Univerzita Hradec Králové
Fakulta informatiky a managementu
Katedra informatiky a kvantitativních metod

Mobilní zařízení a senzory ve sportu

Diplomová práce

Autor: Roman Dušek
Studijní obor: Aplikovaná informatika

Vedoucí práce: doc. Mgr. Tomáš Kozel, Ph.D.

Hradec Králové **Duben 2016**

Prohlášení

Prohlašuji, že jsem diplomovou práci zpracoval samostatně a s použitím uvedené literatury.

V Hradci Králové dne Podpis

Poděkování

Rád bych poděkoval panu doc. Mgr. Tomáši Kozlovi, Ph.D. za jeho odborné vedení, podporu a věcné připomínky, které přispěly ke vzniku této diplomové práce.

Anotace

Práce se zabývá analýzou možností využití integrovaných senzorů v mobilních zařízeních pro účel vývoje sledovacích a fitness mobilních aplikací. Poskytuje základní přehled již existujících řešení dostupných na trhu mobilních aplikací a rozebírá jejich hlavní funkcionalitu a výhody. Dále naráží na absenci aplikace tohoto druhu ve vybraném sportovním odvětví. To je důvodem pro výběr právě této sportovní aktivity pro případovou studii, v rámci které bude vyvíjena vzorová aplikace. Také upozorňuje na chybějící možnost vyhodnocení správnosti průběhu sledované sportovní aktivity u již využívaných aplikací, na což je nahlíženo jako na jejich nedostatek. Dále je představena možnost, jak tento nedostatek kompenzovat právě pro vybranou sportovní aktivitu a zároveň tím vybrané sportovní odvětví obohatit o mobilní sledovací prvek, který zde v této podobě dosud nebyl představen.

Annotation

The thesis analyzes the possibility of using integrated sensors in mobile devices for the purpose of monitoring developments and fitness mobile applications. It provides an overview of existing solutions available on the market for mobile applications and analyzes their key functionality and benefits. Further, referring to the lack of this kind of application in a selected sport activity. This is the reason for choosing just this sport activity for case study within which will be developed sample application. Also points to the absence of possibility to evaluate the correctness of performing during sport activity in already used applications, which is considered to be their lack of. It is also presented how to compensate, just for the chosen sport activity, while the chosen sport activity improve on mobile tracking feature that has not yet been introduced in this form.

Obsah

1. Úvod

Mít přehled o své pravidelné sportovní aktivitě je velmi důležité jak pro profesionální, tak i rekreační sportovce. Každý má různá očekávání a cíle, kterých chce dosáhnout. Může se jednat o počet kilogramů, který chceme snížit. Dalším cílem může být nárůst vytrvalosti a tím prodloužení doby, po kterou je sportovec schopný aktivitu vykonávat. Pro někoho je očekávaným výsledkem už jen samotné udržení se v kondici.

Každý z důvodů, ze kterého je sportovní aktivita provozována, má ale své zásady a pravidla. Pokud jsou však zanedbány, pak se požadovaných výsledků docílí v lepším případě po zbytečně dlouhé době a v horším případě se výsledky nedostaví vůbec. Aby k tomuto nedocházelo, je potřeba právě kontrola a možnost sledování své aktivity.

Mobilní telefony již dávno neslouží pouze pro telefonní hovory nebo zasílání textových zpráv. Naopak tyto funkce jsou dnes spíše jen pozůstatkem dob dřívějších a díky nim se ještě stále tato zařízení mohou nazývat telefony. Čím dál více rozšířeným označením pro tuto nezbytnou součást každodenního života je však mobilní zařízení. Dávno se již u těchto elektronických pomocníků sleduje široká škála stále se rozrůstajících služeb, které ani s telefonními rozhovory nemají nic společného.

Velkému nárůstu mobilních aplikací zásadně napomohl rozmach mobilních senzorů, bez kterých by dnes některé telefony nebyly schopny fungovat tak, jak byly navrženy. Kdo si již na moderních mobilních zařízeních zvykl denně pracovat, nejspíše by si již těžko zvykal nato, že po naklopení zařízení na bok se zároveň spolu s ním nepřeklopí i obrazovka, aby měl uživatel stále na očích obraz, tak jak se očekává. Nezůstává však jen u toho, v jaké pozici se zařízení nachází ve vztahu k uživateli. Není problém sledovat, zaznamenávat a dále zpracovávat polohu zařízení ve vztahu k celému okolnímu světu, o čemž pojednali například (Djuknic & Richton, 2001) a dále specificky (Zeimpekis, Giaglis, & Lekakos, 2002).

Informace o tom, kde a v jaké poloze se právě nacházíme, může být důležitá v mnoha ohledech. Může se jednat o naléhavé případy jako například rychlé vyhledání, případně i přivolání lékařské pomoci, ale ve většině případů nejde jen o nouzové situace. Tyto informace slouží obecně k usnadnění práce na zařízení, které díky nim může uživateli poskytnout ještě větší komfort.

1.1 Cíl práce

Smyslem této práce je možnosti plynoucí z integrace senzorů do mobilních zařízení přenést i do oblasti sportu, konkrétně tedy navrhnout možnosti využití pro dračí lodě jakožto oblast, která zatím není v povědomí veřejnosti příliš rozšířená a s tím souvisí i fakt, že podobná aplikace, jako existují pro různé více rozšířené sportovní aktivity, není momentálně pro dračí lodě na trhu dostupná. Oproti ostatním aplikacím tohoto druhu by měl přínos spočívat nejen v průkopnictví na půdě dračích lodí, ale také v poskytnutí zajímavější statistiky, než může nabídnout samostatný externí GPS modul. Ten již na dračí lodi některými posádkami využíván je. Slouží k zaznamenání doby strávené na vodě a zachycení průměrné rychlosti, kterou se loď během tréninku pohybovala. Našim cílem bude nejprve toto umožnit pomocí integrovaného GPS senzoru v mobilním zařízení skrze platformu Android a poté systém navíc rozšířit pomocí dalšího senzoru, a to nejspíše gyroskopu, jehož princip popisuje ve své práci (Person, 1999), o chytré sledování týkající se správného provedení, zde konkrétně náklonu lodě, který je v rámci tohoto sportu podstatný.

2. Analýza současných sportovních aplikací a jejich možností

Takzvané mobilní zdraví[1] je fenoménem poslední doby. A není divu, protože je založeno na dvou oblastech, a to životním stylu ve spojení se světem mobilních zařízení, přičemž každá z těchto oblastí je velkým tématem sama o sobě. Téma zásad zdravého životního stylu tu bylo již dávno před nástupem technických zázraků, které dnes již bereme jako samozřejmost. Nicméně s příchodem těchto technologií se hranice a možnosti řešení posouvají neustále dál. Je to dáno například prudkým rozvojem nových technologií, chytrých telefonů, chytrých hodinek, či chytrých náramků, které se neustále snaží čím dál více a dokonaleji naplňovat potřeby stejně tak vytrvale narůstajícího počtu nových uživatelů, kteří jeví zájem o udržení a sledování svého zdravotního stavu. Fitness zařízení a aplikace tu jsou a zřejmě již navždy budou. Dle odhadů společnosti ABI Research se jen mezi roky 2011 a 2016 poptávka v odvětví sportovních a fitness zařízeních měla zdesetinásobit. (Hamřík, 2012) Jak vypadaly dříve a jaký je současný stav, bude stručně představeno v následujících kapitolách.

2.1. Původ a vývoj

Zařízení pro sledování sportovní aktivity ve svých dřívějších podobách nenabízely tak širokou škálu možností a funkcí jako je tomu dnes. Technologie v té době neumožňovaly dodat tak příjemný grafický vzhled a různé efekty pro usnadnění, nebo zpestření používání zařízení, natož pak možnosti jako propojení s různými servery pro zálohování a synchronizaci nastavení a dat, využívání různých webových služeb nebo spojení s ostatními uživateli pomocí integrace sociálních sítí. Nicméně co se týče samotného účelu, ke kterému se začaly používat, tak již tehdy bylo sledování a vyhodnocení velmi propracované a pro svůj účel naprosto dostačující.

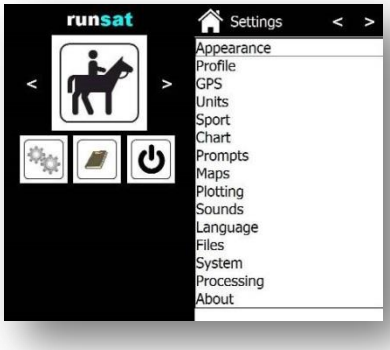
Zařízení tohoto typu postupně vzešla z přenosných GPS navigátorů, které v té době již byly na trhu. Princip těchto zařízení byl totiž velmi podobný. Spočíval v zobrazení aktuální pozice na mapě a dopočítání různých statistik založených na bodech v mapě, jako vzdálenost, rychlost a z toho například odvození zbývajícího času k dosažení cíle.

[1] Také známé pod mezinárodním pojmem mHealth

Příkladem takového zařízení může být přístroj **Forerunner™ 101**. Náramkové běžecké zařízení pro sledování fitness aktivity a trénink. Pomocí vestavěného 12 kanálového GPS přijímače byl schopen vést trénink, vyhodnotit statistické údaje běhu nebo pohybu v terénu a pomocí virtuálního partnera na displeji přístroje mohl uživatel soupeřit se svými vlastními časy na dříve proběhnuté trati. Byla zde dokonce i možnost propojení s PC, odolnost při ponoření do vody a výdrž až 15 hod. v provozu. (Mrkva, 2004) Velmi podobným zařízením byl například přístroj Foretrex.

Obrázek 1 - přístroj Forerunner™ 101 z roku 2004 (flytex.cz)

S pokrokem ve vývoji mobilních aplikací a v integraci senzorů do mobilních telefonů se nevyhnutelně začaly objevovat i mobilní aplikace pro tyto sportovní účely. Příkladem může být zajímavá aplikace **Runsat**. Aplikace byla k dispozici jak pro Windows Mobile, tak pro Windows Phone 7 (placená verze), Samsung Bada a Android OS. Program byl detailně propracovaný a nabízel širokou škálu funkcí a nastavení. Například zde bylo přednastaveno 11 různých sportů, více než 10 jazykových lokalizací a poměrně dost možností personalizace.



Obrázek 2 - ukázka aplikace Runsat (palmserver.cz)

2.2. Současný stav

Díky pokroku techniky není úkolem dnešních sportovních aplikací pouhé snímání surových dat ze senzoru, jejich zobrazení na displeji a uložení do lokálního zařízení uživatele. Mezi běžné schopnosti dnešní sportovní aplikace (sportovního zařízení) patří:

1) zaznamenávat data uživatele
2) zobrazovat statistiky na základě získaných dat
3) učit se na základě předchozích měření
4) hlídat nesprávné chování a radit uživateli
5) komunikovat s trenéry, konzultanty nebo přáteli prostřednictvím sociálních sítí

Některé aplikace potřebují ke své funkci i řadu chytrých periferních zařízení (pedometry, váhy, monitory srdeční frekvence, atd.)

Fitness aplikace je možné rozdělit podle zaměření a to na aplikace pro:

1) sportovní aktivity
2) domácí cvičení
3) správné stravování
4) kombinace předchozích

Dále je možné tyto aplikace rozdělit podle principu funkce a to na aplikace:

1) založené na snímacích senzorech
2) pro práci na základě připravených podkladů
3) wearables[2]

Budou shrnuty aktuálně dostupné oblíbené sportovní aplikace a k nim doplněno shrnutí hlavní funkcionality.

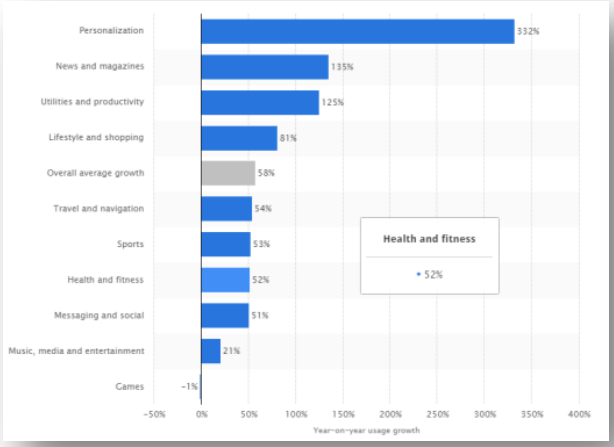
2.2.1 Aplikace založené na snímacích senzorech

Jak vyplývá z kapitoly 2.1, pro tyto aplikace je již od počátku vývoje specifické využití GPS senzoru. Systém byl původně vyvinut především pro navigaci a díky pozdějšímu zmenšení na pouhých několik integrovaných obvodů, čímž se stal více úsporným, našel své uplatnění i ve vývoji chytrých telefonů. Již před několika lety (Jurišica, Vitko,

[2] Jako wearables (nositelná elektronika) se označují miniaturizovaná elektronická zařízení, navržená pro běžné nošení člověkem.

Duchoň, & Kaštan, 2011) vyslovili ve svém článku tvrzení, že systém se může uplatnit ve všech oblastech lidské činnosti a rozsah využití se neustále zvětšuje.

Jak bylo zmíněno v předchozí kapitole, díky mobilním zařízením se GPS již nevyužívá pouze k určení polohy, případně určení trasy, ale dále i k samotnému sledování celého průběhu trasy. Toto dnes zcela běžně vyžívají jak rekreační, tak i profesionální sportovci. Odpovídá tomu i stále rostoucí počet mobilních fitness aplikací. Důkazem toho je statistika (Fastest growing mobile app categories 2015, 2016) ze které vyplývá, že zájem uživatelů o mobilní fitness aplikace v roce 2015 vzrost o 52% oproti roku předchozímu.



Obrázek 3 - statistika z webu statista.com - nárůst oblíbenosti v jednotlivých kategoriích

Mezi nejpodstatnější údaje sledované během jejich sportovního výkonu stále patří počet km uražených za dobu aktivity a dále údaje týkající se rychlosti a času, například maximální/průměrná rychlost, převýšení, celková doba aktivity atd., ale navíc jsou aplikace připraveny na připojení dalších senzorů pro rozšíření funkcionality celého zařízení, například externího snímače tepové frekvence.

Toto vše ale posunuje hranice možností, které mohou tyto systémy nabízet. Ze všech těchto přímých údajů, získaných díky naměřeným hodnotám zvlášť z jednotlivých senzorů, se poté dají v rámci aplikace na míru sestavovat složitější celky a statistiky dle daného účelu, pro který je aplikace vyvíjena. Z pár triviálních hodnot lze sestavit a okamžitě poskytnout uživateli statistiku, která není na první pohled zřejmá a její manuální získání si již vyžádá čas navíc a vyšší míru pozornosti. Příkladem může být údaj o časovém nebo procentuálním úseku, po který byla aktivita vykonávána správně,

oproti celkovému času konání aktivity. Ne po celou dobu je udržováno například správné tempo, nebo dostatečně vysoká tepová frekvence.

V mobilních zařízeních se GPS často objevuje ve formě A-GPS [3] díky možnosti využití mobilních dat k rychlejšímu zjištění polohy než za pomoci pouze dat z GPS družit, která se přijímají pomaleji. (Djuknic & Richton, 2001) tvrdí, že asistované GPS technologie nabízí vynikající přesnost, dostupnost a pokrytí za rozumnou cenu.

Popis vybraných zařízení

Runtastic je sledovací fitness asistent na trhu aplikací pro Android velmi dobře hodnocený (4,5 z 5), který i v neplacené LITE verzi nabízí širokou škálu funkcí, založených na GPS senzoru. Samozřejmostí je sledování doby výkonu, uběhnuté vzdálenosti, převýšení a také měsíční a celkové statistiky z těchto dat sestavené. Dále však nabízí velmi pokročilé funkce vizualizace díky propojení s Google Maps a Google Earth. Díky možnosti připojení externího senzoru pro snímání tepové frekvence jsou k dispozici i pokročilejší možnosti sledování. V neposlední řadě nabízí různé formy propojení s webem.

Obrázek 4 - ukázka aplikace Runtastic z roku 2016 (runtastic.com)

[3] Asistovaná GPS

Runtastic je připraven také na připojení externích senzorů:

1) snímač tepové frekvence
2) snímač tempa na kolo (cadence sensor)

Obrázek 5 - ukázka připojení externího senzoru (runtastic.com)

Obrázek 6 - ukázka snímače tempa pro cyklistiku (runtastic.com)

Velmi podobným asistentem je **Runkeeper**, který je na trhu také hodnocen velmi dobře (4,4 z 5). Oproti Runtastic nabízí automatickou integraci s hudební aplikací v mobilním zařízení. Dále nabízí již integrované tréninkové plány. Mimo to Runkeeper nabízí možnost propojení s jinými populárními fitness aplikacemi a zařízeními, například:

- MyFitnessPal
- Fitbit
- Garmin Connect
- Sleep Cycle

Další velmi známou aplikací tohoto druhu, nabízející prakticky stejnou paletu funkcí, je aplikace **Endomondo**, která se vyvíjí již od roku 2007.

Lehce odlišným druhem aplikací jsou krokoměry. Ty fungují na jiném principu než aplikace předchozí. Odlišnost spočívá hlavně ve využití jiného senzoru než GPS. První zařízení pro jednoduché počítání kroků fungovala na principu kovové kuličky, nebo kyvadélka. Nebyla ale vůbec spolehlivá, protože reagovala na sebemenší otřes. Dnešní systémy jsou pokročilé a využívají **MEMS**[4] a sofistikovaný software pro detekci kroků s vysokou spolehlivostí. Z krokoměrů se později vyvinuly dnešní chytré náramky, ale o těch bude pojednáno až v příslušné kapitole.

Příkladem takového asistenta je aplikace **Pedometer**, jejíž funkce je založena na akcelerometru. Akcelerometr může měřit statické gravitační zrychlení, stejně jako dynamické zrychlení, plynoucí z pohybu, nárazů nebo vibrací (Analog Devices, 2009). Akcelerometry využívají piezoelektrický jev - obsahují mikroskopické krystalové struktury, které při namáhání, které způsobilo zrychlení, začnou generovat napětí. Jiné pracují na principu snímání kapacitance (Juránek, 2007). Hlavní funkcionalitou těchto aplikací je sledování rychlosti a počtu kroků po dobu chůze, z čehož může plynout odhad ušlé vzdálenosti a počtu spálených kalorií na základě zadaných údajů o uživateli, jako jsou pohlaví, věk, výška a váha. Může se zdát, že oproti předchozím aplikacím se zde nejedná o příliš propracovanou aplikaci, nicméně na trhu se setkává s velikou oblíbeností. Hodnocení má aktuálně (4,3 z 5) přičemž přes 80tis. uživatelů dalo aplikaci nejvyšší bodové hodnocení.

