;
byte is_out_lap;
byte is_in_lap;
```

Konstanty pro predikci pitstopu

Zde jsou uvedeny konstanty použité pro výpočet predikcí pitstopu. Všechny konstanty nejsou uvedené v dokumentaci, ale byly odečtené, odpozorované nebo odměřené přímo v simulátoru.

Níže uvedené konstanty jsou použité pro výpočet času stráveného na boxovém stání pro provedení požadovaných úkonů.

■ **Tabulka B.1** Konstanty úkonů při zastávce v boxech

Úkon	čas
Zvednutí auta	3s
Tankování 1 litru paliva	0,2 s
Oprava jednotky poškození karoserie	0,141 s
Oprava jednotky poškození zavěšení	30 s
Výměna kol	30 s
Výměna brzd	45 s

Níže uvedené traťové konstanty jsou použité pro výpočet časové ztráty při průjezdu boxovou uličkou. Pozice vjezdu a výjezdu jsou zvolené tak, aby pokud možno ohraničovaly prostor, kde je patrný vliv průjezdu boxem. Těmito místy jsou bod těsně před začátkem brždění do boxové uličky a bod kde po rozjezdu z boxové uličky je již rychlost pokud možno srovnána s ostatními na trati. Převážně se za tato místa považovala ohraničení začátku vjezdu a konce výjezdu z boxu, pokud na trati byla vyznačena.

Tratě mívají délku řádově jednotky kilometrů, proto pozice vjezdů a výjezdů s přesností na deseti-tisíciny představuje setinu procenta délky tratě. To ve své podstatě znamená přesnost okolo 1 dm, tedy naprosto dostatečnou. Čas průjezdu je získán zprůměrováním několika vzorových průjezdů mezi těmito body a s ohledem na povahu měření je zaokrouhlen na desetiny sekundy.

■ **Tabulka B.2** Traťové konstanty pro zastávku v boxech

Id Traťe	Trať	Poz. vjezdu	Poz. výjezdu	Průjezd [s]
3	Monza Circuit	0,9682	0,1108	43,0
1	Brands Hatch Circuit	0,9052	0,0435	31,1
6	Silverstone	0,9205	0,0949	43,8
5	Circuit Paul Ricard	0,9564	0,0784	41,0
4	Misano World Circuit	0,9068	0,1266	52,3
11	Circuit Zandvoort	0,9584	0,0773	26,6
2	Circuit de Spa-Francorchamps	0,9677	0,2166	96,0
8	Nurburgring	0,9106	0,1068	46,8
7	Hungaroring	0,9034	0,1311	47,0
9	Circuit de Barcelona-Catalunya	0,9179	0,1701	47,6
10	Circuit Zolder	0,9076	0,1485	52,5
19	Autodromo Enzo e Dino Ferrari	0,9084	0,0968	52,0
17	Oulton Park	0,9711	0,0907	26,4
16	Snetterton Circuit	0,9356	0,0685	29,0
18	Donington Park	0,9761	0,0874	27,4
21	Circuit of the Americas	0,9500	0,1040	44,1
22	Indianapolis Motor Speedway	0,8456	0,1679	70,5
23	Watkins Glen International	0,9131	0,0943	47,5
13	Mount Panorama Circuit	0,9441	0,0785	42,4
14	WeatherTech Raceway Laguna Seca	0,8702	0,1585	47,9
15	Suzuka Circuit	0,9137	0,0541	41,3
12	Kyalami Grand Prix Circuit	0,9189	0,1101	35,8
24	Circuit Ricardo Tormo Valencia	0,9331	0,1437	40,6

Dotazníky a scénáře pro testování

V této příloze jsou uvedeny dotazníky a scénáře použité pro uživatelské testování aplikace.