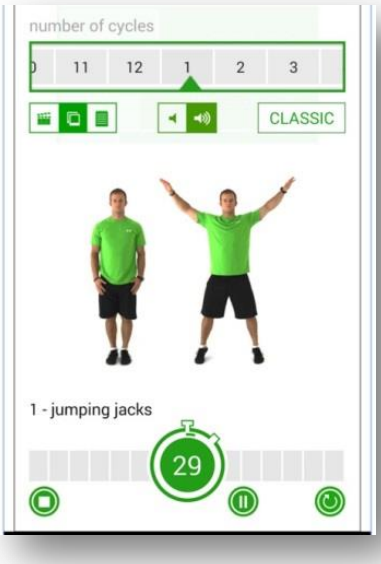
Obrázek 7 - ukázka aplikace Pedometer (play.google.com)

[4] Mikroelektromechanické systémy

2.2.2 Aplikace založené na připravených podkladech

Předchozí druhy aplikací jsou určeny spíše pro venkovní použití. Existují ale i mobilní pomůcky pro aktivitu v domácnosti. Ty by se ale řadily spíše mezi aplikace, které jsou navržené pro práci na základě připravených podkladů (např. prezentací nebo video návodů). Tyto aplikace nebudou rozebrány detailněji, protože jejich podstata nespočívá ve využití mobilních senzorů.

Pouze pro příklad může být uvedena aplikace **7 Minute Workout Challenge**, která uživatele provede procesem střídání 12 vybraných cviků, kde každý ze cviků se provádí intenzivně vždy 30 sekund a následuje 10 sekundová pauza. Cvičení tak zabere pouze 7 minut, ale mělo by být efektivní. Dále existují podobné aplikace pro domácí cvičení, se kterými se dá cvičit déle a poskytují podrobné video návody jak dané cviky provádět co nejlépe.



Obrázek 8 - ukázka aplikace 7 Minute Workout Challenge (play.google.com)

2.2.3 Wearables

V úvodu kapitoly bylo zmíněno, že jako wearables se označují miniaturizovaná elektronická zařízení, navržená pro běžné nošení člověkem. Ty se objevují nejčastěji ve formě náramků a hodinek.

První chytré náramky se vyvinuly z původních krokoměrů, což bylo již zmíněno v kapitole věnované krokoměrům. Postupně byla přidána funkce přenosu dat do

mobilních telefonů nebo do počítače. Následně přibyly další funkce od funkce klasických hodinek až po hlídání zdravého spánku.

Výrazně se vylepšilo softwarové zpracování získávaných dat. Díky všemu výše zmíněnému se postupně z chytrých náramků stávají chytré hodinky, dokonce se zabudovaným GPS senzorem. Jedná se o velmi atraktivní segment trhu a svědčí o tom i fakt, že např. v roce 2014 se těchto chytrých náramků prodalo přes 19 miliónů (Vítek, 2015).

Stručný přehled některých společností zabývajících se touto oblastí:

Fitbit – společnost vyvinula několik generací náramků i se zabudovanou GPS, monitorem srdečního rytmu a řadu dalších funkcí. Fitbit má k dispozici také chytrou váhu (Fitbit Aria), která umožňuje bezdrátový přenos váhy a dalších parametrů do mobilní i desktopové aplikace.

Obrázek 9 - ukázka náramku od společnosti Fitbit (play.google.com)

Jawbone – společnost přišla v roce 2011 se svým stylovým náramkem **Up**. Nejprve měl náramek problémy s výdrží baterie a synchronizací dat. Nové generace ale již tyto nedostatky eliminovaly a na trhu si drží své postavení. Systém by měl fungovat v chytrých telefonech, tabletech a jiných zařízeních využívajících integrované senzory.

Garmin – společnost má k dispozici řadu náramků **Vivofit**. Hlavní výhodou těchto náramků je neobyčejně dlouhá výdrž, která dosahuje až jeden rok bez nutnosti dobíjení. Společnost má i pokročilejší řady, jako jsou Vivofit2 či Vivoactive.

Microsoft Band – představila náramek, který svými funkcemi míří spíše mezi chytré hodinky. Má integrovaný GPS senzor, komunikuje s mobilním telefonem, vypadá velmi zajímavě a dokonce umožňuje odesílat i krátké textové zprávy.

Dalšími společnostmi úřadujícími v této oblasti jsou například Nike, Acer, Misfit, Withings, nebo Polar.

Obrázek 10 - ukázka náramku od společnosti Microsoft (play.google.com)

Zmíněné aplikace jsou velmi propracované a poskytují spoustu zajímavých možností použití. Některé z nich pro výpočet statistik využívají pouze výpočetní algoritmus na základě údajů z jednoho senzoru. Některé využívají hned několik senzorů a jejich spoluprací docílí daleko přesnějších vyhodnocení. Dále jsou k dispozici aplikace poskytující různé formy asistence a navádění pro správné provádění sportovní aktivity, ať už formou textu, nebo v podobě video návodu. Žádná z těchto aplikací ale již nenabízí možnost kontroly, zda je vše v pořádku i v případě, že vše ostatní (jako např. udržovaná rychlost nebo tepová frekvence) je jinak optimální.

Tématem dále tedy bude, jak těchto možností docílit i pro dračí lodě. Tedy obohatit sportovní aktivitu, která byla vybrána jednak obecně z důvodu absence mobilního sledovacího asistenta pro tento sport na trhu mobilních aplikací, ale mimo jiné také z důvodu potřeby sledovat správný náklon lodě po dobu jízdy a v neposlední řadě také z důvodů stereotypu a demotivace sportovců v tomto odvětví. Tyto důvody budou detailněji rozebrány v následující kapitole.

3. Zásady vybrané sportovní aktivity a možnosti využití senzorů

3.1. Motivace

Jak uvádějí (Perič & Dovalil, Sportovní trénink, 2010), slovní spojení sportovní trénink označuje „*přípravu jedince či týmu na soutěže – závody či utkání*“. Tato sportovní příprava má svůj cíl, kterým je především „*dosažení individuálně nejvyšší sportovní výkonnosti ve zvoleném sportovním odvětví na základě všestranného rozvoje sportovce*“ (Perič & Dovalil, Sportovní trénink, 2010). Každé zvýšení výkonnosti nastává díky procesu přizpůsobení se, neboli adaptace. (Perič, Sportovní příprava dětí, 2008) definuje adaptaci jako „*schopnost živého organizmu reagovat na podněty z okolního prostředí*“. Lidské tělo vždy usiluje o udržení stálého stavu vnitřního prostředí, tzv. homeostázu. Vnější podněty tento rovnovážný stav narušují, avšak organismus se jej snaží vracet do původního stavu. Při opakovaném a dostatečně dlouhém působení organizmus přestane usilovat o navrácení do homeostázy a adaptuje se na podněty (Perič, Sportovní příprava dětí, 2008). Z uvedeného vyplývá, že rozvoj konečné výkonnosti nezávisí pouze na fyzické zdatnosti, ale promítá se do ní mnoho souvisejících faktorů. Z toho důvodu jsou dnes k tréninkovému procesu často přizváni odborníci z různých oblastí vědy (Perič & Dovalil, Sportovní trénink, 2010) a stejně tak toto vše může být důvodem do tréninku zahrnout i odborníky elektronické a to mobilní fitness asistenty.

Z toho plyne potenciál, jaký taková zařízení mohou mít, protože jejich využití spočívá hlavně v tréninku, kterým sportovci tráví drtivou většinu svého času. Vrcholy snažení, které představují různé závody, jsou jen jednou za čas, zatímco trénink bývá pravidelnou záležitostí a to v některých případech i několik fází tréninku denně.

V následující kapitole budou kompletně představeny základy vybrané sportovní aktivity. V rámci toho bude probíhat hledání potenciálních možností využití snímacích zařízení.

3.2. Představení sportovní aktivity

Dračí lodě jsou sportovní aktivita, která má dlouhou tradici, hlavně v asijských zemích. V České republice jsou sice novým, avšak dynamicky se rozvíjejícím sportovním odvětvím. Pojmenování vzniklo doslovným přeložením slovního spojení slov dragon a boat, které se používá v češtině také ve formě termínu dragonboat či dragonboating.

Slovní spojení tedy přímo poukazuje na specifickou podobu lodi, která díky tradičnímu zdobení připomíná draka. Důkazem tvrzení o malém povědomí veřejnosti o tomto sportu, které zaznělo v úvodu práce, může být fakt, že v žádné české publikaci dosud není uvedena plná definice dragonboatingu.

Lidé, kteří mají o tomto sportu pouze tušení, ale neznají detaily, mají většinou představu, že jde pouze o rekreační a volnočasovou aktivitu provozovanou pro účely zábavy, například ve formě firemních teambuildingů, školních soutěží, nebo závodů v rámci festivalů pořádaných u vody. O dragonboatingu však můžeme uvažovat jako o plnohodnotném sportu, neboť plně odpovídá definici sportu dle (Bílá kniha o sportu, 2007), kde jsou za sport považovány „*veškeré formy tělesné aktivity, které jsou provozovány příležitostně nebo organizovaně, usilují o vyjádření nebo vylepšení fyzické kondice a duševní pohody, utvoření společenských vztahů či dosažení výsledků v soutěžích na všech úrovních*“.

Plnohodnotnost a perspektivu tohoto sportu dále potvrzuje fakt, že se jedná o Hongkongský národní sport. Hongkong je považován za kolébku moderního dragonboatingu. V 70. letech 20. století se zde totiž nastartovala cesta od tradiční k současné závodní podobě dračích lodí. Důkazem může být více než 300 místních posádek. Rozšíření dračích lodí do celého světa je pak výsledkem snahy o zvýšení cestovního ruchu.

Na téma dragonboatingu můžeme říci, že se jedná o týmový lodní sport, který je prováděn rekreačně i profesionálně, v přírodě na přirozených nebo uměle a účelně vytvořených vodních plochách - kanálech. Kompletní posádku tvoří 22 členů, z toho je 20 pádlujících, kteří pohánějí loď vpřed pomocí jednolistého pádla. Zbylí dva členové jsou bubeník na přídi a kormidelník stojící vzadu. Všichni kromě bubeníka jedou po směru jízdy. Cílem sportovního výkonu v dragonboatingu je co možná nejrychleji projet stanovenou trať v souladu s pravidly tak, aby výsledný čas byl rychlejší, než výsledný čas soupeřů. Jezdí se více závodních tratí, od sprintů na 200m až po tratě dlouhé třeba 7km. (Krauzová, 2014)

Dračí lodě jsou sportem, kde je důležitá nejen síla, vytrvalost a zvládnutí správné techniky, ale také týmová práce a společný cíl. K efektivnímu posunu lodi vpřed je zapotřebí, aby veškeré pohyby všech členů posádky probíhaly synchronně a

nedocházelo k tříštění sil. Toto vše již napovídá, kde všude by se dal hledat potenciál ve využití sledovacích senzorů (měření síly, sledování techniky, synchronizace posádky).

3.3. Hledání potenciálu využití senzorů

V následující kapitole bude vybraná sportovní aktivita rozebrána z různých pohledů, ve kterých se bude hledat potenciál a možnosti využití senzorů. Nejprve z pohledu složení kompletní posádky týmu a dále z pohledu vybavení posádky, nezbytného pro provozování aktivity.

3.3.1. Posádka dračí lodě

Jak již bylo zmíněno v úvodu kapitoly, kompletní posádku tvoří kromě 20 pádlujících také bubeník a kormidelník. Právě dva posledně zmínění mají v lodi velmi specifickou funkci a představují možná místa, mezi kterými můžeme uvažovat umístění senzorů, případně vyhodnocovacího zařízení, protože oproti pádlujícím mají relativně velký prostor pro kontrolu a manipulaci s tímto zařízením. Existují ale i možnosti, jak senzory integrovat dovnitř posádky a to jak umístěním na lavice pro pádlující, tak i samotní závodníci mohou mít senzor umístěn přímo na sobě a měřit tak své osobní tělesné údaje.

Bubeník (drummer)

Bubeník sedí na stoličce na zvýšené špici lodě, svými koleny objímá buben a má hned několik funkcí. Tím nejdůležitějším úkolem je po celou dobu závodu sledovat háčka [5] a údery do bubnu udávat tempo

zbytku posádky. Bubeník ale neurčuje tempo pádlování. Pouze jej kopíruje pro posádku podle háčka, aby mohlo dojit k naprosté synchronizaci pohybů na lodi.

Bubeník vedle bubnovaní také sleduje situaci za lodí a instruuje posádku i kormidelníka předem domluvenými povely, neboť je jediným členem posádky, který jede v protisměru. V neposlední řadě také bubeník svými údery spoluvytváří celkovou atmosféru dračích závodů a motivuje posádku.

Vzhledem k tomu, že je z hlediska pohonu lodě pasivním členem posádky, měla by vždy být brána v potaz jeho váha, aby zbytečně nezatěžoval posádku. Pro možnosti využití sledovacích zařízení na lodi je však zajímavou postavou, protože má prostor k tomu zařízení sledovat a případně rovnou předávat informace do posádky.

[5] Obvykle ten nejzkušenější, sedící na první lavičce a udávající tempo

Kormidelník (helm)

Kormidelník je jediným stojícím členem posádky. Jeho pozice je na zádi lodě, kde má pomocí pružných gum připevněno dlouhé dřevěné kormidlo, díky němuž řídí směr lodi. Ačkoli by se mohlo zdát, že na krátkých tratích nemá kormidelník mnoho starostí, opak je pravdou. Při závodech za lodí vznikají silné vlny, které mohou loď naprosto strhnout z původního směru. Při závodech může výkon kormidelníka ovlivnit výsledek celé posádky poměrně výrazně. Výsledek týmu záleží nejen na výkonnosti pádlujících, ale i zkušenostech kormidelníka, jehož úkolem je navést loď tak, aby nedošlo k vzájemnému kontaktu s další posádkou a také, aby jeho posádka překonávala co nejmenší vlnu.

I přesto, že má kormidelník relativně více starostí než bubeník, můžeme u něj hledat možnosti práce se zařízeními na palubě.

Háčci

Jedná se o dvojici pádlujících, kteří sedí v lodi na první lavičce. Obvykle to bývají ti nejzkušenější, neboť jejich úkolem je nastavit příslušné tempo závodu a to podle potřeby a vývoje závodu průběžně měnit. Nicméně i mezi háčky je určitá hierarchie - jeden z nich je hlavní a ten rozhoduje o tempu. Zároveň to bývá právě hlavní háček, kdo komunikuje s bubeníkem a předává tak informace posádce. Od druhého háčka se tedy předpokládá dokonalá synchronizace s pohyby háčka hlavního, aby obě strany lodě pádlovaly naprosto stejně a nedocházelo k zbytečnému rozhoupání lodě do stran. Jedná se ale již o členy posádky, kteří nemají během jízdy možnost manipulovat se sledovacím zařízením. Jelikož ale právě oni jsou těmi, kdo rozhodují o změnách tempa, pak zde by se dalo nejvíce uvažovat o umístění zařízení alespoň tak, aby ho měli háčci na očích. Krátkým povelem pak jen mohou upozornit bubeníka, že bude následovat například změna tempa.

Zbývající posádka

Zbylých 18 pádlujících členů posádky se snaží přidat svůj díl energie do výsledného pohybu lodě vpřed. I když je zvoleno správné tempo a posádka je v dobré kondici, jízda lodi je efektivní jen v případě, pokud dojde k naprostému souznění pohybů všech členů posádky. Vychýlení frekvence pádlování i jen malé skupinky jednotlivců od zbytku posádky má nepříznivý dopad na pohyb celé lodi.

Pokud má některý člen posádky pocit, že tempo není optimální a bude se pokoušet ostatní pobízet, posádce tím ve finále spíše uškodí. Při závodě i tréninku je potřeba potlačit vlastní pocity a synchronizovat 20 závodníků tak, aby tvořili jednotný tým.

Teprve v takovém případě je pohyb lodi ve vodě maximálně efektivní. U těchto členů posádky se již nedá uvažovat o možnosti, že by nějakým způsobem vůbec věnovali pozornost sledovacímu zařízení. Zato však by se dalo uvažovat o umístění některých typů senzorů přímo na tělo, nebo vybavení jednotlivých členů a snímat statistiky individuálně, aniž by samotná osoba, nesoucí dané zařízení, musela svou pozornost upírat na něco jiného, než na svůj záběr.

3.3.2. Vybavení pro provoz sportovní aktivity

Výše byly uvedeny možnosti, jak se senzory naložit z pohledu možného rozmístění zařízení mezi jednotlivé členy posádky. Nyní bude představeno vybavení pro účely tohoto sportu a pro účel analýzy možností, jak by se senzory daly použít ve vztahu právě k vybavení posádky.

Vybavení pro posádku dračí lodě i ve své základní podobě sice zabere více místa, než osobní automobil a také svou váhou znemožňuje manipulaci v malém počtu lidí[6]. Zato výčet všech položek je dosti krátký. Mezi základní a nezbytné patří pouze dračí loď, kormidlo a sada pádel. S tímto se již posádka obejde a je možné vyrazit na vodu. Pro jiné účely než trénink nebo závod není nezbytně nutné vozit s sebou například vybavení pro bubeníka, což je sedačka, buben a palička. Další možnou součástí výbavy, kterou již začaly využívat i některé české posádky, je cox box[7].

Dračí loď

Dračí loď má tvar otevřené kánoe, přesně danou délku a minimální váhu. Součástí předepsané podoby závodní lodě je typická ozdoba v podobě dračí hlavy na špici a ocasu na zádi. To ale nebude pro tuto práci podstatné. Pro tvorbu sportovních lodí se vzorem pro IDBF [8]staly tradiční hongkongské lodě. Tyto modely dračí lodě mají přesně dané parametry, které musí splňovat alespoň na minimum.

[6] Historicky ověřeno, že loď je bezpečně možno přemístit minimálně v počtu 5 – 6 osob
[7] Elektronické zařízení známé z veslování obsahující snímač rychlosti, mikrofon a reproduktor
[8] International Dragon Boat Federation – mezinárodní federace dračích lodí

Parametry dračí lodě

- délka lodě bez ozdob: 12,4m
- šířka lodě v nejširším bodě: 1,16m
- hloubka lodě: 0,54m
- váha lodě: 250kg
- počet lavic: 10 – 20 dle kategorie (small / standard boat)

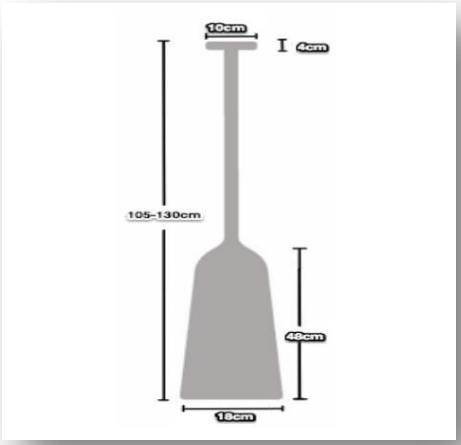
Obrázek 11 - znázornění dračí lodě včetně příslušenství (seagull.com)

Dalo by se tedy uvažovat nad umístěním zařízení přímo na lavici vedle pádlujícího. Otázkou však je, zda by toto mělo smysl, protože na lavici by zřejmě nešlo o senzor snímající přímo tělesné údaje pádlujícího, ale spíše o senzor který nějaká data přijímá a vyhodnocuje. V takovém případě ale senzor nemusí být mezi pádlující posádkou, ale jako lepší umístění se jeví pozice blíže někomu, kdo bude naměřená data dále reprodukovat, například háčci s bubeníkem, nebo kormidelník. Poté se dá uvažovat nad umístěním zařízení přímo na podlahu lodě. Překážkou může být voda, která se během jízdy do lodě dostává a postupně přibývá. Tomu tak bývá při odhazování vody pádly, nebo při jízdě dvou lodí vedle sebe. Poté se natolik zvedne vlna, kterou loď proplouvá, že voda začne přesahovat přes borty. Při umístění na záď ke kormidelníkovi není potřeba řešit ochranu zařízení proti vodě. Kormidelník totiž stojí na vyvýšené ploše a i v případě, že je v lodi značné množství vody, nikdy hladina nedosáhne takové úrovně, aby zařízení ohrozila. V jiném případě by bylo potřeba hledat buďto vodotěsné zařízení, nebo tento problém řešit jinak, například vložením do vodotěsného krytu. Pokud máme k dispozici zařízení, kterému voda nevadí. Pak bychom mohli senzor umístit dokonce i na venkovní stěnu lodě. Muselo by se ale počítat s velmi pevným upevněním a vymyslet zajištění pro případ, že by senzor neunesl nápor průtoku vody a došlo k odtržení od stěny. Příkladem takového senzoru by mohl být například snímač vlhkosti pro sledování

ponoru a v porovnání se senzorem i na druhé straně i možné sledování náklonu celé lodě.