C.1 Dotazník

1. Jak dlouho hraješ hru Asseto Corsa Competizione?
 - a. Méně než 1 rok
 - b. 1–3 roky
 - c. 3 a více let
2. Jak často se účastníš závodů?
 - a. Častěji než 1x týdně
 - b. 1x týdně až 1x měsíčně
 - c. Méně často než 1x měsíčně
3. Jak bys ohodnotil/a svoje závodní dovednosti ve hře?
 - a. Začátečník – teprve zjišťuješ co závodění obnáší, sbíráš zkušenosti nebo většinou dojíždíš na konci pořadí
 - b. Pokročilý – v závodech se vyznáš, víš, co je potřeba u auta nastavit, pohybuješ se většinou v půlce pořadí
 - c. Zkušený závodník – máš za sebou řadu závodů, umíš vůz správně nastavit, pravidelně bojuješ na čele startovního pole
4. Jak bys ohodnotil/a tuto aplikaci?
 - a. 1 – špatná
 - b. 2
 - c. 3
 - d. 4
 - e. 5 – skvělá
5. Objevily se při testování nějaké problémy? Pokud ano, jaké?
6. Jak bys ohodnotil/a vzhled aplikace?

- a. 1 – vůbec se nelíbí
 - b. 2
 - c. 3
 - d. 4
 - e. 5 – skvělý vzhled
7. Jak bys ohodnotil/a ovládání aplikace?
- a. 1 – velmi špatné ovládání
 - b. 2
 - c. 3
 - d. 4
 - e. 5 – ovládání je bezproblémové
8. Jak bys ohodnotil/a podobu zobrazování dat?
- a. 1 – data nepřehledná
 - b. 2
 - c. 3
 - d. 4
 - e. 5 – data zobrazena ideálně
9. Jak bys ohodnotil/a přesnost predikcí?
- a. 1 – predikce naprosto nepřesné
 - b. 2
 - c. 3
 - d. 4
 - e. 5 – predikce dávají přesné předpovědi
10. Je něco, co bys na aplikaci změnil?
11. Je nějaká funkce, která ti na aplikaci chyběla, nebo fungovalo jinak, než jsi očekával?

C.2 Scénáře

Uživatelé provádějící testování byli požádáni o provedení testů s pomocí níže uvedených scénářů postupně tak, jak jsou uvedeny. Tyto testovací scénáře vycházejí z požadavků a případů použití a reflektují očekávaný způsob využívání aplikace.

Scénář 1

1. Spustit aplikaci a hru Asseto Corsa Competizione.
2. V aplikaci otevřít nastavení, zkontrolovat údaje pro UDP připojení a upravit je, pokud neodpovídají individuálnímu nastavení hry.
3. Ve hře spustit samostatný trénink.
4. Spustit připojení aplikace na hru a ověřit, že je připojena a kontext dat byl vytvořen.

5. Provést krátkou jízdu v simulátoru a ověřit, že hra zobrazuje všechna očekávaná data na všech stránkách aplikace včetně grafů.
6. Odpojit Aplikaci od hry.
7. Zkontrolovat, že příslušný kontext dat byl uzavřen.
8. Provést export uložených dat do formátu JSON a zkontrolovat exportovaná data.
9. Vypnout aplikaci.

Scénář 2

1. Spustit aplikaci a hru Asseto Corsa Competizione.
2. V aplikaci otevřít nastavení, zkontrolovat údaje pro UDP připojení a upravit je pokud neodpovídají individuálnímu nastavení hry.
3. Ve hře spustit krátký závod, libovolně online či offline.
4. Spustit připojení aplikace na hru a ověřit, že je připojena a příslušný kontext dat byl vytvořen.
5. Přepnout aplikaci na obrazovku zastávky v boxu a zobrazit ji na druhé obrazovce tak, aby byla viditelná i v průběhu jízdy.
6. Během závodu průběžně sledovat predikce zastávky v boxu a kontrolovat jejich správnost.
7. Po dojetí závodu odpojit aplikaci od hry.
8. Zkontrolovat, že příslušný kontext dat byl uzavřen.
9. Provést export uložených dat do formátu CSV a zkontrolovat exportovaná data.
10. Vypnout aplikaci.

Scénář 3

1. Spustit aplikaci a hru Asseto Corsa Competizione.
2. V aplikaci otevřít nastavení, zkontrolovat údaje pro UDP připojení a upravit je, pokud neodpovídají individuálnímu nastavení hry.