Pádlo

Rozměry a tvar pádla jsou taktéž specifikovány a je možno je dohledat v oficiálních dokumentech IDBF. Pádlo pro dragonboating lze, podobně jako jiné typy pádel, rozdělit na tři pomyslné části. Odspodu jsou jimi list, žerď a hlavička. List musí mít hladký povrch. Žerď nesmí být nijak zahnutá a
její průměr by měl být maximálně 35mm. Hlavička pádla může mít volitelný tvar.



Obrázek 12 - rozměry pádla dle specifikace IDBF (sabahdragonboat.com)

Jak bylo zmíněno výše, máme k dispozici tři části pádla, u kterých můžeme prozkoumat možnosti umístění senzorů.

Žerď a hlavička

Úchop pádla vypadá tak, že horní ruka svírá hlavičku a spodní ruka svírá žerď v její dolní polovině, tedy polovině blíže k listu. Většinou se nepoužívají žádné neoprenové rukavice proto, aby závodník příliš neztrácel cit pro pádlování. Z toho důvodu by se dalo uvažovat o možnosti měření tepové frekvence závodníka z jeho rukou. Existují sportovní zařízení, například rotopedy, které mají v řídítkách takový senzor zabudovaný. Na displeji se poté zobrazují statistiky dopočítané z naměřené tepové frekvence, například počet spálených kalorií. Toto by se dalo využít i zde, pokud by byl k dispozici senzor, nejlépe v podobě pásku kolem žerdi nebo hlavičky, který by naměřené hodnoty odesílal ke zpracování. Předpokladem by ale muselo být, že

přítomnost senzoru nebude pro závodníka rušivým elementem. Pokud je závodník zvyklý na určitou šířku žerdi, mohlo by takové rozšíření o snímací pás pod rukou být nepříjemné. Takový senzor by tedy musel být dostatečně tenký.

List

Umístění senzoru na list by již vyžadovalo odolnost proti vniknutí vody. Dále by stejně jako v předchozím případě mělo jít o senzor, který by svou přítomností nenarušil samotné pádlování. V tomto případě aby se nezměnil příliš tvar listu a neovlivnil tak pohyb pádla vodou v průběhu záběru. Pokud je osoba zvyklá na určitou velikost a tvar listu, dlouhou dobu trvá, než si osvojí záběr s pádlem kde je i poměrně malá odlišnost. Například pokud si pořídí nové pádlo s třeba jen o milimetr větší šířkou, nebo pokud má list lehce odlišný tvar. Než si i na tak malou změnu osoba zvykne, pádlo se při záběru chová jinak, má tendenci v ruce vibrovat, odskakovat do stran a cákat při zápichu pádla do vody.

Pokud by tyto předpoklady byly splněny, dalo by se uvažovat o možnosti měření síly záběru. Síla záběru je přímo úměrná jednak tlaku, který je vyvíjen na list pádla na straně záběru a dále také ohybu celého listu ve směru proti záběru. I když se to může zdát podivné, sportovní pádla jsou takto navržena. Při záběru se list pádla vždy lehce ohýbá. Je tomu tak kvůli odlehčení záběru. Pokud by list držel pevně svůj tvar po celou dobu záběru, záběr by byl daleko namáhavější. Důsledkem toho je rychlejší ztráta síly a pro netrénované jedince většinou znamená dřívější nástup bolesti ramen. Dále by pak bylo potřeba vyřešit problém, jak takto nasnímaná data přenést do jednotky pro vyhodnocení údajů.

Cox box

Jedná se o elektronické zařízení, známé hlavně ze světa rowingu[9]. V základní verzi kombinuje monitoring frekvence záběrů a celkového uběhlého času spolu s megafonem a mikrofonem. Mikrofon má u sebe kormidelník a dává tak pomocí megafonu pokyny dovnitř posádky. Jsou i jiné typy, kde je místo megafonu využita soustava propojených reproduktorů, které jsou rozmístěny na lavičky mezi členy posádky. V rozšířené verzi obsahuje navíc i senzor pro měření rychlosti. Ten funguje na principu vrtulky ponořené ve vodě. Rychlost otáčení vrtulky je potom přímo úměrná rychlosti jízdy. Oproti výše

[9] Mezinárodní označení pro veslování

sepsanému vybavení není potřeba přemýšlet nad umístěním senzoru, protože senzor je již součástí celého modulu, stejně tak jako displej pro samotné zobrazení. Není zde tedy již potřeba návrh způsobu přenosu naměřených údajů do zobrazovacího zařízení, jako v předchozích případech.

Toto by se kromě rowingu dalo využít i v jiných vodních sportech. Konkrétně sportovní aktivita, která je tématem této práce se jeví jako ideální kandidát pro využití tohoto zařízení a to z důvodu velkého počtu členů posádky a délky lodě. Běžně se totiž stává, že povely od háčka již nejsou slyšet v zadní polovině posádky. Pokud tedy přijde povel na zvýšení tempa a zareaguje pouze polovina posádky, je zásadně porušena jedna z hlavních zásad správné jízdy na dračí lodi a to synchronizace všech členů, což má negativní dopad na jízdu. Zvýšené tempo pádlování pak již nevede ke zvýšení rychlosti jízdy, ale pouze k rychlému vyčerpání těch, kteří na povel stačili zareagovat.

Obrázek 13 - vyhodnocovací a zobrazovací jednotka cox boxu (commercialrc.ie)

Pokud by ale bylo požadováno více, než nabízí toto zařízení a uvažovalo se nad zapojením dalších senzorů do tréninku, například snímat navíc i kmitání přídě lodi senzorem umístěným na špici, jehož hodnoty neumí přijímat vyhodnocovací modul cox boxu, stejně by bylo potřeba navrhnout způsob, jak a kam data zasílat k vyhodnocení. Tím by ale nastala situace, kdy je sice využito více senzorů, zato však již dvě vyhodnocovací zařízení, což není ideální. Lepším řešením by bylo navrhnout způsob, jak využít co největší množství senzorů a tím docílit vyhodnocení široké škály naměřených statistik, ale zároveň data ze všech senzorů shromažďovat v pouze jednom centrálním zařízení, ze kterého by se po ukončení aktivity získala všechna data naráz.

3.4. Základní model využití integrovaných senzorů

Ze všeho výše uvedeného vyplývá, že ideálním řešením by bylo navrhnout model využití senzorů integrovaných v běžně dostupném mobilním zařízení. Naměřené hodnoty by tak pro vyhodnocení nebylo potřeba nikam zasílat. Vyhodnocení a zobrazení dat by tedy probíhalo přímo na daném zařízení. Pokud by tento základní model nestačil, bylo by i přesto možné systém dále rozšiřovat a to jak na straně snímacích senzorů, tak na straně vyhodnocovací části, které budou nyní prozkoumány.

3.4.1. Výhody využití mobilního zařízení

- více senzorů společně na jednom místě
- nasnímaná data jsou zpracována ihned na místě
- zařízení je dnes již rozšířené a běžně dostupné široké veřejnosti
- není potřeba pořizovat a instalovat žádné el. příslušenství
- mobilní OS nabízí API pro usnadnění přístupu k senzorům
- snadné sdílení výsledků po uložení do zařízení
- možnost případného rozšíření o externí služby

Pokud by bylo nutné navrhnout složitější systém vyžadující senzory a funkce, které tento základní model nenabízí, dalo by se uvažovat nad využitím externího senzoru nebo využitím některých webových služeb pro přenos dat na externí server, kde by mohlo proběhnout komplexnější zpracování a zobrazení dat než pouze v samotném mobilním zařízení.

3.4.2. Rozšíření na straně snímacího zařízení

Potřeba rozšířit základní model o externí senzor může nastat v případě, že neexistuje mobilní zařízení, které by mělo integrovaný senzor, který potřebujeme pro navržený model využití.

Typický výběr senzorů integrovaných v běžně dostupném mobilním zařízení:

- GPS – senzor pro určení polohy zařízení
- Gyroskop – senzor pro určení úhlové rychlosti při otáčení ve 3 osách
- Akcelerometr – senzor pro určení zrychlení při pohybu ve 3 osách

Dále také například

- Gravitační senzor
- Senzor orientace
- Magnetický senzor
- Tepelný senzor
- Světelný senzor

Jsou ale i senzory, které jsou potřebné obzvláště pro aplikace využitelné právě v rámci sportovních aktivit, ale nejsou k sehnání v integrované podobě. Typickým příkladem využití externího senzoru je připojení snímače tepové frekvence. Tyto senzory mají nejčastěji podobu hrudního pásu nebo náramku. Data z těchto senzorů se nejčastěji vyhodnocují a zobrazují ve snímacím zařízení, které má podobu hodinek.

Obrázek 14 - snímač tepové frekvence společnosti Sigma (vlastní snímek)

Pro náš model by se ale hodilo data přijímat mobilním zařízením. Zpracování dat by poté probíhalo hromadně spolu s daty přijatými z integrovaných senzorů daného mobilního zařízení, buďto přímo v zařízení, nebo by se sada všech dat odeslala pomocí webové služby na externí server, kde by proběhlo teprve vyhodnocení a případně i prezentace zpracovaných dat. Přenos dat ze senzoru do mobilního zařízení by mohl probíhat pomocí technologie BT Low Energy[10].

[10] Technologie pro úsporný přenos dat mezi snímacím a přijímacím zařízením

3.4.3. Rozšíření na straně zpracování dat

Jak bylo zmíněno výše, v případě že nastane potřeba navrhnout složitější systém s širší škálou možností, než nabízí základní model, je zde možnost model rozšířit i na straně zpracování dat. Taková potřeba může nastat, pokud je vyžadován například jeden z následujících bodů:

- potřeba vyššího výpočetního výkonu, než které nabízí MZ
- potřeba využít robustnější statistický software
- potřeba mít data lépe zálohovaná
- potřeba zpřístupnit data veřejnosti
- potřeba sjednotit data z více mobilních zařízení

V opačném případě pak data zůstávají uložená pouze v mobilním zařízení a případné zálohování poté musí být řešeno jinou formou. Jinak hrozí ztráta uložených dat v případě odcizení, nebo poruchy zařízení. Pokud by se tedy základní model rozšířil a naměřená data by se rovnou pomocí webové služby zasílala na externí server, pak by tím výše zmíněné bylo vyřešeno a data byla bezpečně uložena. Spolu s tím by toto řešení přineslo i další výhody. Zpracovaná data by ležela na serveru přístupném pomocí sítě, což nabízí zároveň možnost k datům přistupovat z více míst. Toho by se dalo využít hlavně v rámci kolektivních sportovních aktivit, například:

- jednotlivci odesílají svá data na oddílový server
- na serveru probíhá srovnání všech dat od jednotlivců
- trenér stahuje výsledky živě do svého zařízení
- týmové prezentace a společné hodnocení nad jednou sadou dat

3.5. Reálný model využití pro analýzu a návrh aplikace

V úvodu kapitoly byla představena vybraná sportovní aktivita. Následovala analýza potencionálních možností využití senzorů v rámci vybrané aktivity, a to jak z pohledu složení posádky týmu, tak z pohledu nezbytného vybavení pro účely provozování této aktivity. V průběhu se dospělo k závěru, že jako ideální se jeví řešení, kdy máme k dispozici seskupení všech potřebných senzorů na jednom místě, například v podobě mobilního zařízení s integrovanými senzory. Postupně se dospělo k podobě základního modelu využití integrovaných senzorů a shrnutí jeho výhod. Na závěr byly představeny možnosti, jak tento základní model dále rozšiřovat.

Nyní bude následovat popis reálného procesu sportovní aktivity, na který bude navazovat zbytek práce, a požadavků, které si vybraná aktivita klade na sledované údaje. Z toho by mělo také vyplynout konkrétně, pro co by bylo možno využít navrhované sportovní zařízení. Cílem je navrhnout jednoduché řešení, které naplní požadavky při udržení míry složitosti na rozumné úrovni.

3.5.1. Typický průběh sportovní aktivity

Na úvod bude vysvětleno, jak vypadá klasický dračí trénink. V první řadě se nalodí celá posádka, kromě kapitána. Ten nejprve zkontroluje pohledem přímý náklon lodě, zda není příliš zatopena příď lodě. Příď by měla být minimálně stejně zanořená jako záď lodě, spíše je vždy lepší, když je o něco výše nad hladinou než záď. Je tomu tak z důvodu odporu vody, kterou loď musí během jízdy překonávat. Pokud by příď byla příliš zatopena, odpor vody, kterou před sebou loď při jízdě hrne, je poté zbytečně veliký. V případě, že náklon této zásadě neodpovídá, probíhá přeskupování posádky. Tomu tak může být v případě, že se na trénink nesejde celá posádka a posádka v přední části převáží zadní polovinu. Řešením poté je zaplnit více zadní lavice a některé přední nechat volné. Pokud již je tento přímý náklon v pořádku, nalodí se i kapitán. V tu chvíli je posádka kompletní a každý již zaujímá své místo dle následujících zásad:

- sedí co nejvíce ke své straně opřený o boční bort
- jedna noha je natažená dopředu a zapřená pevně o opěrku
- druhá noha je ohnutá pod lavici, na které sedí a také je pevně zapřena
- bubeník sedí rovně na stoličce a opírá se o buben
- kormidelník se postaví co nejvíce doprostřed, aby loď nepřevažoval

V tuto chvíli by se nikdo neměl hýbat a loď by již měla být vyvážená. To kontroluje pohledem kormidelník zezadu. Tak jako má své zásady přímý náklon lodi, stejně tak boční náklon by měl být dle zásady a to tak, aby obě strany byly stejně vysoko nad vodní hladinou. Toto je jednak z důvodu, že bočně nevyvážená loď má tendenci stáčet se více k jedné straně a poté musí kormidelník více tlačit na kormidlo v příslušném směru, což vede ke znatelnému brždění lodě. Dalším důvodem je, že pádlující posádce se hůře zabírá, pokud je příliš vysoko nad hladinou a má tak zbytečně daleko k dosáhnutí pádlem do vody. Stejně tak se hůře pádluje, pokud má závodník vodu naopak příliš blízko a musí tak ruce přikrčovat, čímž porušuje zásady správného pádlování.

Obrázek 15 - ukázka špatně vyvážené dračí lodě (infotrebon.cz)

Vše výše zmíněné je tedy nyní kontrolováno pouhým pohledem a lidským odhadem. Navržený systém by toto měl být schopen spolehlivě určit. Když je toto v pořádku, posádka vyráží na vodu. Začíná se volným tempem, kdy se do záběru nevkládá skoro žádná síla. Když se posádka takzvaně zahřeje, což je minimálně po 500m, začínají se jezdit úseky při střídání tempa. Tempa se dělí dle tratí, ve kterých se oficiálně závodí. To jsou tratě na 500m a 2000m, ale také sprint na 200m. Klasickými úseky jsou 100-200m tzv. pětistovkovým tempem, což je dobré na začátku pro synchronizaci pádlujících. Dále se jezdí úseky pro trénování správného tempa, například 500m tempem na 2km.

Stejně jako byl náklon kontrolován na začátku, je potřeba aby kormidelník toto sledoval po celou dobu tréninku. Další možností, která se nabízí pro sledování navrženým systémem, je tedy správnost zvoleného tempa. Když se zvolí tempo na 2km, znamená to, že se jede tempem, kterým by se jel závod na 2km tak, aby se nejelo příliš pomalu a zároveň, aby síly závodníkům nedošly příliš brzy. Toto si posádka také hlídá sama a vždy po úseku následuje debata, zda tempo bylo vyhovující. Toto by tedy také mělo být pod kontrolou mobilního asistenta.

3.5.2. Požadavky na využití senzorů

Nyní bude následovat shrnutí možností, jak tyto požadavky naplnit a zda k tomuto účelu postačí základní model, který má v sobě integrované všechny potřebné senzory a nebude potřeba rozšiřovat o externí systémy.

V kapitole o základním modelu využití mobilního zařízení byly shrnuty základní senzory, které bývají integrovány ve dnes běžně dostupných mobilních zařízeních. Úkolem nyní bude určit, které z těchto senzorů bychom mohli využít pro naplnění potřeb, plynoucích z vybrané sportovní aktivity. Z předchozí analýzy tedy vyplývají následující potřeby:

- kontrolovat vyvážení lodě
- kontrolovat správnost zvoleného tempa

V porovnání s ostatními běžně dostupnými sportovními asistenty by navíc mohly být doplněny i další požadavky, které budou užitečné i pro vybranou aktivitu:

- měřit celkový čas tréninku
- sledovat polohu a celkovou vzdálenost
- určovat rychlost pohybu lodi

3.5.3. Shrnutí senzorů vybraných pro návrh vzorové aplikace

Pro sledování vyvážení by se hodil některý z integrovaných senzorů pro zjišťování polohy zařízení. Mezi takové patří například gyroskop, nebo gravitační senzor. Gyroskop měří úhlovou rychlost otáčení ve třech osách. Mohl by tedy poskytnout informace o tom, že se vyvážení lodě změnilo, popřípadě jak intenzivní náklon byl. Zda se ale loď nachází ve vodorovné poloze vůči hladině, by bylo z těchto údajů poměrně složité zjistit. Vhodné pro tento případ se tedy jeví spíše využití gravitačního senzoru, který pro každou ze tří os vrací hodnotu náklonu zařízení vůči zemskému povrchu, tedy i vůči vodní hladině.

Kontrolovat správnost zvoleného tempa je již poněkud složitější. Abychom byli schopni určit, zda je tempo správné, potřebujeme ve své podstatě znát čas, jaký by zabralo projetí dané vzdálenosti. Pro každou trať je známý pro danou posádku průměrný čas, jaký by mělo správné projetí dané tratě zabrat. Například pro trať 2km je to pro většinu posádek čas přibližně 9 minut. Znamená to tedy, že pokud posádka tuto trať projede za 10 a více minut, tempo bylo nízké. Naopak i při maximálním úsilí se jen málokdy podaří trať projet rychleji. Problémem tedy bude určení odhadovaného času projetí určené vzdálenosti.

Abychom byli schopni odhadnout, jaký čas zabere projetí známé vzdálenosti, chybí nám již poslední údaj a tím je aktuální rychlost pohybu. Pokud známe vzdálenost i aktuální rychlost, jsme schopni dopočítat i čas. Určit rychlost pohybu by bylo možno například pomocí cox boxu, který byl zmíněný v kapitole dříve. Dále by se dalo uvažovat o využití průtokového čidla. Oboje by ale vyžadovalo rozšíření základního modelu, což by připadalo v úvahu při nemožnosti určení pomocí senzorů mobilního zařízení.

Vzhledem k tomu, že pro sledování polohy a určení ujeté vzdálenosti budeme potřebovat služby GPS senzoru, můžeme ho rovnou využít pro tento účel. Pokud budeme určovat polohu periodicky s dostatečnou přesností a dostatečně vysokou frekvencí, jsme schopni poměrně přesně určit, s jakou rychlostí se zařízení pohybuje. Při známosti aktuální rychlosti již můžeme odhadnout, jaký čas by nám trvalo touto rychlostí projet určenou vzdálenost. Mohlo by se zdát, že pro určení správného tempa tedy stačí znát rychlost. Vzhledem k tomu, že se vzdálenosti, pro které chceme tempo sledovat, budou průběžně měnit, bude pro posádku přehlednější kromě samotného zobrazení rychlost rovnou i přepočítávat na odhadovaný čas.

Ze shrnutí tedy nakonec vyplývá, že by mělo být možné výčet požadavků naplnit a vystačit se samotným mobilním zařízením bez nutnosti rozšíření o externí moduly za využití pouze některých integrovaných senzorů, konkrétně senzoru pro určení polohy – GPS a senzoru pro určení náklonu zařízení vůči zemskému povrchu – gravitačního senzoru.

4. HW principy mobilních senzorů

V následující kapitole budou detailněji představeny senzory, které vyplynuly z předchozí analýzy, jako vhodné pro využití v případové studii.

4.1. GPS

Počátky GPS (Global Positioning System) se datují do 70. let minulého století. Systém začal být vyvíjen pro účely Ministerstva obrany Spojených států. Původním plánem bylo zjistit aktuální polohu kdekoliv na zemském povrchu za pomoci relativně jednoduchého přijímače.

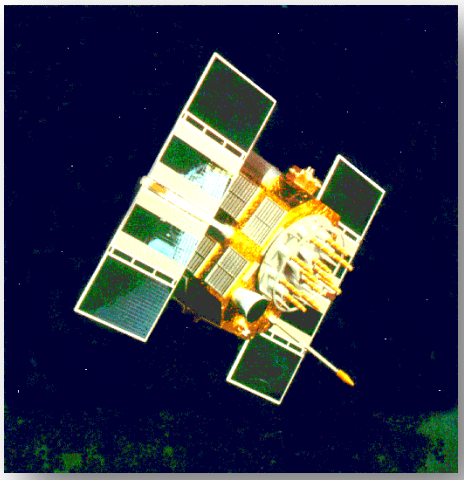
V devadesátých letech došlo k uvolnění systému i pro širokou veřejnost. Signál byl ale uměle zkreslován a vznikala tak záměrně odchylka. Odchylka byla kolem 20 - 30 metrů a navíc signál byl dostupný jen někde. Jednalo se o tzv. selektivní dostupnost. Bylo to opatření hlavně kvůli zneužití teroristy. 1. května 2000 byla tato omezení zrušena. Podobně uvedl Rydval (2005) svůj odborný článek na webu.

Celý systém se dělí na tři hlavní části:

- kosmickou
- uživatelskou
- řídící

4.1.1. Kosmická část

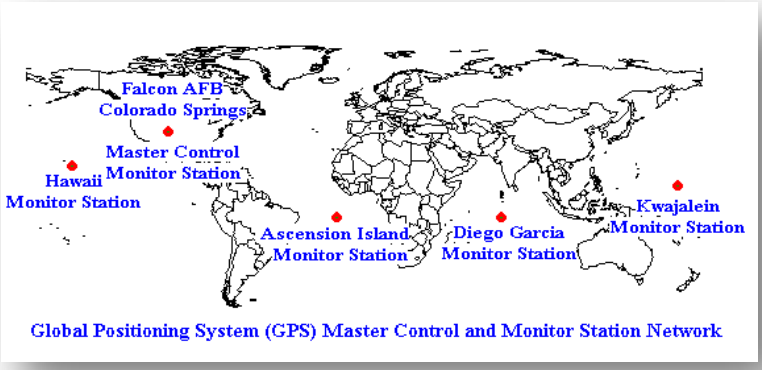
Kosmickou část tvoří 24 satelitů Navstar od firmy Rockwell International. Poslední z 24 satelitů byl vypuštěn na oběžnou dráhu 17. ledna 1994. Tři satelity jsou pouze záložní a jsou schopné okamžitě nahradit jakýkoliv jiný v případě poruchy. Podle Smejkala (2007) satelity obíhají Zemi ve výšce 20200 km nad povrchem a doba oběhu prvá 11hodin a 58 minut. Tyto satelity obíhají Zemi na šesti oběžných drahách skloněných o 60 stupňů. Každý satelit obsahuje přijímač, vysílač, cesiové atomové hodiny s přesností miliardtin sekundy a jiná dalších zařízení, které již pro vlastní určování polohy nejsou potřebné (např. detekce výbuchů jaderných zbraní). Přijímač slouží k předávání dat z řídícího střediska na Zemi do vnitřního počítače satelitu. Na základě těchto dat pak upravuje např. svou dráhu. Vysílač je určen jednak k zasílání dat zpět do řídících center, ale hlavně také k vysílání dat do uživatelské části.



Obrázek 16 - jeden z 24 satelitů obíhajících Zemi (colorado.edu)

4.1.2. Řídící část

Řídící část má za úkol monitorovat běh satelitů, v případě problémů je i řešit a získané údaje předávat satelitům zpět. Řídicí systémy mají na Zemi své stanice. Těchto stanic je celkem devět a jsou rozmístěny podél rovníku. Hlavní řídicí stanice je v Colorado Springs. Dále do této části patří pět monitorovacích stanic a ještě tři pozemní řídící stanice, které spolupracují s hlavní řídící stanicí. O tom ve svém odborném článku pojednávají například (Jurišica, Vitko, Duchoň, & Kaštan, 2011).



Obrázek 17 - rozmístění monitorovacích stanic (beruna.cz)

4.1.3. Uživatelská část

Uživatelskou část tvoří samotný přijímač, který dokáže určit polohu na zemském povrchu, dále rychlost a směr pohybu. Jak uvádí (Rydval, 2005) jde jednak o klasické přijímače s primitivním displejem a také o přijímače zabudované do dalších zařízení,

například PDA, telefony a další. Většina přijímačů je pasivní, tedy pouze přijímají a nevysílají. Přijímač zpracovává najednou signál ze tří a více satelitů. Dokáže určit polohu s přesností kolem 5 metrů. Pokud jsou příznivé podmínky jako dobré počasí a viditelnost nebo otevřený terén, může být odchylka ještě menší.

4.1.4. Systém jako celek

Každý satelit vysílá informace o své poloze, přesný čas z atomových hodin a dále přibližné polohy ostatních družic. Přijímač, který musí mít přímou viditelnost na oblohu, poté pro výpočet polohy využívá časového rozdílu mezi okamžikem vyslání a okamžikem přijmutí dat. Pokud takto získá a zpracuje data alespoň ze tří družic, dokáže určit zeměpisnou šířku a délku - 2D poloha. Pro výpočet nadmořské výšky je pak potřeba signál alespoň ze čtyř satelitů - 3D poloha. Při příjmu z ještě většího počtu satelitů se výpočet ještě více zpřesňuje.

Obrázek 18 - pohled na systém GPS jako na celek (odishasuntimes.com)

Ke zmíněnému výpočtu je ale nutné, aby i v přijímači byl přesný čas, kterého se docílí jednodušším zařízením, než jsou atomové hodiny. Je tomu tak z důvodu rozměrů a také cena. Při načítání informací o satelitech se aktuální čas upraví. Pokud by byl čas rozdílný, třeba jen o tisícinu vteřiny, chyba v určení polohy by byla řádově stovky kilometrů. To vše uvádí (Rydval, 2005) a dále upozorňuje na skutečnost, že po zapnutí přístroje do získání prvních údajů může uběhnout několik desítek vteřin až několik minut. Je to z důvodu, že na začátku lokalizace je nutné načíst informace o jednotlivých družicích a navíc data zvaná almanach. Tento proces se nazývá inicializace. Rychlost načítání načtení je také ovlivněna prostředím, kde se přijímá. Například v uličkách mezi vysokými budovami je situace horší než na vrcholu kopce. Z tohoto důvodu také

doporučuje pro případ, že se plánuje pohyb v místech se zhoršeným signálem, zapnout přijímač a tím načíst almanach v místě, kde je výhled na oblohu ještě dostatečný. Kromě almanachu si musí přístroj načíst ještě informace o sobě - efemeridy, což je ale vzhledem k almanachu již zanedbatelný časový okamžik.

GPS přijímač komunikuje s počítači a jinými zařízeními nejčastěji pomocí protokolu NMEA[11]. V tomto protokolu se mimo jiné předávají informace o čase, poloze, polohách družic, rychlosti, azimutu, počtu aktivních satelitů a další. Formát dat je textový a některé aplikace jej umí ukládat na disk k dalšímu zpracování.

4.2. Gravitační senzor

U tohoto senzoru se jako vhodné na začátek zdá vysvětlení, co vlastně tento pojem ve skutečnosti znamená. Například Android nabízí programové API pro přístup k senzorům mobilního zařízení. Toto rozhraní nám nabízí přístup k senzoru stejně jako k ostatním, které mají v zařízení svůj čip. Gravitační senzor mobilního zařízení však nefiguruje jako samostatná jednotka. Jeho princip spočívá ve spolupráci senzorů jiných, jejichž propojením teprve vzniká jednotka, ze které jsou výstupem požadovaná a relativně přesná data o vyvážení zařízení.

4.2.1. Vysvětlení pojmu

V různých zdrojích je možné se setkat s několika různými pojmy:

- gravitační čip
- gravity sensor
- G-senzor
- akcelerometr

Ve své podstatě všechny tyto pojmy popisují tu samou věc. Z hlediska technologie se zdá nejvhodnějším termínem akcelerometr neboli senzor zrychlení.

4.2.2. Akcelerometr

Dobré úvodní seznámení s akcelerometry podává (Juránek, 2007). Akcelerometr je zařízení určené k měření zrychlení. Je navržen tak, aby při změně z konstantní nebo z nulové rychlosti zaznamenal tuto změnu. Při změně dochází k vibracím spojených s tímto pohybem. Akcelerometr využívá mikroskopické krystaly, na kterých se při působení vibrací generuje napětí odpovídající určitému zrychlení. Jedná se o tzv.

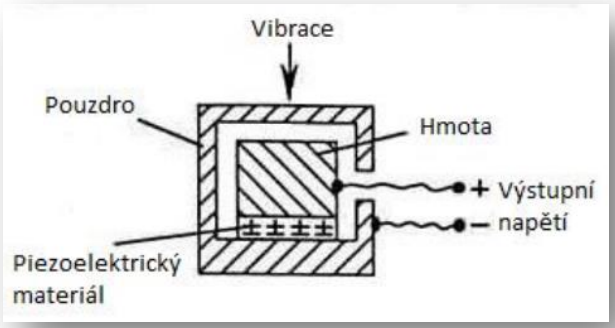
[11] Zkratka National Marine Electronics Association

piezoelektrický jev. Pomocí tohoto jevu lze určit směr gravitace a tedy i natočení přístroje. Této vlastnosti využila poprvé společnost Apple u svých modelů iPhone pro otáčení obrazovky vzhledem k poloze telefonu. Princip později převzali i ostatní výrobci a nyní je již automatické otáčení displeje zcela běžnou záležitostí.

Princip

Jak bylo již zmíněno, akcelerometry měří zrychlení tak, že přeměňují zrychlení (změnu pohybu) na měřitelný elektrický signál. Jakých principů je pro toto využíváno prezentoval například (Vylegala, 2014). Využívá se hlavně následujících tří principů:

- piezoelektrické akcelerometry (PE) - využívají piezoelektrický krystal (přírodní nebo keramiku), který generuje náboj úměrný působící síle, která při zrychlení působí na každý objekt
- piezoresistivní akcelerometry (PR) - využívají mikro křemíkovou mechanickou strukturu, kde zrychlení odpovídá změně odporu
- akcelerometry s proměnnou kapacitou (VC) - využívají mikro křemíkovou mechanickou strukturu, kde zrychlení odpovídá změně kapacity (Vylegala, 2014)



Obrázek 19 - vnitřní struktura piezoakcelerometru (automatizace.hw.cz)

Akcelerometr našel uplatnění v mnoha aplikacích a také v oblasti mobilní zábavy, kde jsou jeho možnosti využity při hraní her. Typickým příkladem jsou mobilní hry, u kterých je zapotřebí pomocí natáčení zařízení udržet kuličky na dráze. U pokročilejších aplikací je možno zařízením otáčet jako s volantem a řídit tím automobil nebo formuli, natočením můžeme řídit náklony motorky nebo zaměřovat zbraní.

4.2.3. Gyroskopický senzor

K akcelerometru bývá v mobilních telefonech často připojován gyroskopický senzor (gyroskop). Ten slouží podobně jako akcelerometr k tomu, aby určoval naklonění a natočení telefonu. Obě zařízení rozebral ve své prezentaci (Vylegala, 2014). Co však tyto dva druhy odlišuje je fakt, že akcelerometr měří zrychlení, zatímco gyroskop úhlovou rychlost. Je proto výhodné použít jejich kombinaci. Akcelerometr určí směr, kterým se mobilní telefon pohybuje pouze ve dvou osách. Rozpoznání pohybu i ve třetí ose zajišťuje gyroskop. Může tak být přesněji určen skutečný pohyb zařízení v prostoru. Spojení gyroskopu s akcelerometrem pochází rovněž z dílny společnosti Apple, která jej poprvé představila ve svém modelu iPhone 4.

Princip

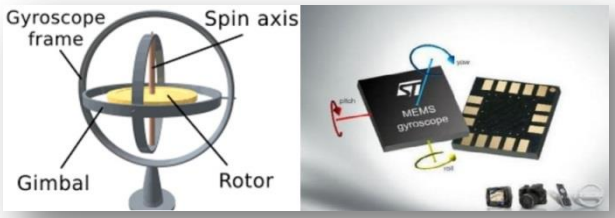
Gyroskopy jsou již dlouhou dobu známy a využívány pro měření a určování změny polohy nebo natočení libovolného předmětu, ke kterému jsou připevněny. Klasická podoba má mechanické provedení. Existuje ale i provedení optické s využitím světla nebo světlovodných vláken. Dnes je již lze najít v integrované podobě klasických součástek obsahující mimo samotný snímač i celou škálu vyhodnocovacích obvodů a logiky. Výstup je pak analogový, digitální nebo obojí. Díky tomu lze gyroskopy použít i v běžných aplikacích, nejen ve vědě a výzkumu. (Vylegala, 2014) prezentuje jak klasickou tak i integrovanou podobu gyroskopu.

Gyroskopy jsou obecně určené pro měření úhlové rychlosti, což je údaj o tom, jak rychle se sledovaný objekt otáčí. Udává se v jednotkách stupňů za sekundu. Rotaci je možné typicky měřit vzhledem k jedné ze tří os x, y a z, někdy označované jako svislá (kolmá) osa (yaw), příčná osa (pitch) a podélná osa (roll). Integrované gyroskopy, vyráběné různými výrobci jako integrované MEMS obvody, pracující na principu Coriolisovy síly[12].

Zde se již jedná o pokročilejší téma, kterému se věnuje například (Vojáček, 2009). Při praktickém použití Coriolisovy síly v integrovaných gyroskopech se využívá technologie MEMS, kde se vytváří na čipu spolu s elektrickými obvody i mechanické mikrosoučásti, které tvoří samotný snímač. Různí výrobci sice používají odlišné

[12] Virtuální síla, která působí na libovolný hmotný předmět či objekt, který se pohybuje rychlostí (v) v soustavě rotující kolem osy rotace úhlovou rychlostí (ω) - FC = 2.m.v X ω, kde X je operátor vektorového součtu

struktury, ale základní princip je vždy podobný. Základem je periodicky se pohybující struktura přesně dané hmotnosti upevněná pomocí pružin v rámu. Směr pohybu musí však vždy být kolmý ke směru otáčení. Za těchto podmínek vzniká a na hmotnou pohybující část snímače působí Coriolisova síla, jejíž velikost je úměrná úhlové rychlosti otáčení. Ta způsobuje stlačení vnějších pružin rámu a způsobí vzájemný posuv měřících plošek fungujících jako elektrody vzduchových kondenzátorů. Výstupem je tedy změna kapacity úměrná úhlové rychlosti otáčení.



Obrázek 20 - porovnání klasické a MEMS podoby gyroskopu (troyhunt.com, techulator.com)

4.2.4. Spolupráce senzorů

Mohlo by se zdát, že je již dále zbytečné zajímat se o data z gyroskopu, když nám akcelerometr poskytuje směrové úhly ve všech 3 osách. Důvodem je, že na akcelerometr se nelze vždy spolehnout. Je tomu tak proto, že akcelerometr měří setrvačnou sílu a tato síla může být vyvolána gravitací a též případným nečekaným pohybem zařízení. Tedy i v případě, že je akcelerometr v relativně stabilní poloze, je velmi citlivý na vibrace a celkově na mechanické šumy. To je hlavní důvod, proč většina systémů pro měření setrvačné síly používá gyroskop pro vyhlazení chyb akcelerometru.

Ani tak toto není zcela dokonalé řešení. Gyroskop není zcela zbaven šumu, ale protože měří rotační pohyb, je méně náchylný k lineárním mechanickým pohybům - šumům, na které trpí akcelerometr. Na druhou stranu, gyroskop má jiné typy chyb, například setrvačnost. To se projevuje tak, že i poté co rotace skončí, trvá nějakou dobu, než se navrátí do nulové polohy. Podstatné však je, že společným zpracováním dat, která jdou současně z akcelerometru a gyroskopu, jsme schopni docílit lepšího odhadu skutečného náklonu zařízení, než jaký bychom získali při využití dat jen ze samotného akcelerometru.

5. API pro práci se senzory

V softwarovém inženýrství se pod zkratkou API[13] chápe rozhraní nad nějakým systémem nebo částí systému, popisující jednak co vše za služby systém nabízí a dále jakým způsobem k těmto službám lze přistupovat. Popisuje také, jaká data můžeme od systému po zavolání služby očekávat a v neposlední řadě v jakém formátu budou data přenášena.

Pro tuto práci bude podstatné rozhraní nad sadou senzorů, integrovaných v mobilním zařízení, na kterém bude v rámci případové studie vyvíjen navrhovaný systém. V následující kapitole bude představeno API, které pro tento účel poskytuje systém Android.

Systém Android umožňuje vývoj pokročilých mobilních aplikací s využitím senzorů, integrovaných v zařízení. Pro tento účel byl vyvinut Android Sensor Framework, který bude detailněji představen v této kapitole. Všechny potřebné informace jsou dostupné přímo v dokumentaci systému (Google, 2016), která bude považována za nejspolehlivější zdroj informací a obrázků pro tuto kapitolu.

5.1. Android Sensor Framework

Většina zařízení na platformě Android má dnes zabudované senzory pro sledování pohybu, orientace a všech možných stavů okolního prostředí. Senzory jsou schopny poskytovat data s vysokou frekvencí i přesností.