3. Ve hře spustit kompletní závodní víkend, libovolně online či offline.
4. Spustit připojení aplikace na hru a ověřit, že je připojena a příslušný kontext dat byl vytvořen.
5. Přepnout aplikaci na obrazovku závodní situace a zobrazit ji na druhé obrazovce tak, aby byla viditelná i v průběhu jízdy.
6. Během jízdy průběžně sledovat predikce časových odstupů a kontrolovat jejich správnost.
7. Po dojetí závodu odpojit aplikaci od hry.
8. Zkontrolovat, že každá část závodního víkendu (trénink, kvalifikace, závod) má vlastní kontext dat a všechny jsou uzavřeny.
9. Provést export uložených dat do formátu CSV a zkontrolovat exportovaná data.
10. Vypnout aplikaci.

Scénář 4 Tento scénář je nutné provést buď s asistencí jiného uživatele, nebo s vlastním druhým počítačem.

1. Spustit aplikaci, případně i na druhém počítači.
2. V aplikaci nastavit port pro příjem sdílených dat a předat jej druhému uživateli nebo nastavit parametry sdílení na druhém počítači.
3. Spustit sdílení pro příjem dat a zkontrolovat že se vytvořil kontext dat bez bližších informací.
4. Vyčkat, než podpora oznámí spuštění sdílení dat nebo na druhém počítači připojit aplikaci ke hře a spustit sdílení.
5. Ověřit, že aplikace přijímající sdílená data zobrazuje všechna očekávaná data na všech stránkách aplikace včetně grafů, a informace v kontextu byly správně vyplněny.
6. Vypnout příjem sdílení a oznámit druhému uživateli, že může vypnout sdílení nebo provést vypnutí sdílení na druhém počítači.
7. Zkontrolovat, že příslušný kontext dat byl na přijímací straně uzavřen.
8. Provést export uložených dat do formátu CSV a zkontrolovat exportovaná data.
9. Vypnout aplikaci.

Bibliografie

1. KUNOS SIMULAZIONI SRL. *Assetto corsa competizione* [online]. 2023. [cit. 2023-04-24]. Dostupné z: <https://assettocorsa.gg/assetto-corsa-competizione>.
2. SRO MOTORSPORTS GROUP. *Fanatec GT World Challenge Europe Powered by AWS* [online]. 2024. [cit. 2024-02-07]. Dostupné z: <https://www.gt-world-challenge-europe.com/>.
3. FÉDÉRATION INTERNATIONALE DE L'AUTOMOBILE. *APPENDIX J – ARTICLE 257A: Technical Regulations for Grand Touring Cars (Group GT3)* [online]. 2024. [cit. 2024-03-24]. Dostupné z: https://www.fia.com/sites/default/files/257a_2024_wmsc_2024.02.28_lg.pdf.
4. SRO MOTORSPORTS GROUP. *2023 SEASON REVIEW* [online]. 2024. [cit. 2024-03-24]. Dostupné z: <https://www.gt-world-challenge-europe.com/images/pdfs/2023%20FGTWCEU-Season%20Review-LR.pdf>.
5. FÉDÉRATION INTERNATIONALE DE L'AUTOMOBILE. *APPENDIX H TO THE INTERNATIONAL SPORTING CODE: Recommendations for the supervision of the road and emergency services* [online]. 2024. [cit. 2024-02-08]. Dostupné z: https://www.fia.com/sites/default/files/appendix_h_2024_published_21.12.2023.pdf.
6. FÉDÉRATION INTERNATIONALE DE L'AUTOMOBILE. *INTERNATIONAL SPORTING CODE* [online]. 2012. [cit. 2024-02-08]. Dostupné z: https://www.fia.com/sites/default/files/2024_international_sporting_code_fr-en_clean_version-_14.12.2023.pdf.