Android podporuje tři základní kategorie senzorů:

- pohybové senzory – snímání zrychlení a rotace ve třech osách
- senzory prostředí – měření teploty, tlaku, vlhkosti, viditelnosti
- polohové senzory – určení polohy zařízení

K senzorům lze programově přistupovat pomocí ASF[14]. Ten poskytuje třídy a rozhraní pro široké možnosti využití senzorů v rámci aplikace. Příklady využití:

- určení všech dostupných senzorů
- určení parametrů konkrétního vybraného senzoru

[13] Zkratka pro Application Programming Interface
[14] Zkratka pro Android Sensor Framework

- opakované načítání dat ze senzoru s určenou frekvencí
- (de)aktivovat listenery pro sledování změny stavu senzoru

Nyní budou představeny hlavní objekty, ze kterých se Framework skládá.

5.1.1. SensorManager

Třída poskytuje přístup ke službám senzorů. Nabízí množství metod pro přístup a práci se senzory, aktivaci a deaktivaci listenerů pro zachycení událostí na senzorech a získání dostupných informací. Třída dále obsahuje konstanty, které slouží k určování přesnosti senzorů, frekvence aktualizace dat ze senzorů a kalibraci senzorů.

5.1.2. Sensor

Objekt sloužící k vytváření instancí, které reprezentují samotný senzor. Třída poskytuje různé metody pro využití možností, které senzor nabízí.

5.1.3. Sensor event

Třídu využívá systém k vytváření instancí, které reprezentují jednotlivé události na senzorech. Události na senzorech poskytují tyto informace: samotná data, která senzor zasílá, přesnost se kterou byla data naměřena, referenci na senzor který data zasílá a určení časového okamžiku, ve kterém daná událost nastala.

5.1.4. SensorEventListener

Jedná se o rozhraní, jehož implementace obsahuje dvě metody pro získání informací obsažených v události předané formou parametru metody, které jsou vyvolány v případě naměření hodnoty senzorem, nebo při změně přesnosti senzoru.

Vše výše zmíněné dohromady poskytuje Framework, který pomáhá snadno a přehledně přistupovat k integrovaným senzorům zařízení a v rámci vyvíjené aplikace využívat jejich služby. Dále bude prozkoumáno, jaké služby to typicky jsou.

5.2. Identifikace senzorů a jejich parametrů

Každé zařízení má zabudované jiné senzory. V některých případech může být užitečné mít možnost zobrazit si seznam všech dostupných senzorů a jejich možností, například pokud aplikace obsahuje funkcionalitu postavenou právě na spolupráci se senzory. Toto umožnuje například předem identifikovat, které funkce aplikace nebudou na daném zařízení fungovat z důvodu absence potřebného senzoru a tyto funkce tedy v rámci aplikace deaktivovat.

ASF poskytuje množství metod, které usnadňují za běhu aplikace identifikovat dostupné senzory. Dále poskytuje metody pro určení parametrů těchto senzorů, například rozsah měřených hodnot nebo požadavky na systémové prostředky.

K identifikaci senzorů dostupných v zařízení je nejprve potřeba získat referenci na senzorové služby. K tomu je potřeba nejprve vytvořit instanci objektu SensorManager.

```
private SensorManager mSensorManager;
...
mSensorManager = (SensorManager) getSystemService(Context.SENSOR_SERVICE);

List<Sensor> deviceSensors = mSensorManager.getSensorList(Sensor.TYPE_ALL);
```

Obrázek 21 - ukázka kódu pro výpis všech dostupných senzorů

V předchozí ukázce je uveden příklad, jak vypsat všechny dostupné senzory bez ohledu na typ. To je řečeno konstantou **Sensor.TYPE_ALL** předanou jako parametr metodě getSensorList(). Stejným způsobem je možné vypsat seznam senzorů dle vybraného typu. Toho se docílí pomocí předání příslušné konstanty, například: TYPE_GYROSCOPE, TYPE_LINEAR_ACCELERATION, TYPE_GRAVITY a jiné. Také můžeme rovnou zjistit, zda je vůbec daný typ senzoru k dispozici a to pomocí metody getDefaultSensor(). Pokud zařízení obsahuje více senzorů daného typu, jeden je určen jako defaultní. Pokud zmíněná metoda vrací hodnotu null, znamená to, že zařízení neobsahuje tento typ senzoru. Následuje ukázka zjištění přítomnosti magnetometru v zařízení.

```
private SensorManager mSensorManager;
...
mSensorManager = (SensorManager) getSystemService(Context.SENSOR_SERVICE);
if (mSensorManager.getDefaultSensor(Sensor.TYPE_MAGNETIC_FIELD) != null){
  // Success! There's a magnetometer.
  }
else {
  // Failure! No magnetometer.
  }
```

Obrázek 22 - ukázka kódu pro zjištění přítomnosti magnetometru v zařízení

V tuto chvíli bylo předvedeno, jak snadno v rámci aplikace zjistit, zda má pro své účely k dispozici potřebný HW. Pokud již víme, které senzory máme plně k dispozici, lze přistoupit k hlavnímu účelu senzorů a to je příjem naměřených dat.

5.3. Příjem dat ze senzoru

Pro přijímání dat ze senzoru je potřeba implementovat metody rozhraní **SensorEventListener**, které systém zavolá vždy, když nastane jedna z následujících situací.

5.3.1. Změna přesnosti

Systém vyvolá metodu **onAccuracyChanged()**. Formou parametru poskytne referenci na objekt, kterého se událost týká (**Sensor**) a také poskytne novou hodnotu přesnosti. Přesnost je zde reprezentována jednou ze čtyř konstant:

- SENSOR_STATUS_ACCURACY_LOW
- SENSOR_STATUS_ACCURACY_MEDIUM
- SENSOR_STATUS_ACCURACY_HIGH
- SENSOR_STATUS_UNRELIABLE

5.3.2. Změna stavu

Systém zavolá metodu **onSensorChanged().** Opět vrací referenci na objekt (**SensorEvent**). Ten obsahuje informace o nově pořízených datech včetně přesnosti, se kterou byla data naměřena, referenci na senzor, kterým byla data naměřena a informace o časovém okamžiku, ve kterém byla data naměřena.

Následuje ukázka kódu možného použití metody **onSensorChanged()** pro příjem dat ze senzoru světla:

```
public class SensorActivity extends Activity implements SensorEventListener {
  private SensorManager mSensorManager;
  private Sensor mLight;

  @Override
  public final void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);

    mSensorManager = (SensorManager) getSystemService(Context.SENSOR_SERVICE);
    mLight = mSensorManager.getDefaultSensor(Sensor.TYPE_LIGHT);
  }

  @Override
  public final void onAccuracyChanged(Sensor sensor, int accuracy) {
    // Do something here if sensor accuracy changes.
  }

  @Override
  public final void onSensorChanged(SensorEvent event) {
    // The light sensor returns a single value.
    // Many sensors return 3 values, one for each axis.
    float lux = event.values[0];
    // Do something with this sensor value.
  }

  @Override
  protected void onResume() {
    super.onResume();
    mSensorManager.registerListener(this, mLight, SensorManager.SENSOR_DELAY_NORMAL);
  }

  @Override
  protected void onPause() {
    super.onPause();
    mSensorManager.unregisterListener(this);
  }
}
```

Obrázek 23 - ukázka kódu pro příjem dat ze senzoru světla

5.3.3. Parametrizace příjmu dat

V předchozím příkladu je určena výchozí frekvence snímání dat pomocí konstanty **SENSOR_DELAY_NORMAL** v rámci metody registerListener(). Frekvence snímání určuje interval, ve kterém senzor zasílá aplikaci naměřená data pomocí metody onSensorChanged(). Výchozí hodnota tohoto intervalu je 200 000 mikrosekund. Takto se dá frekvence upravovat dle potřeby a požadavků kladených na aplikaci.

Následuje přehled možných nastavení frekvence aktualizace dat ze senzoru:

- **SENSOR_DELAY_NORMAL** – 200 000 ms
- **SENSOR_DELAY_UI** – 60 000 ms
- **SENSOR_DELAY_GAME** – 20 000 ms
- **SENSOR_DELAY_FASTEST** – 0 ms

Od Androidu verze 3.0 je možné také tuto hodnotu zadat pomocí absolutní hodnoty.

Takto určená hodnota frekvence je pouze navržená hodnota. Systém Android a okolní aplikace mohou tuto hodnotu měnit. Správně by měla být zadána nejvyššího přípustného zpoždění, protože systém většinou používá nižší hodnot, než která byla zadána. Měla by tedy být zadána hodnota co největšího možného zpoždění, při které jsou stále splněny požadavky. Pokud je toto vše dodrženo, pak si aplikace klade nižší nároky na procesor zařízení a tím pádem není aplikace tolik náchylná na rychlé vybíjení baterie.

Neexistuje žádná veřejná metoda pro určení, v jaké frekvenci senzor data zasílá do aplikace. Pro tento účel je možné využít informace o časovém okamžiku pořízení dat. Z několika po sobě následujících aktualizací dat je možné určit, s jakou frekvencí aplikace data přijímá. Jakmile je jednou nastavena hodnota frekvence přijímání dat, neměla by již být měněna. Pokud taková potřeba z nějakého důvodu nastane, pak je potřeba deaktivovat a znovu aktivovat listener na změnu hodnoty senzoru.

Důležité je také upozornit na použité motody v předchozím příkladu - **onResume() a onPause().** Vždy by se měli deaktivovat listenery o kterých víme, že již dále nebudou potřeba, například v případě pozastavení aktivity. Pokud se toto zanedbá, hrozí úplné vybití baterie během krátké doby, protože některé senzory mají značné nároky na výkon a při jejich používání se baterie vybíjí rychleji. Systém totiž senzory nedeaktivuje automaticky při vypnutí obrazovky.

5.4. Konfigurace senzorů

Android nespecifikuje standardní konfiguraci senzorů v závislosti na zařízení. Toto znamená, že zařízení mohou obsahovat množství senzorů se širokou škálou možností konfigurace. Dále také to, že pokud je senzor integrován v jednom zařízení, neznamená to, že bude obsažen i ve všech ostatních. Pokud je aplikace závislá na některém druhu senzoru, pak bychom se vždy měli přesvědčit, zda dané zařízení požadovaný senzor obsahuje. Nabízí se možnosti, jak toto zkontrolovat.

5.4.1. Programové určení dostupných senzorů

Pokud aplikace využívá některý typ senzoru, na kterém však není závislý chod celé aplikace, je možné využít Sensor Framework pro detekci senzoru až za běhu aplikace a poté případně zapnout či vypnout některé systémové funkce dle potřeby. Například aplikace může využívat senzory polohy a teploty. V případě, že je aplikace spuštěna na zařízení, které nemá integrován snímač tlaku, je možné pomocí frameworku absenci tohoto senzoru předem zjistit a poté vypnout některé funkce aplikace, které jsou na tomto senzoru postavené. Následuje ukázka kódu pro zjištění, zda je v zařízení dostupný senzor tlaku:

```
private SensorManager mSensorManager;
...
mSensorManager = (SensorManager) getSystemService(Context.SENSOR_SERVICE);
if (mSensorManager.getDefaultSensor(Sensor.TYPE_PRESSURE) != null){
// Success! There's a pressure sensor.
}
else {
// Failure! No pressure sensor.
}
```

Obrázek 24 - ukázka kódu pro detekci tlakového senzoru

5.4.2. Specifikace v rámci manifestu projektu

V manifestu aplikace by měl být využit element <uses-feature> (obzvláště pokud se aplikace po dokončení pro veřejnost publikuje na Google Play) pro určení zařízení, které svou konfigurací odpovídají naší aplikaci. Element poskytuje možnosti hardwarového popisu přítomnosti různých senzorů.

```
<uses-feature android:name="android.hardware.sensor.accelerometer"
              android:required="true" />
```

Obrázek 25 - ukázka hardwarového popisu využitého senzoru

Měl by se používat atribut **android:required="true"**, ale pouze v případě, že naše aplikace na daném senzoru závisí. Pokud aplikace pouze využívá pro některé funkce, ale běží i bez přítomnosti senzoru, pak by tento atribut měl být nastaven na

android:required="false". Toto by mělo případně zabránit zařízení v instalaci aplikace, která závisí na senzoru, který v zařízení není dostupný. Pro uživatele, kteří si na Google play filtrují aplikace dle konfigurace, by se aplikace, jejíž požadavky neodpovídají požadované hardwarové konfiguraci, neměly ani zobrazit.

5.5. Doporučení pro programování senzorů

5.5.1. Deaktivace nevyužitých listenerů

Vždy by se mělo dbát na to, aby byla zajištěna deaktivace všech listenerů pro senzory, se kterými již byla práce dokončena, nebo pokud je aktivita pozastavena. Pokud je listener aktivován a aktivita je pozastavena, senzor nadále pokračuje ve snímání a poskytování dat a tím pádem i nadále využívá systémové prostředky, dokud není listener deaktivován. Následuje ukázka kódu, ve kterém je toto ošetřeno.

```
private SensorManager mSensorManager;
  ...
@Override
protected void onPause() {
  super.onPause();
  mSensorManager.unregisterListener(this);
}
```

Obrázek 26 - ukázka kódu pro deaktivaci listeneru při pozastavení aktivity

5.5.2. Testování aplikace

Pro testování aplikace založené na využití senzorů integrovaných v zařízení nestačí pouze emulátor, jako pro testování chodu jednoduché aplikace. Emulátor není schopen simulovat senzory. Pro správné otestování takové aplikace bude potřeba propojení s klasickým mobilním zařízením. Na něj poté touto cestou vyvíjenou aplikaci nainstalovat a samotné testování dále probíhá již fyzicky na samotném zařízení.

5.5.3. Použití deprecated metod a typů senzorů

Některé metody a konstanty, které Android pro programování nabízí, jsou již deprecated[15]. Pro vyšší verze Androidu by tak mohl později nastat problém s kompatibilitou. Například konstanta TYPE_ORIENTATION pro senzor orientace je již zastaralá a pro zjištění orientace by měla být dále využívána spíše metoda getOrientation(). Dále například TYPE_TEMPERATURE by měla pro zařízení s Android verzí 4.0 a výše nahradit TYPE_AMBIENT_TEMPERATURE.

5.5.4. Ověření přítomnosti senzorů

Vždy by mělo proběhnout ověření, zda daný senzor v zařízení opravdu existuje předtím, než nastanou pokusy získávat z něj naměřená data. Nelze na přítomnost senzoru spoléhat pouze z důvodu, že se jedná o často využívaný senzor. Výrobci zařízení totiž nejsou nijak vázání k tomu, kterými senzory své zařízení opatří. „*Android does not require device manufacturers to build any particular types of sensors into their Android-powered devices, so devices can have a wide range of sensor configurations.*“ (Google, 2016)

5.5.5. Využití zpoždění senzorů

V okamžiku, kdy je aktivován listener metodou registerListener(), mělo by se obezřetně zvážit, jaká frekvence aktualizace dat bude od senzoru očekávána a která bude dostatečná pro splnění požadavků v návrhu aplikace. Senzory mohou poskytovat data ve velmi vysoké frekvenci. Povolením systému získávat nadbytečné množství dat, která nejsou potřeba, je způsobeno vysoké zatížení HW prostředků zařízení a tím i vysoký úbytek energie.

[15] Termín softwarového inženýrství pro zastaralé komponenty, které byly nahrazeny novou komponentou a již nemusí být podporovány.

6. Analýza a návrh vzorové aplikace

Cílem bude navrhnout, vyrobit a následně uvést do provozu systém, který sice nebude nabízet širokou škálu funkcí a možností rozšíření a personalizace, jako jiné známé fitness asistenty, zato však naplní požadavky plynoucí ze zásad vybrané sportovní aktivity, o kterých bylo pojednáno v příslušné kapitole. K dispozici pro řešení bude mobilní zařízení na bázi OS Android s integrovanými senzory a za pomoci této platformy bude cílem vyrobit systém s očekávanými vlastnostmi za pomoci dostupných odborných zdrojů.

Systém bude mít podobu mobilní aplikace pro sledování a vyhodnocení tréninku na dračí lodi, který nebude nabízet pouze možnost vyčíslit dobu trvání tréninku, počet ujetých kilometrů a maximální rychlost, kterou se loď pohybovala, jako tomu bylo dosud za použití samostatného externího GPS senzoru, ale bude navíc jako přidanou hodnotu umět rozeznat a vyhodnotit, zda loď po dobu tréninku plula dle zásad ve správném náklonu, což je pro posádku velmi podstatný údaj.

Pokud má být takový tréninkový asistent uveden do provozu, pak bude rozhodně dále potřeba i dostatečné otestování v reálném provozu a následně vyhodnocení, zda je systém vůbec použitelný, případně vhodně citlivý, zda má očekávaný přínos a vhodné by bylo i vyhodnocení a upozornění na vypozorované nedostatky, v lepším případě i návrh možností vylepšení a očekáváné překážky při řešení těchto nedostatků.

6.1. Očekávané řešené problémy a postup práce:

1. Návrh jednoduchého řešení s minimalizací vstupních nákladů
2. Zprovoznění vývojového prostředí
3. Analýza a návrh aplikace
4. Využití Android API pro práci se senzory
5. Tvorba a dokumentace aplikace
6. Řešení problému s uspáváním procesů
7. Převod jednoduchých dat na požadované statistiky
8. Testování aplikace

6.2. Příprava před započetím vývoje

6.2.1. HW strana

Pro HW řešení bude využito mobilní zařízení nižší řady, zato však dnes stále ještě běžně dostupné a poskytující požadované vlastnosti a má integrovány potřebné senzory. Tím bude HW strana vyřešena.

- Zařízení: Samsung Galaxy S3 Neo
- Systém urč. polohy: A-GPS, GLONASS[16]
- Integrované senzory: Hallův snímač, snímač zrychlení, geomagnetický snímač, gyroskop, světelný snímač, snímač vzdálenosti, snímač RGB
- Průměrná cena: 5000Kč

(SAMSUNG, 2015)

6.2.2. SW strana

SW strana bude vyřešena vývojem Android aplikace, z čehož nevyplývají žádné extra náklady při předpokladu, že máme k dispozici základní, i v domácnostech dnes běžně používanou IT výbavu, kterou případně doplníme potřebným SW vybavením. Pro vývoj bude použito PC, na kterém bude zprovozněno vývojové prostředí a dále pomocí USB kabelu připojené zařízení pro testování vyvíjené aplikace.

6.2.2.1. Zprovoznění lokálního vývojového prostředí:

- instalace JDK
- instalace Android SDK pomoci SDK Manageru
- instalace IDE (např. základní Eclipse IDE)
- instalace pluginu v rámci Eclipse IDE
- instalace USB driveru pro připojení mobilního zařízení
- povolení vývojářského režimu v mobilním zařízení

V tuto chvíli by již mělo být možné založit nový Android projekt a spustit ho přímo na mobilním zařízení, připojeném přes USB rozhraní.

[16] Ruská alternativa amerického systému GPS

6.2.2.2. Problematika uspávání procesů a řízení spotřeby

Problematika uspávání a probouzení procesů v rámci úspory energie je velmi podstatná i pro práci se senzory. Pokud je systém nečinný, pak nastává problém, že systém Android práci senzorů potlačí, což pro náš účel dlouhodobého snímání není žádoucí. Obecně tento problém vysvětluje samotný Google (Google, 2016). Dále je také tato problematika probírána specificky v různých odborných pracích. Například (PARK, Dongwon, & Hojung, 2015) ve své práci navrhují schéma pro snížení spotřeby energie, díky identifikaci a zamezení častých nepotřebných probouzení systému.