7. CAMPBELL-BRENNAN, Jahee. *Tech Explained: Data Acquisition* [online]. 2021. [cit. 2024-04-20]. Dostupné z: <https://www.racecar-engineering.com/tech-explained/tech-explained-data-acquisition>.
8. RACECAR ENGINEERING. *How do tyre blankets work?* [online]. 2023. [cit. 2024-04-20]. Dostupné z: <https://www.racecar-engineering.com/tech-explained/how-do-tyre-blankets-work>.
9. COLLINS, Sam. *F1 2014 explained: Brake systems* [online]. 2014. [cit. 2024-04-20]. Dostupné z: <https://www.racecar-engineering.com/tech-explained/f1-2014-explained-brake-systems>.
10. SRO MOTORSPORTS GROUP. *Assetto Corsa Competizione / Fanatec GT World Challenge Europe Powered by AWS* [online]. 2024. [cit. 2024-03-24]. Dostupné z: <https://www.gt-world-challenge-europe.com/about/assetto-corsa>.
11. KUNOS SIMULAZIONI SRL. *Assetto Corsa Competizione ve službě Steam* [online]. 2019. [cit. 2024-03-24]. Dostupné z: https://store.steampowered.com/app/805550/Assetto_Corsa_Competizione/.

12. POSTEL, J. *User Datagram Protocol* [online]. 1980-08-28. [cit. 2024-03-29]. Dostupné z: <https://datatracker.ietf.org/doc/html/rfc5405>.
13. EGGERT, L. *UDP Usage Guidelines* [online]. 2017. [cit. 2024-03-29]. Dostupné z: <https://datatracker.ietf.org/doc/html/rfc8085>.
14. ANGELESCU, Silviu. *CCNA Certification: All-in-One For Dummies*. 1st edition. Wiley Publishing, Inc, 2010. ISBN 978-0-470-48962-8.
15. KUNOS SIMULAZIONI SRL. *Assetto Corsa Competizione Dedicated Server* [online]. 2023. [cit. 2023-11-25]. Dostupné z: <https://steamdb.info/app/1430110/info/>.
16. ALBAHARI, Joseph. *C#10 in a Nutshell: The Definitive Reference*. 1st edition. O'Reilly Media, Inc., 2022. ISBN 978-1-098-12195-2.
17. MICROSOFT CORPORATION. *Managing Memory-Mapped Files* [online]. 2010-06-30. [cit. 2024-03-29]. Dostupné z: <https://learn.microsoft.com/en-us/dotnet/machine-learning/tutorials/time-series-demand-forecasting>.
18. MICROSOFT CORPORATION. *Memory-Mapped Files* [online]. 2022-12-14. [cit. 2024-03-29]. Dostupné z: <https://learn.microsoft.com/en-us/dotnet/standard/io/memory-mapped-files?source=docs>.
19. BARBAROSSA, Fernando. *ACC Shared Memory Documentation* [online]. 2023. [cit. 2023-11-25]. Dostupné z: <https://www.assettocorsa.net/forum/index.php?threads/acc-shared-memory-documentation.59965>.
20. PLAYLOGY. *ACC Tools* [online]. 2022. [cit. 2024-03-29]. Dostupné z: <https://acctools.co.uk/home>.
21. RACEDEPARTMENT. *Telemetry Tool for ACC* [online]. 2023. [cit. 2023-11-23]. Dostupné z: <https://www.racedepartment.com/downloads/telemetry-tool-for-acc.34563/>.
22. RACINGGAME TELEMETRY. *Running the Telemetry for Assetto Corsa Competizione* [online]. 2023. [cit. 2023-11-23]. Dostupné z: <https://www.youtube.com/watch?v=nbT2j9xjUVc>.
23. RACINGGAME TELEMETRY. *How a sim race engineer could use this Telemetry tool*. [online]. 2023. [cit. 2023-11-23]. Dostupné z: <https://www.youtube.com/watch?v=Q2t2UpHBdn4>.
24. UNAMEDIA. *Sim Racing Telemetry* [online]. 2023. [cit. 2023-11-23]. Dostupné z: https://store.steampowered.com/app/845210/Sim_Racing_Telemetry.
25. UNAMEDIA. *Sim Racing Telemetry* [online]. 2023. [cit. 2023-11-23]. Dostupné z: <https://www.simracingtelemetry.com/games/ACC>.
26. MRPIG100. *ACC DRIVE Features and Information* [online]. 2022. [cit. 2023-11-28]. Dostupné z: <https://www.youtube.com/watch?v=TdT4gwt5j0Y>.
27. MRPIG100. *ACC DRIVE* [online]. 2023. [cit. 2023-11-28]. Dostupné z: <https://accdrive.com/>.
28. KLARENBERG, Reinier. *Race Element* [online]. 2022. [cit. 2023-11-28]. Dostupné z: <https://www.overtake.gg/downloads/race-element.50578>.
29. KLARENBERG, Reinier. *Race-Element* [online]. 2022. [cit. 2023-11-28]. Dostupné z: <https://github.com/RiddleTime/Race-Element>.
30. HASSAN, Muhammad. *Prediction - Definition, Types and Example* [online]. 2024. [cit. 2024-03-26]. Dostupné z: <https://researchmethod.net/prediction>.
31. CONCEPTA TECHNOLOGIES LLC. *The Best Data Science Methods for Predictive Analytics* [online]. 2019. [cit. 2024-03-26]. Dostupné z: <https://www.conceptatech.com/blog/best-data-science-methods-for-predictive-analytics>.

32. WATTS, Stephen. *What is Predictive Modeling? An Introduction* [online]. 2023. [cit. 2024-03-26]. Dostupné z: https://www.splunk.com/en_us/blog/learn/predictive-modeling.html.
33. BLUMENFELD, Zach. *Predictive Modeling Techniques: Types, Benefits & Algorithms* [online]. 2023. [cit. 2024-03-26]. Dostupné z: <https://neo4j.com/blog/predictive-modeling-techniques>.
34. OPONG, Rich. *What Does Interval Mean In F1? (Timings Explained)* [online]. 2022. [cit. 2024-03-23]. Dostupné z: <https://flowracers.com/blog/what-does-interval-mean-in-f1/>.
35. MICROSOFT CORPORATION. *Microsoft .NET* [online]. 2023. [cit. 2023-12-02]. Dostupné z: <https://dotnet.microsoft.com/>.
36. MICROSOFT CORPORATION. *An overview of Windows development options* [online]. 2023. [cit. 2023-12-02]. Dostupné z: <https://learn.microsoft.com/en-us/windows/apps/get-started/?tabs=winappsdk-winui%2Cnet-maui>.
37. VALKO, Denis. *C#/.NET GUI toolkit comparison* [online]. Brno, 2023 [cit. 2023-12-02]. Dostupné z: <https://is.muni.cz/th/nsr72/>. Diplomová práce. Masarykova univerzita, Fakulta informatiky.
38. ASHCRAFT, Alvin. *Learn WinUI 3.0: Leverage the Power of WinUI, the Future of Native Windows Application Development*. 1st edition. Packt Publishing, 2021. ISBN 978-1-80020-866-7.
39. STONIS, Michael. *Enterprise Application Patterns Using .NET MAUI* [online]. 2023. [cit. 2023-12-04]. Dostupné z: <https://dotnet.microsoft.com/en-us/download/e-book/maui/pdf>.
40. MICROSOFT CORPORATION. *Community Toolkits for .NET Documentation* [online]. 2023. [cit. 2023-12-02]. Dostupné z: <https://learn.microsoft.com/en-us/dotnet/communitytoolkit/>.
41. MICROSOFT CORPORATION. *TemplateStudio for WinUI (C#)* [online]. 2022. [cit. 2024-04-21]. Dostupné z: <https://marketplace.visualstudio.com/items?itemName=TemplateStudio.TemplateStudioForWinUICs&ssr=false#overview>.
42. SERILOG CONTRIBUTORS. *Serilog — simple .NET logging with fully-structured events* [online]. 2023. [cit. 2023-12-02]. Dostupné z: <https://serilog.net/>.