Dále (Jindal, Pathak, Hu, & Midkiff, 2013) ve své práci tvrdí, že OS chytrých zařízení implementují až agresivní politiku uspávání systému, což značně znesnadňuje práci vývojářům, kteří toto vše musí brát v potaz a snadno se dopouštějí programových chyb, které vedou k náhlému a nečekanému vybití akumulátoru při používání aplikace.

Například (Chang, 2011) se věnoval technikám konfigurace a kalibrace mobilních senzorů za účelem zvýšení jejich výkonnosti při zachování nízké spotřeby energie.

6.3. Non-funkční analýza

Cílem bude se držet základního modelu, který byl vysvětlen v kapitole Základní model využití integrovaných senzorů a udržet tak zařízení soběstačné i bez napojení na externí systémy. Proto i ukládání výsledků bude řešeno offline formou a to ukládáním v podobě souboru přímo do zařízení. Aplikace bude ukládat pouze jednoduchý výpis výsledků měření. Ty nemusí být nijak provázané, nebude potřeba nad nimi nijak filtrovat ani vyhledávat. Z toho důvodu se použití SQL lite databáze, se kterou Android pracuje, jeví jako příliš složité řešení.

Co se týče verze Androidu, s vývojem se dostupnost senzorů měnila z verze na verzi. Dle (Google, 2016) jsou typy senzorů, které byly představeny v některé z nižších verzí, ale již nebyly implementovány ve vyšších verzích. Některé typy byly zavedeny ve verzi Android 2.3 (API Level 9) a Android 4.0 (API Level 14). Některé zase byly zastaralé a byly nahrazeny novějšími. Aplikace tedy nemusí být kompatibilní s verzemi staršími, než byly zmíněny. I vzhledem k tomu, že většina dnes běžně používaných Android zařízení používá verze minimálně API Level 14, měla by být aplikace vyvíjena tak, aby její kompatibilita necílila pouze na nejnovější verze, ale spadala i do starších verzí.

Předpokladem je, že zařízení musí být opatřeno potřebnými senzory. Co se týče frekvence snímání dat ze senzoru, budeme potřebovat velmi krátkou prodlevu mezi aktualizacemi dat a to u obou použitých senzorů. Z pohledu snímání polohy pomocí GPS je, že z těchto údajů budeme chtít dopočítávat další statistiky. Čím delší prodleva mezi jednotlivými okamžiky určení polohy, tím větší nepřesnost bude vznikat mezi skutečně uraženou vzdáleností a dopočítanou vzdáleností z dvou posledních bodů polohy. Při tomto přepočtu se uvažuje, že zařízení se pohybovalo po přímce mezi danými body. Pokud bude interval snímání příliš dlouhý, pak je velká pravděpodobnost, že se zařízení v průběhu intervalu vícekrát vychýlilo ze směru a celková uražená vzdálenost je o to větší. Tato nepřesnost by se pak samozřejmě dále zanášela do dopočítaných údajů. Pokud by tedy naměřená vzdálenost byla výrazně kratší, než ta skutečná, projevilo by se to i na vypočtené rychlosti, u které by vycházely také nižší hodnoty, než je rychlost skutečná.

Z pohledu gravitačního senzoru je to z důvodu impulzivních náklonů lodě. Na dračí lodi se nejedná a táhlé náklony trvající několik vteřin, známé z nákladních lodí. Zde jsou výkyvy nečekané a prudké, například při nevhodném pohybu někoho z posádky. Takový výkyv je poté možné zachytit pouze v krátkém okamžiku, protože dojde-li v posádce k pohybu, který výkyv vyvolá, loď se vykloní a během 1-2 vteřin se může opět vrátit do vyvážené polohy. Pokud by tedy interval snímání byl 2 sekundy, nemusel by být tento náklon zachycen. Z výše uvedených důvodů je tedy požadováno, aby aplikace aktualizovala data ze senzorů s nejvyšší možnou frekvencí.

Další možností pro zvýšení přesnosti z pohledu snímání polohy, kromě vysoké frekvence aktualizace polohy, je využití asistované GPS, která využívá mobilní data pro zpřesnění určení polohy. Z toho tedy vyplývá další požadavek a to ten, že zařízení provozující aplikaci by mělo mít zpřístupněno využití mobilních dat od operátora. Pokud bude toto splněno, poté by aplikace měla být schopna díky A-GPS poměrně rychle aktualizovat polohu zařízení a to relativně přesně dle tvrzení v kapitole pojednávající o HW principech GPS.

Během sledování aktivity je očekáváno, že zařízení nebude vyžadovat interakci s uživatelem. To znamená, že poté co je zařízení umístěno v lodi na vhodném místě pro sledování a je zapnut sledovací režim, měla by mít posádka aplikaci na očích, aniž by se v průběhu objevovala různá dialogová okna a stejně tak není žádoucí, aby se zařízení

klasicky uspávalo po určitém intervalu uživatelské nečinnosti. Vzhledem k tomu, že aplikace by během sledování neměla upadat do úsporného režimu a navíc je požadována co nejvyšší frekvence aktualizace dat ze senzorů, bude potřeba počítat s vysokými nároky na systémové prostředky a mělo by tedy být zajištěno, aby bylo zařízení vždy opatřeno kvalitní baterií (popřípadě připojením na powerbank[17]) s kapacitou dostatečnou pro udržení zařízení v provozu po celou dobu tréninku.

Z výše uvedené analýzy tedy vyplývají následující požadavky:

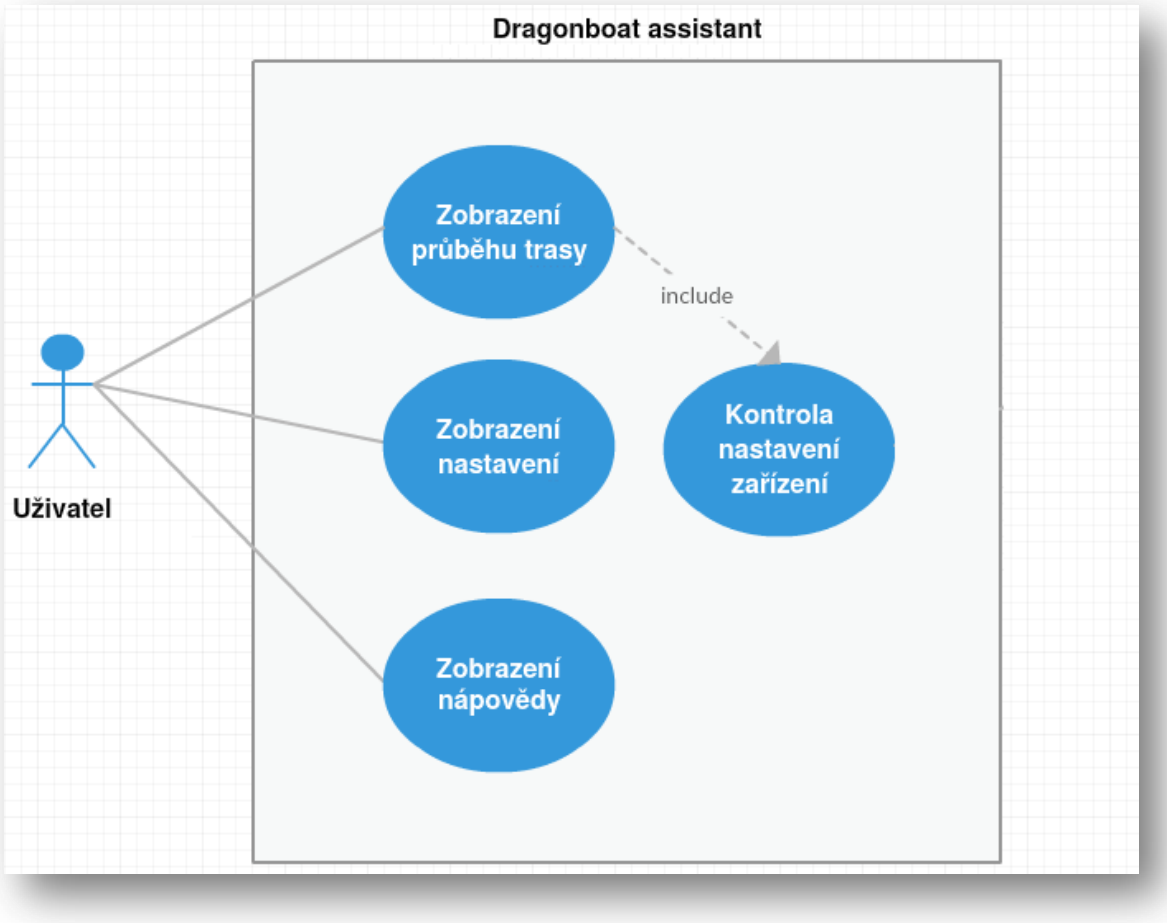
- data aplikace se budou ukládat do zařízení ve formě souboru
- aplikace bude kompatibilní s verzemi Android již od API level 14
- aplikace bude navržena bez ohledu na otočení displeje
- zařízení bude opatřeno potřebnými senzory
- je očekávána co nejvyšší frekvence aktualizace dat ze senzorů
- bude zajištěn přístup aplikace k internetu
- aplikace nebude během sledování upadat do režimu spánku
- zařízení bude opatřeno dostatečným zdrojem energie
- zařízení bude opatřeno ochranou proti vniknutí vody
- aplikace nebude obsahovat uživatelské role a přihlášení

[17] Přenosné zařízení pro dobíjení kompatibilního mobilního zařízení bez přístupu k elektrické síti

6.4. Funkční analýza

V této části bude následovat analýza funkčních požadavků a popis jednotlivých částí.

Hlavní menu



Obrázek 27 - funkcionalita v rámci hlavního menu aplikace

Po spuštění aplikace bude mít uživatel možnost zobrazit průběh trasy, zobrazit nastavení a zobrazit nápovědu. Každá z těchto akcí povede k přechodu na novou obrazovku. Zobrazení průběhu trasy bude znamenat přechod na obrazovku, kde již bude klasický panel, známý z jiných sportovních aplikací, tedy zobrazení polohy zařízení, dále komponenty pro zobrazení statistik aktuální sledované aktivity a samozřejmě i ovládací panel pro řízení sledování. Po přechodu na tuto obrazovku si již aplikace začne načítat data o poloze, tedy by mělo být zajištěno, že zařízení je nastaveno tak, aby aplikace měla všechny potřebné prostředky k dispozici. Proto v tuto chvíli systém provede kontrolu a v případě, že nastavení zařízení neodpovídá potřebám (například jsou v zařízení vypnuty služby pro zjišťování polohy), nabídne systém uživateli dialog

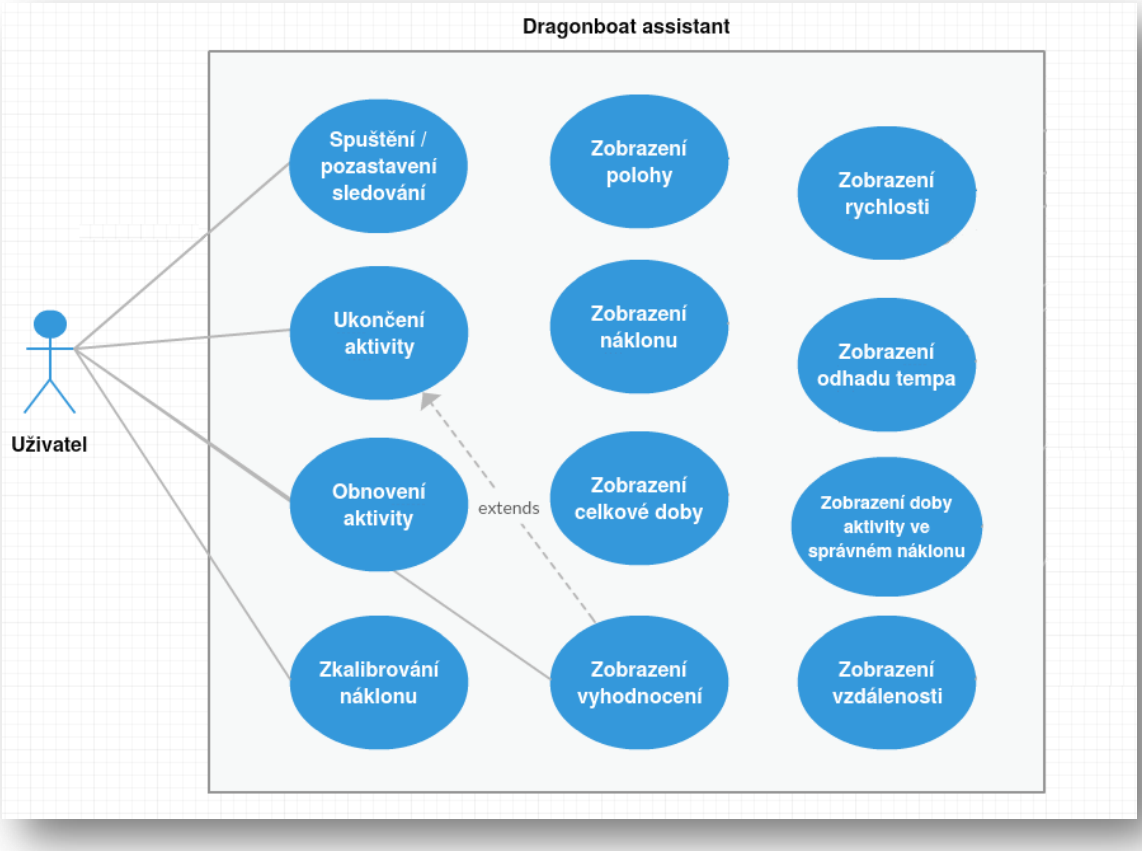
s popisem, které nastavení je pro správnou funkci potřeba změnit a spolu s tím nabídne uživateli možnost potvrdit automatickou změnu nastavení systémem.

Po zobrazení nastavení systém přejde na novou obrazovku a zobrazí uživateli prvky pro změnu nastavení aplikace. Vzhledem k tomu, že aplikace nebude obsahovat uživatele a tím pádem ani uživatelské role, bude uživateli umožněn přístup kamkoliv včetně úpravy všech možností nastavení.

Možnosti nastavení:

- GPS – nejkratší vzdálenost, po které se aktualizuje pozice
- GPS – nejkratší časový úsek, po kterém se aktualizuje pozice
- GPS – priorita přesnosti polohy
- Gravitační senzor – citlivost na náklon
- Odhad tempa – vzdálenost, pro kterou se bude odhadovat čas dle rychlosti

Sledování aktivity

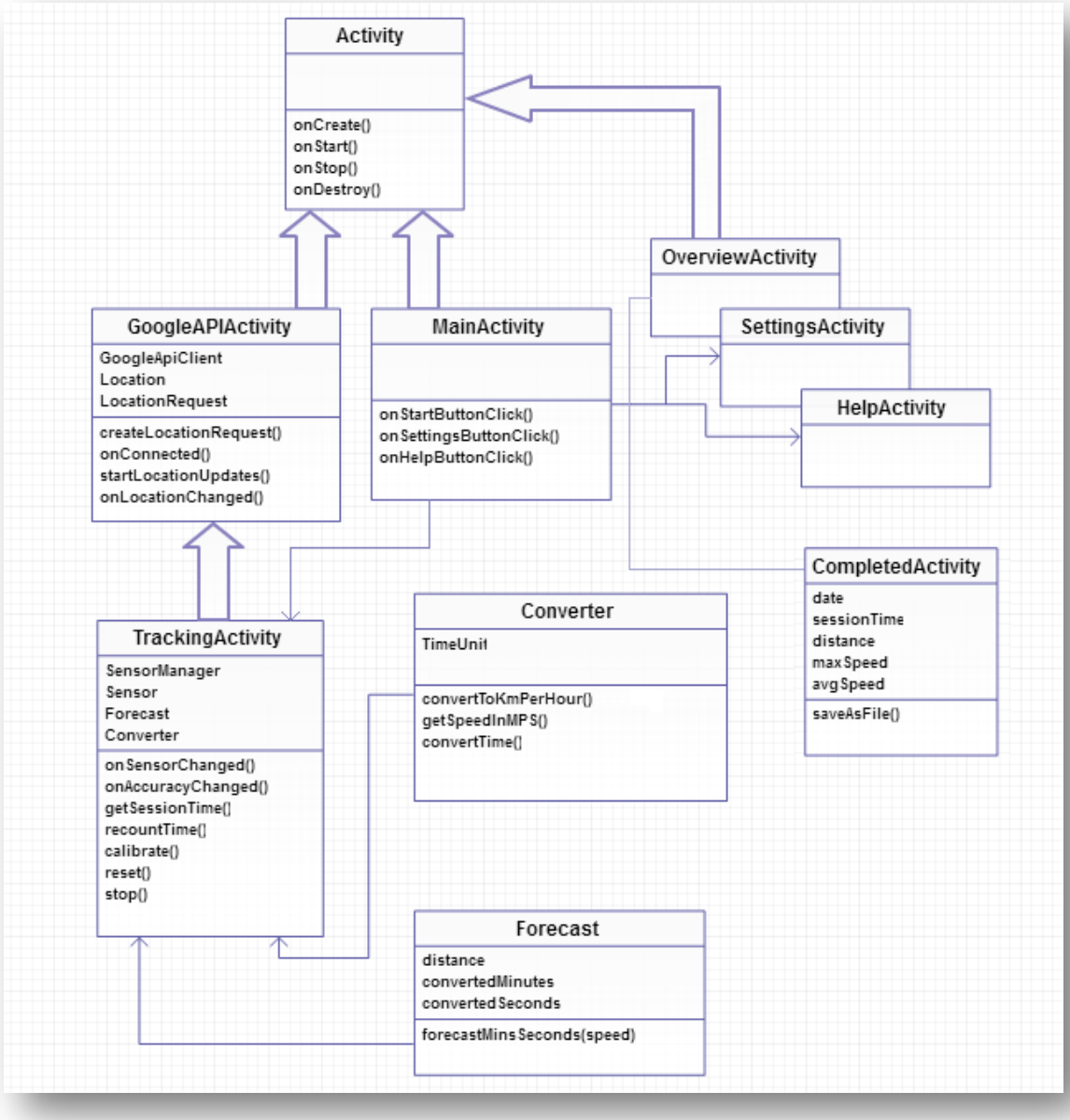


Obrázek 28 - funkcionalita v rámci sledování aktivity

Na obrazovce pro sledování aktivity bude mít přehled o všech dostupných statistikách, nebude potřeba nijak přecházet mezi obrazovkami v rámci jednoho sledování. Uživatel bude mít možnost sledování kdykoliv spustit a v průběhu sledování libovolně pozastavovat. Pokud bude potřeba začít sledování od začátku, bude mít uživatel možnost statistiky vynulovat a dostat tak do stavu jako po zapnutí aplikace. Pokud se nepodaří zařízení v lodi umístit ve vodorovné pozici a aplikace tak bude stále hlásit špatný náklon, bude možnost toto v aplikaci zkalibrovat, tedy v jakémkoliv okamžiku určit, v jaké poloze má být zařízení uvažováno, jako ve vodorovné poloze. Na závěr sledování bude možnost aktivitu ukončit, tedy nebude již možné ve sledování dále pokračovat. Teprve tehdy nastane případ, kdy bude uživatel odkázán na další obrazovku. Pokud uživatel potvrdí, že chce aktivitu vyhodnotit, systém přejde na další obrazovku s výpisem výsledných statistik aktivity. Zde bude mít uživatel možnost po prohlédnutí výsledky uložit do zařízení.

6.5. Aplikační logika

V této kapitole bude představena analýza aplikační logiky, tedy návrh použitých aktivit a tříd poskytujících podpůrné služby. Dále bude následovat popis jednotlivých částí.



Obrázek 29 - aplikační logika se zachycením nejdůležitějších tříd a jejich metod

Activity

Původní Android aktivita představující obrazovku aplikace a implementující známé metody, volané při příchodu nebo odchodu z obrazovky aplikace, nebo při uzamknutí obrazovky mobilního zařízení. Z ní dědí další třídy, které mají mít charakter aktivity a budou popsány nyní.

MainActivity

Aktivita představující hlavní menu aplikace. Obsluhuje tlačítka pro přechod na další obrazovky a tedy spouští ostatní aktivity dle výběru – TrackingActivity, SettingsActivity, HelpActivity

GoogleAPIActivity

Abstraktní aktivita sloužící jako API pro přístup ke službám Google ze které budou dědit aktivity, ve kterých se bude pracovat s informacemi o poloze zařízení.

TrackingActivity

Aktivita, na které je postavena hlavní část aplikace a to sledování průběhu aktivity. Zde je pomocí ASF implementován přístup k senzorům zařízení. Dále aktivita dědí z GoogleAPIActivity přístup ke službám Google pro zpracování údajů o poloze. Aktivita využívá některé pomocné třídy.

- **Converter** – třída poskytující služby pro převody dat, například převod jednotek rychlosti, dále převod jednotek času, ale také převod dat o poloze zařízení na aktuální rychlost pohybu
- **Forecast** – třída podporující proces kontroly správnosti tempa sloužící pro přepočet aktuální rychlosti na odhadované časy dle zadané trati

HistoricalActivity

Třída reprezentující ukončenou sledovanou aktivitu pro uložení do souboru a obsahuje konečné statistiky týkající se aktivity.

Dále některé jednoduché pomocné aktivity:

- OverviewActivity – slouží k zobrazení vyhodnocení a konečných statistik ukončené aktivity
- SettingsActivity – slouží k parametrizaci aplikace (viz **Možnosti nastavení**)
- HelpActivity – slouží k zobrazení nápovědy

7. Praktické řešení a tvorba aplikace

V této kapitole bude představena ukázková aplikace. Konkrétně půjde o prototyp aplikace, který neobsahuje všechny funkce, které vzešly z analýzy.

7.1. Prototyp ukázkové aplikace

Cílem prototypu bylo ověřit řešení navržená v předchozích částech práce a uvést do provozu funkcionalitu důležitou pro splnění svého účelu v rámci vybrané sportovní aktivity. Prototyp, který bude v této kapitole představen, je již použitelný při tréninku na dračí lodi a i přes absenci některých navržených funkcí již nyní plní svůj účel v reálném provozu.

7.1.1. Vybrané funkce prototypu

Smyslem bylo v první řadě ukázat práci se senzory a využití jednoduchých dat ze senzorů ke smysluplnému vyhodnocení celé aktivity. Některé funkce se však ukázaly jako ne příliš podstatné pro tento účel a mohou být naplánovány pro pozdější implementaci ve formě možného vylepšení aplikace. Z toho důvodu se pro prototyp aplikace upustilo od implementace následující funkcionality.

Nastavení

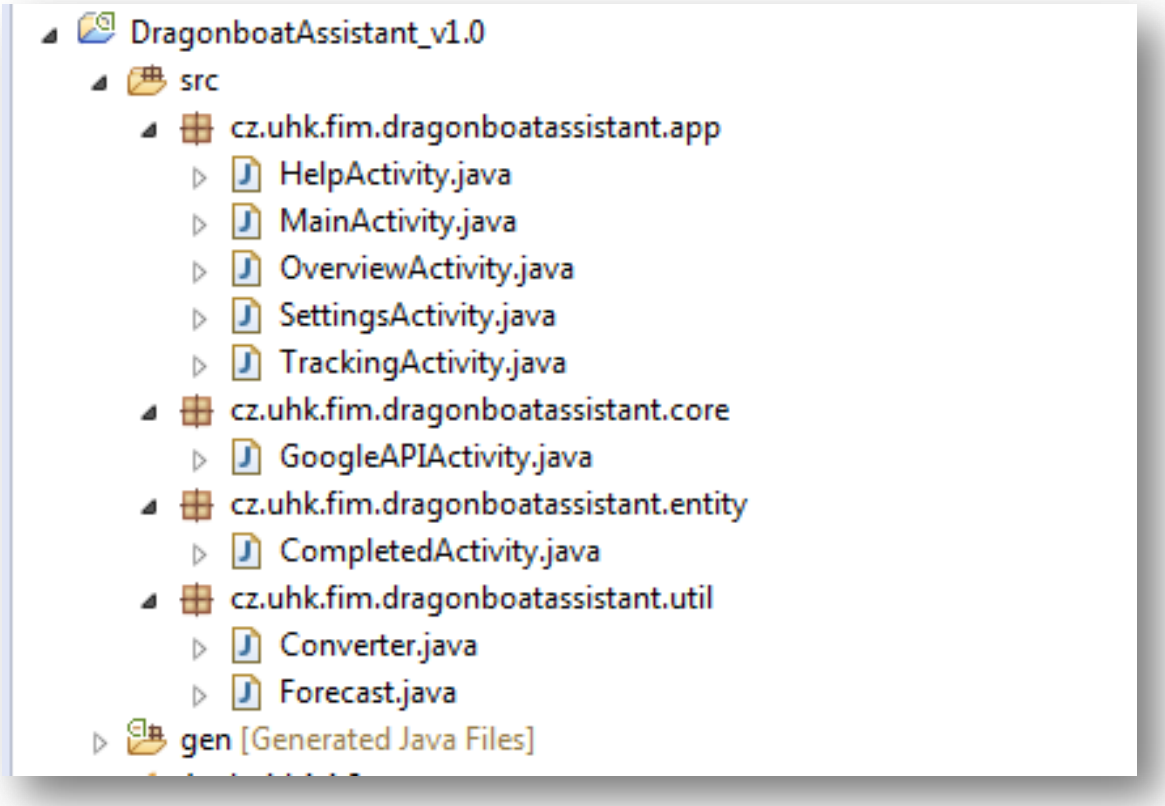
Pro prototyp aplikace bylo použito výchozí programové nastavení, které se ukázalo jako využitelné pro získání smysluplných dat. V rámci prototypu toto nastavení měnit nelze, ale v rámci postupného vylepšování aplikace bude funkce implementována pro možnost uživatele snížit nároky aplikace na systémové prostředky, v případě že i při snížení těchto nároků bude aplikace vracet dobře využitelné údaje.

Ukládání do souboru

Po ukončení aktivity je zobrazeno vyhodnocení, které se po zhlédnutí a uzavření uživatelem ztratí. Implementace ukládání výsledků do souboru již není pro tuto případovou studii podstatná, proto v rámci prototypu nebylo implementováno.

7.2. Struktura aplikace

7.2.1. Balíčky



Obrázek 30 - struktura projektu aplikace

Package app

Aktivity pro konkrétní účel, u kterých se nepředpokládá znovupoužitelnost.

Package core

Abstraktní aktivity řešící nějaký obecný problém, u kterého se předpokládá, že bude potřeba později řešit i na dalším místě. Nové třídy pak mohou dědit z těchto abstraktních. Předpokládá se u nich tedy znovupoužitelnost.

Package entity

Třídy představující celek, nesoucí množinu dat, které spolu v určitém kontextu souvisí.

Package util

Pomocné třídy, které nabízí množinu metod pro řešení obecných problému, například převody jednotek.

7.2.2. AndroidManifest

Je základem Android aplikace a shrnuje všechny použité aktivity. Zde je určeno, která aktivita je hlavní. Dále je zde uvedena verze aplikace a verze Androidu, pro které je aplikace vyvíjena. Také je zde určeno, jaká oprávnění aplikace v zařízení bude vyžadovat.

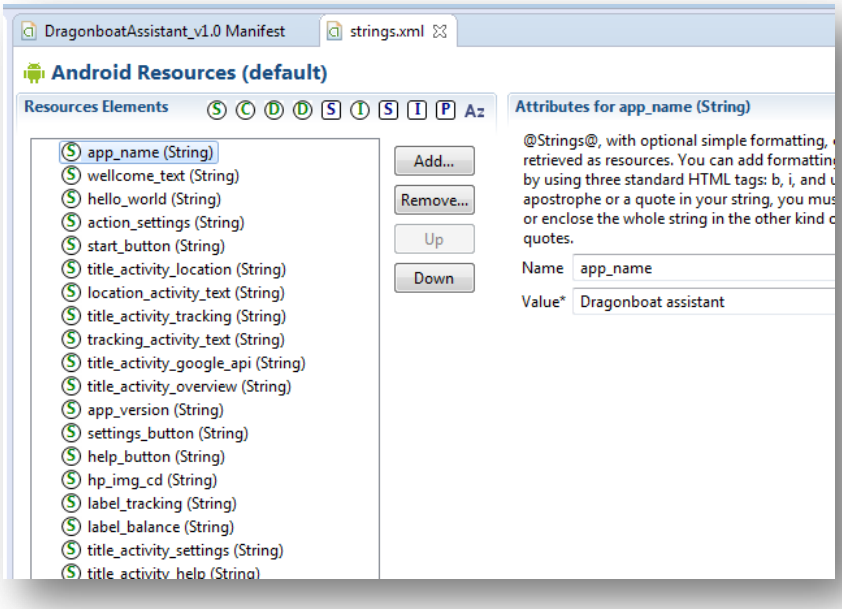
DragonboatAssistant_v1.0 Manifest

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3      package="cz.uhk.fim.dragonboatassistant"
4      android:versionCode="1"
5      android:versionName="1.0" >
6
7      <uses-sdk
8          android:minSdkVersion="14"
9          android:targetSdkVersion="21" />
10
11     <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
12     <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
13     <uses-permission android:name="android.permission.INTERNET" />
14
15     <application
16         android:allowBackup="true"
17         android:icon="@drawable/ic_launcher"
18         android:label="@string/app_name"
19         android:theme="@style/AppTheme" >
20         <activity
21             android:name="cz.uhk.fim.dragonboatassistant.app.MainActivity"
22             android:label="@string/app_name" >
23             <intent-filter>
24                 <action android:name="android.intent.action.MAIN" />
25                 <category android:name="android.intent.category.LAUNCHER" />
26             </intent-filter>
27         </activity>
28
29         <meta-data
30             android:name="com.google.android.gms.version"
31             android:value="@integer/google_play_services_version" />
32         <meta-data
33             android:name="com.google.android.geo.API_KEY"
34             android:value="AIzaSyB8phN0A0x4fa_oNfIshxvwJ0CDaNSXe74" />
35
36         <activity
37             android:name=".TrackingActivity"
38             android:label="@string/title_activity_tracking" >
39         </activity>
40         ...
41     </application>
42
43 </manifest>
44
```

Obrázek 31 – soubor AndroidManifest.xml

7.2.3. Strings

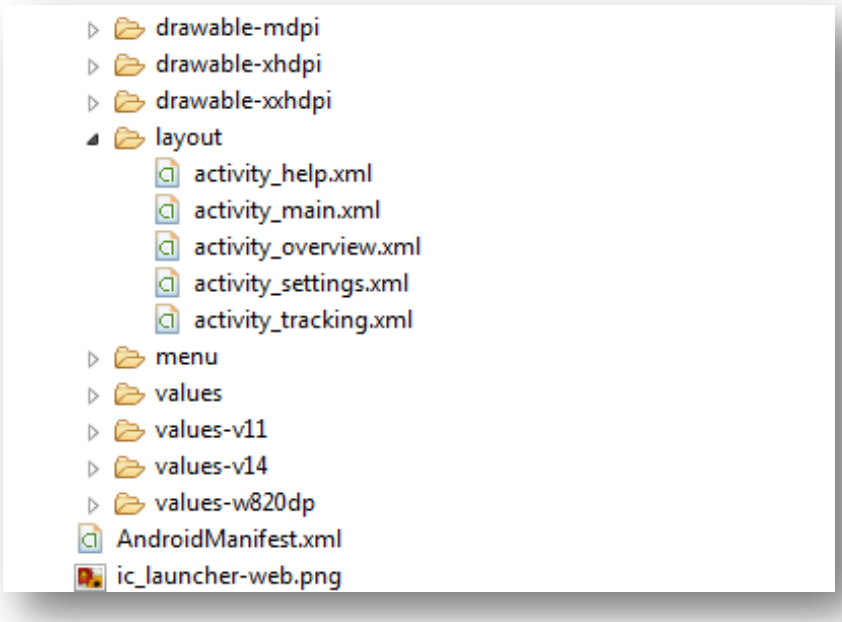
Soubor textových řetězců, do kterého lze odkazovat z různých částí aplikace. Při změně se úprava zpropaguje na všechna místa, ze kterých je na řetězec odkazováno.



Obrázek 32 - soubor strings.xml

7.2.4. Layouts

Soubory ve formátu XML, obsahující strukturu GUI pro každou aktivitu.



Obrázek 33 - umístění layout souborů ve struktuře projektu

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:background="@color/background_blue"
    tools:context="cz.uhk.fim.dragonboatassistant.TrackingActivity" >

    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/tracking_activity_text" />

    <fragment
        android:id="@+id/map2"
        android:name="com.google.android.gms.maps.MapFragment"
        android:layout_width="match_parent"
        android:layout_height="170dp"
        android:layout_below="@+id/textView1"
        android:layout_marginTop="5dp" />

    <TextView
        android:id="@+id/textBalanceCenter"
        android:layout_width="wrap_content"
```

Obrázek 34 - ukázkový příklad kódu layout souboru

7.3. Implementace

V následující části bude předvedena konkrétní implementace a řešení hlavních problémů.

7.3.1. GoogleAPIActivity

V následující ukázce je vidět implementace přístupu ke službám Google a dále je zde nastaveno, aby se neuzamykala obrazovka, pokud se uživatel nachází v aktivitě, pomocí **getWindow().addFlags()**.

GoogleAPIActivity.java

```
1  package cz.uhk.fim.dragonboatassistant.core;
2
3  import com.google.android.gms.common.ConnectionResult;
27
28 public class GoogleAPIActivity extends Activity
29         implements ConnectionCallbacks, OnConnectionFailedListener, OnMapReadyCallback, LocationListener {
30
31     GoogleApiClient googleApiClient;
32     public Location lastKnownLocation;
33     LocationRequest locationRequest;
34     protected MapFragment mapFragment;
35     protected GoogleMap map;
36
37
38     @Override
39     protected void onCreate(Bundle savedInstanceState) {
40         super.onCreate(savedInstanceState);
41
42         getWindow().addFlags(WindowManager.LayoutParams.FLAG_KEEP_SCREEN_ON);
43
44         // Create an instance of GoogleAPIClient.
45         System.out.println("Vytvářím instanci GoogleAPIClient");
46         if (googleApiClient == null) {
47             googleApiClient = new GoogleApiClient.Builder(this).addConnectionCallbacks(this)
48                     .addOnConnectionFailedListener(this).addApi(LocationServices.API).build();
49         }
50         createLocationRequest();
51     }
52
53     @Override
54     protected void onStart() {
55         googleApiClient.connect();
56         super.onStart();
57     }
58
```

Obrázek 35 - ukázka kódu z GoogleAPIActivity pro přístup ke službám Google

V metodě onStart() je volána metoda **createLocationRequest()**, kde se parametrizuje příjem informací o poloze zařízení.

```
protected void createLocationRequest() {
    // Create an instance of LocationRequest.
    locationRequest = new LocationRequest();
    locationRequest.setSmallestDisplacement(DISPLACEMENT);
    locationRequest.setInterval(10000);
    locationRequest.setFastestInterval(5000);
    locationRequest.setPriority(LocationRequest.PRIORITY_HIGH_ACCURACY);
    locationRequest.setPriority(100);

    LocationSettingsRequest.Builder builder = new LocationSettingsRequest.Builder()
            .addLocationRequest(locationRequest);

    PendingResult<LocationSettingsResult> result = LocationServices.SettingsApi
            .checkLocationSettings(googleApiClient, builder.build());
```

Obrázek 36 - ukázka kódu z GoogleAPIActivity

7.3.2. TrackingActivity

Toto je aktivita implementující hlavní funkcionalitu aplikace. Následuje ukázka kódu pro práci s polohou zařízení. V rámci každého průběhu metody onLocationChanged() probíhá přepočet celkové vzdálenosti, dále aktuální a průměrné rychlosti.

```
214     @Override
215     public void onConnected(Bundle connectionHint) {
216         super.onConnected(connectionHint);
217         if (lastKnownLocation != null) {
218             if (from == null) {
219                 from = lastKnownLocation;
220                 fromTime = System.currentTimeMillis();
221                 to = lastKnownLocation;
222             }
223         }
224     }
225
226     @Override
227     public void onLocationChanged(Location location) {
228         super.onLocationChanged(location);
229         accuracy = location.getAccuracy();
230         if (tracking) {
231             if (accuracy < DISPLACEMENT) {
232                 to = location;
233                 toTime = System.currentTimeMillis();
234                 dist = dist + from.distanceTo(to);
235                 speed = getSpeedInMPS(from, to, fromTime, toTime);
236                 forecastTimeFor(FC_DISTANCE);
237                 // max speed
238                 if (speed > maxSpeed) maxSpeed = speed;
239                 // avg speed (total)
240                 avgSpeedTotal = (elapsedDistMeters + dist) / (elapsedTimeSecs + seconds);
241                 from = location;
242                 fromTime = System.currentTimeMillis();
243             }
244         }
245         updateMap(location, map);
246         renderTextViews();
247     }
248
249     protected void updateMap(Location loc, GoogleMap map) {
250         System.out.println("debug update map");
251         super.updateMap(loc, map);
252         if (tracking && (accuracy < DISPLACEMENT)) {
253             lineOptions.add(new LatLng(loc.getLatitude(), loc.getLongitude()));
254             map.addPolyline(lineOptions);
255         }
256     }
257
```

Obrázek 37 - ukázka kódu z TrackingActivity pro práci s polohou zařízení

Další ukázkou kódu je ukázka práce s gravitačním senzorem. V metodě **onStart()** je určeno, s jakou frekvencí se mají data ze senzoru aktualizovat. V rámci každé aktualizace dat ze senzoru je zkontrolováno, zda náklon nepřekročil určitou mez, do které je považován náklon za správný. Dle toho je upraveno GUI a přepočítán čas aktivity ve správném náklonu.

```java
@Override
protected void onStart() {
    super.onStart();
    sManager.registerListener(this, sManager.getDefaultSensor(Sensor.TYPE_GRAVITY),
            SensorManager.SENSOR_DELAY_FASTEST);
}

@Override
public void onSensorChanged(SensorEvent event) {

    if (event.accuracy == SensorManager.SENSOR_STATUS_UNRELIABLE) {
        System.out.println("Senzor je nyní nespolehlivý");
        return;
    }
    if (sensor.getType() == Sensor.TYPE_GRAVITY) {
        if (calibrate) {
            calibration = (SENSITIVITY * event.values[0]) * (-1);
            Toast.makeText(TrackingActivity.this,"Byla provedena kalibrace",Toast.LENGTH_LONG).show();
            calibrate = false;
        }
        balanceAngle = (SENSITIVITY * event.values[0]) + calibration;
        seekBar.setProgress((int) (50 - balanceAngle));
        System.out.println(balanceAngle);
        THRESHOLD = (SENSITIVITY / 2) - 13;
        if (balanceAngle > THRESHOLD) {
            excessiveTilt = true;
            textBalance.setBackgroundColor(Color.RED);
            balanceColorRight.setBackgroundColor(Color.TRANSPARENT);
        }
        if (balanceAngle < THRESHOLD && balanceAngle > (-THRESHOLD)) {
            excessiveTilt = false;
            textBalance.setBackgroundColor(Color.TRANSPARENT);
            balanceColorRight.setBackgroundColor(Color.TRANSPARENT);
            textBalanceCenter.setBackgroundColor(Color.GREEN);
        }
        if (balanceAngle < (-THRESHOLD)) {
            excessiveTilt = true;
            textBalance.setBackgroundColor(Color.TRANSPARENT);
            balanceColorRight.setBackgroundColor(Color.RED);
        }
    }

    if (tracking) {
        totalCounter ++;
        if (!excessiveTilt) corectCounter ++;
        recountTime();
    }
}
```

Obrázek 38 - ukázka kódu z TrackingActivity pro práci s gravitačním senzorem

```java
public void recountTime() {
    seconds = millisToSecsConverter.convertTime(getSessionTime(), TimeUnit.SECONDS, TimeUnit.MILLISECONDS);
    mins = secsToMinsConverter.convertTime(elapsedTimeSecs + seconds, TimeUnit.MINUTES, TimeUnit.SECONDS);
    if (mins > 0)
        textTime.setText("Celkový čas: " + mins + "min " + ((elapsedTimeSecs + seconds) - mins * 60) + "sec");
    else
        textTime.setText("Celkový čas: " + (elapsedTimeSecs + seconds) + "sec");

    if (totalCounter > 0) {
        tiltPercentage = (100 * corectCounter) / totalCounter;
    }
    percentageText.setText("Čas aktivity ve spr. náklonu: " + tiltPercentage + "%");
}
```

Obrázek 39 - ukázka kódu metody recountTime() z TrackingActivity

Nyní budou následovat ukázky kódu z pomocných tříd.