43. SERILOG CONTRIBUTORS. *serilog/serilog* [online]. 2023. [cit. 2023-12-02]. Dostupné z: <https://github.com/serilog/serilog>.
44. RODRÍGUEZ, Alberto. *LiveCharts2* [online]. 2023. [cit. 2023-12-10]. Dostupné z: <https://livecharts.dev/>.
45. MAURÍCIO, David. *LiteDB : LiteDB - A .NET NoSQL Document Store database in a single data file*. [online]. 2023. [cit. 2023-12-18]. Dostupné z: <https://www.litedb.org/>.
46. MAURÍCIO, David. *SQLite* [online]. 2023. [cit. 2023-12-18]. Dostupné z: <https://www.sqlite.org/index.html>.
47. KOHL, Tomáš. *LiteDB vs SQLite: a comparison for .NET developers* [online]. 2022. [cit. 2024-03-25]. Dostupné z: <https://github.com/teekay/litedb-handbook/tree/master>.
48. ARNOTT, Andrew. *MessagePack-CSharp* [online]. 2024. [cit. 2024-04-10]. Dostupné z: <https://github.com/MessagePack-CSharp/MessagePack-CSharp>.
49. MATH.NET TEAM. *Math.NET Numerics* [online]. 2024. [cit. 2024-04-10]. Dostupné z: <https://numerics.mathdotnet.com/>.

50. MICROSOFT CORPORATION. *Co je ML.NET a jak funguje?* [online]. 2024. [cit. 2024-04-10]. Dostupné z: <https://learn.microsoft.com/cs-cz/dotnet/machine-learning/how-does-mldotnet-work>.
51. MICROSOFT CORPORATION. *Microsoft Learn: Design toolkits and samples for Windows apps* [online]. 2024. [cit. 2024-02-11]. Dostupné z: <https://learn.microsoft.com/en-us/windows/apps/design/downloads/>.
52. FIGMA, INC. *Figma: The Collaborative Interface Design Tool* [online]. 2024. [cit. 2024-02-11]. Dostupné z: <https://www.figma.com/>.
53. MICROSOFT CORPORATION. *Messenger - Community Toolkits for .NET* [online]. 2024. [cit. 2024-03-25]. Dostupné z: <https://learn.microsoft.com/en-us/dotnet/community-toolkit/mvvm/messenger>.
54. MATH.NET TEAM. *Curve Fitting: Linear Regression* [online]. 2024. [cit. 2024-04-10]. Dostupné z: <https://numerics.mathdotnet.com/Regression>.
55. MICROSOFT CORPORATION. *Tutorial: Forecast bike rental service demand with time series analysis and ML.NET* [online]. 2024. [cit. 2024-04-10]. Dostupné z: <https://learn.microsoft.com/en-us/dotnet/machine-learning/tutorials/time-series-demand-forecasting>.
56. MICROSOFT CORPORATION. *Dependency injection - .NET* [online]. 2024. [cit. 2024-03-25]. Dostupné z: <https://learn.microsoft.com/en-us/dotnet/core/extensions/dependency-injection>.
57. VLADASEL, Horatiu. *Installing a test certificate directly from an MSIX package* [online]. 2019. [cit. 2024-04-18]. Dostupné z: <https://www.advancedinstaller.com/install-test-certificate-from-msix.html>.

Obsah přiloženého média

readme.txt.....	Stručný popis obsahu média
RaceDataEngineer	
├─ RaceDataEngineer_x64.msix.....	Instalační MSIX balíček aplikace
├─ RaceDataEngineer_x64.zip.....	Archiv s přímo spustitelnou aplikací
└─ RaceDataEngineer_manual.pdf	Uživatelský manuál k aplikaci
src	
├─ RaceDataEngineer.zip.....	Archiv se zdrojovými kódy aplikace
└─ BP_Čermák_Marek_2024.zip...	Archiv se zdrojovými kódy textu práce ve formátu L ^A T _E X
text	
└─ BP_Čermák_Marek_2024.pdf	text práce ve formátu PDF