7.3.3. Converter

Jsou zde obsaženy metody pro převod mezi jednotkami času, převod rychlosti v m/s na km/h a dále určení rychlosti ze vzdálenosti a časového úseku mezi dvěma určenými polohami zařízení.

```
1  package cz.uhk.fim.dragonboatassistant.util;
2
3  import java.util.concurrent.TimeUnit;
4
5  import android.location.Location;
6
7  public class Converter {
8
9      TimeUnit convertFrom;
10     TimeUnit convertTo;
11     private Converter millisToSecsConverter;
12
13     public long convertTime(long time) {
14         return convertTo.convert(time, convertFrom);
15     }
16
17     public long convertTime(long time, TimeUnit convertTo, TimeUnit convertFrom) {
18         return convertTo.convert(time, convertFrom);
19     }
20
21     public float convertToKmPerHour(float mPerSec) {
22         return (float) (3.6 * mPerSec);
23     }
24
25     public float getSpeedInMPS(Location from, Location to, long timeFrom, long timeTo) {
26         float distance = from.distanceTo(to);
27         long time = millisToSecsConverter.convertTime((timeTo - timeFrom), TimeUnit.SECONDS, TimeUnit.MILLISECONDS);
28         float speed = distance / time;
29         return speed;
30     }
31
```

Obrázek 40 - ukázka kódu některých metod třídy Converter

7.3.4. Forecast

Metoda slouží k přepočtu aktuální rychlosti na odhadovaný čas projetí trati zadané délky.

```java
package cz.uhk.fim.dragonboatassistant.util;

import java.util.concurrent.TimeUnit;

public class Forecast {

    float distance;
    Converter timeConverter = new Converter();
    public long convertedSeconds;
    public long convertedMinutes;

    public long forecastSeconds(float speed) {

        long seconds = 0;
        if (speed > 0) {
            seconds = (long) (distance / speed);
        }
        return seconds;
    }

    public void forecastMinsSeconds(float speed) {

        long seconds = forecastSeconds(speed);
        long mins = 0;
        mins = timeConverter.convertTime(seconds, TimeUnit.MINUTES, TimeUnit.SECONDS);
        if (mins > 0) {
            convertedMinutes = mins;
            convertedSeconds = seconds - (mins*60);
        } else {
            convertedMinutes = 0;
            convertedSeconds = seconds;
        }
    }
```

Obrázek 41 - ukázka kódu pomocné třídy Forecast

7.4. Použití aplikace

V následující kapitole bude představena výsledná verze aplikace a způsob jejího použití.

7.4.1. Spuštění aplikace

Po nainstalování aplikace do mobilního zařízení se spouštěcí ikona zobrazí mezi ostatními aplikacemi.

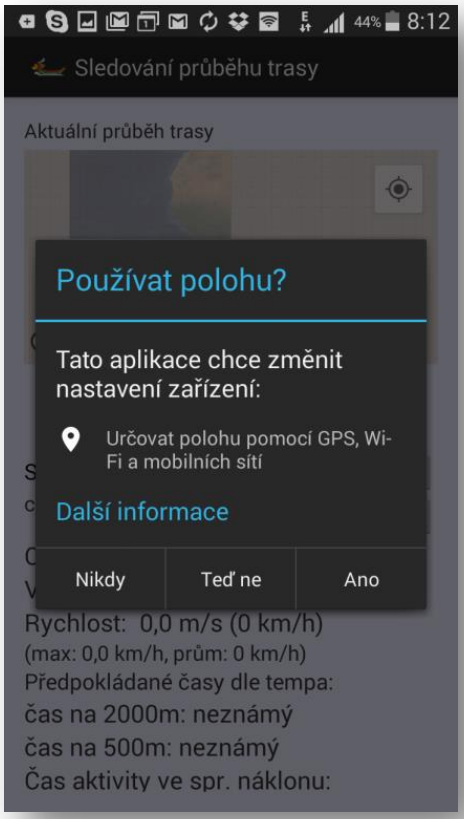
Obrázek 42 - ikona aplikace na ploše mobilního zařízení

Po spuštění aplikace je uživateli zobrazena hlavní nabídka.

Obrázek 43 - hlavní nabídka aplikace

7.4.2. Sledování aktivity

Po výběru nabídky **Začít aktivitu** aplikace přejde do sledovací části aplikace obsahující hlavní funkcionalitu aplikace. Ihned po přechodu do sledovací části aplikace zkontroluje nastavení zařízení. Pokud nastavení zařízení neodpovídá očekávanému stavu pro správnou funkci aplikace, je uživateli zobrazena informace s možností odsouhlasit automatické provedení potřebných změn.



Obrázek 44 - dialog při nedostatečném nastavení zařízení

Po odsouhlasení změn uživatelem aplikace provede potřebné změny v nastavení a pokračuje dále na obrazovku sledovací části. Po příchodu na obrazovku je zatím samotné sledování zastaveno. To znamená, že se nesleduje celkový čas aktivity, ani vzdálenost a tím pádem ani rychlost. Senzory jsou však aktivní po celou dobu přítomnosti uživatele na sledovací obrazovce.

I při vypnutém režimu sledování je sledována pozice a náklon zařízení. Je tomu tak ze dvou důvodů:

- dostatek času na vyhledání pozice zařízení před spuštěním sledování
- možnost řádně a v klidu vyvážit loď a zkalibrovat náklon v aplikaci

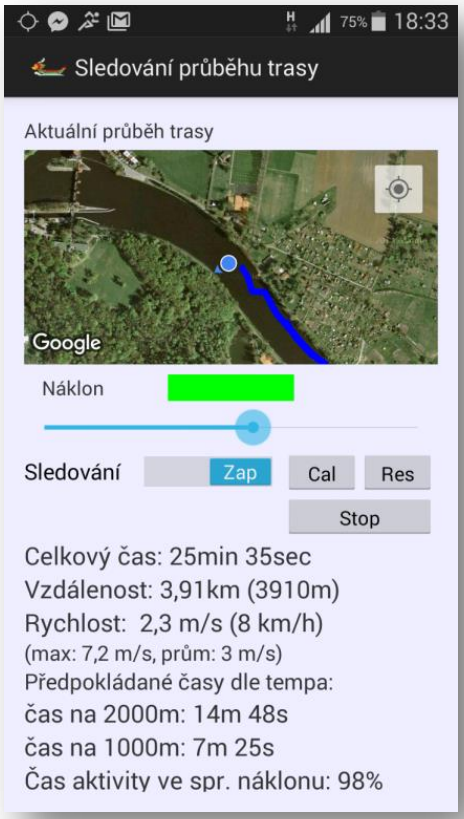
Náklon je znázorněn posuvníkem a barevnými políčky, kdy při správném náklonu je rozsvícena pouze zelená část a kulička posuvníku se nachází na jeho úrovni. V případě překročení meze správného náklonu se rozsvítí červená část na příslušné straně. Pokud po umístění zařízení do polohy, která má být považována za vodorovnou, není kulička

posuvníku na úrovni středu zeleného pole, je potřeba náklon nejprve zkalibrovat pomocí tlačítka **Cal**.

Obrázek 45 - sledovací část aplikace před započetím sledování

Pokud je vše připraveno, zkalibrován náklon a aplikace již zjistila polohu zařízení, je možné spustit a případně později řídit samotné sledování. Uživatel má tedy následující možnosti:

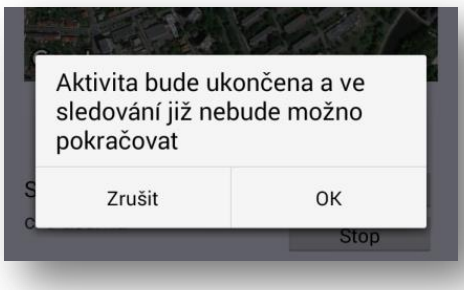
- spustit režim sledování
- pozastavit spuštěné sledování
- zrušit a vynulovat probíhající sledování tlačítkem **Res**
- kalibrovat náklon v případě potřeby tlačítkem **Cal**
- ukončit celé sledování pro vyhodnocení tlačítkem **Stop**



Obrázek 46 - sledovací část aplikace během režimu sledování

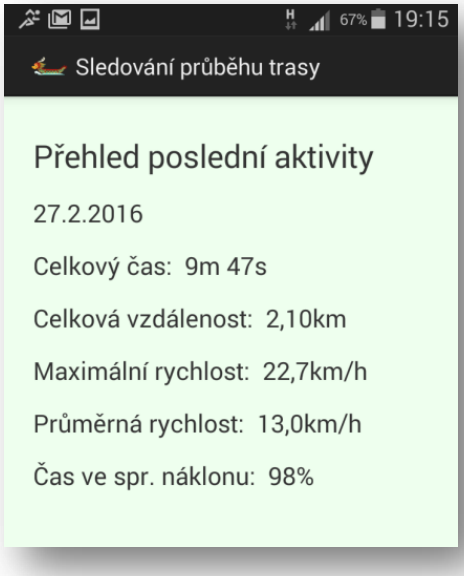
7.4.3. Ukončení aktivity

Po ukončení tréninku následuje vyhodnocení aktivity. Aktivita se ukončí stiskem tlačítka **Stop**. Uživateli je zobrazen dialog, ve kterém je dotázán, zda opravdu souhlasí s ukončením aktivity.



Obrázek 47 - dialog při ukončování probíhající aktivity

Pokud uživatel odsouhlasí ukončení aktivity, již nejde toto konkrétní sledování opět spustit a pro další sledování bude uživatel začínat opět od nuly. Aplikace přejde na novou obrazovku, kde mu je zobrazen souhrn celé aktivity.



Obrázek 48 - vyhodnocení po ukončení aktivity

8. Shrnutí a závěr

V úvodu práce proběhla analýza dnes běžně dostupných a oblíbených sportovních aplikací a forem, v jakých jsou na trhu k sehnání. S tím souvisel i rozbor, na jakém principu tyto aplikace pracují a co uživateli přinášejí. Přestože bylo ukázáno, že senzory jsou v mobilních aplikacích velmi rozšířeny a nabízejí uživateli širokou škálu funkcí, ke konci dané kapitoly se dospělo k závěru, že většině z těchto aplikací chybí prvek kontroly správnosti provedení. Součástí další kapitoly byl tedy návrh, jak tímto prvkem obohatit sportovní aplikaci, která bude navržena pro vybranou sportovní aktivitu. Nejprve byl detailně představen dragonboating, jako vybraná sportovní aktivita, spolu s návrhem, jak by se v rámci něj daly mobilní senzory využít. Z návrhu vyplynuly konkrétní typy senzorů, které lze využít k návrhu sportovního asistenta, splňujícího požadavky vybrané sportovní aktivity. Následující kapitoly se týkaly těchto vybraných typů senzorů, konkrétně jejich HW principů a dále programového API pro práci s těmito senzory v rámci vyvíjené aplikace. V praktické části práce byla nejprve představena analýza vzorové aplikace a dále i samotná implementace včetně závěrečného předvedení práce s hotovou aplikací.

Cílem práce bylo přenést možnosti, plynoucí z integrace senzorů do mobilních zařízení, i do oblasti vybraného sportu a to se podařilo. Konkrétně navrhnout a implementovat sportovního asistenta, který nabídne více než běžné, z ostatních sportovních aplikací známé, údaje o uplynulém čase a vzdálenosti. Pro vybranou sportovní aktivitu bylo z důvodu jejích zásad podstatné kontrolovat i správný náklon lodě. Toto se také podařilo naplnit. Aplikace je schopná určit, jaká část tréninku pobíhala ve správném náklonu, tedy dle zásad správně. Pro posádku z tohoto údaje plyne poznání, po jakou procentuální část tréninku docházelo ke zbytečně vysokému úbytku sil a jak důležité je dohledat příčinu tohoto jevu a tu co nejdříve odstranit. Aplikace již je v reálném provozu a je možné jí dále vylepšovat a systém rozšiřovat. Možnosti vylepšení a rozšíření systému byly v průběhu práce představeny.

Zdroje

Analog Devices. (2009). Získáno 2016, z Web Analog: http://www.analog.com/media/en/technical-documentation/data-sheets/ADXL325.pdf

Bílá kniha o sportu. (2007). Brusel: Komise evropských společenství.

Djuknic, G., & Richton, R. (2 2001). Geolocation and assisted GPS. *Computer*.

Fastest growing mobile app categories 2015. (8. 1 2016). Získáno 1 2016, z Web Statista.com: http://www.statista.com/statistics/251096/fastest-growing-shopping-app-categories/

Google. (2016). *Building Your First App*. Načteno z Web Android.com: http://developer.android.com/training/basics/firstapp/index.html

Google. (2016). *Keeping the Device Awake*. Načteno z Web Android.com: http://developer.android.com/training/scheduling/wakelock.html

Google. (2016). *Sensors overview*. Získáno 2016, z Web Android.com: developer.android.com/guide/topics/sensors/sensors_overview.html

Hamřík, M. (28. Srpen 2012). *Sportovní senzory budou přesnější.* Získáno 2016, z Web E15.cz: http://magazin.e15.cz/sport/sportovni-senzory-budou-presnejsi-908913

Chang, E. Y. (2011). *Sensor-aided mobile information management and retrieval.* Získáno 2016, z Web ACM.org: http://dx.doi.org/10.1145/2009916.2010187

Jindal, A., Pathak, A., Hu, C., & Midkiff, S. (2013). *Hypnos: understanding and treating sleep conflicts in smartphones.* Získáno 2016, z Web ACM.org: http://dx.doi.org/10.1145/2465351.2465377

Juránek, M. (2007). Získáno 2016, z Web HomeL VŠB: http://homel.vsb.cz/~jur286/prostredky_aut_rizeni/preklad.htm

Jurišica, L., Vitko, A., Duchoň, F., & Kaštan, D. (2011). Systém GPS - princíp, prednosti a nedostatky. *Automa*.

Krauzová, M. (2014). *Dračí lodě - nové sportovně rekreační odvětví s potenciálem ovlivnění životního stylu člověka.* Získáno Březen 2016, z Web Theses.cz: http://theses.cz/id/kgcs9j/BP_KrauzovM_Dra_lod.pdf

Mrkva, V. (13. Prosinec 2004). *GARMIN GPS ruční-turistické-sportovní-auto-paragliding-ultralight-námořní-fitnes.* Získáno 2016, z Flytex.cz: http://www.flytex.cz/turisticke%20sport.htm

PARK, S., Dongwon, K., & Hojung, C. (2015). *Reducing Energy Consumption of Alarm-induced Wake-ups on Android Smartphones.* Načteno z Web ACM.org: http://doi.acm.org/10.1145/2699343.2699346

Perič, T. (2008). *Sportovní příprava dětí.* Praha: Grada Publishing.

Perič, T., & Dovalil, J. (2010). *Sportovní trénink.* Praha: Grada Publishing.

Person, C. (1999). *How a gyroscope works.* Získáno 2016, z Web Gyroscopes.org: http://www.gyroscopes.org/how%5Chagwa4.pdf

Rydval, S. (2005). *Princip a fungování GPS*. Získáno Březen 2016, z Web Rydval.cz: http://www.rydval.cz/phprs/view.php?cisloclanku=2005110301

SAMSUNG. (2015). *Technická specifikace: Galaxy S3 Neo*. Získáno 2016, z Web Samsung.com: http://www.samsung.com/cz/consumer/mobile-devices/smartphones/galaxy-s/GT-I9301MBIETL

Smejkal, J. (30. 11 2007). *Vše o Hardware - Vše o GPS*. Získáno 2016, z Web Vseohw.net/: http://vseohw.net/clanky/software/gps

Vítek, L. (28. Září 2015). *Přehled aplikací, zaměřených na sportovní aktivity.* Získáno 2016, z Sportvital.cz: http://www.sportvital.cz/zdravi/diagnostika/prehled-aplikaci-zamerenych-na-sportovni-aktivity/

Vojáček, A. (Říjen 2009). *Integrované MEMS GYROSKOPY*. Získáno 2016, z Web Automatizace.hw.cz: http://automatizace.hw.cz/integrovane-mems-gyroskopy

Vylegala, P. (2014). *Gyroskopy, akcelerometry a infračervené snímače.* Získáno 2016, z Web Sse-najizdarne.cz: http://www.sse-najizdarne.cz/projekty/roboti/dokumenty/v_prez_ss_5.pdf

Zeimpekis, V., Giaglis, G., & Lekakos, G. (12 2002). A taxonomy of indoor and outdoor positioning techniques for mobile location services. *ACM SIGecom Exchanges*.

Obrázky

Přílohy

Příloha 1

CD obsahující aplikaci Dragonboat Assistant v1.0 ve formě zabaleného Android projektu.