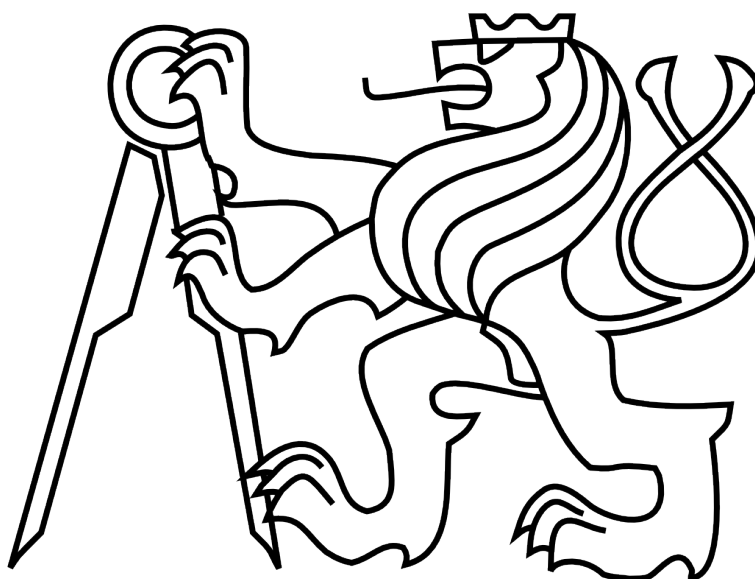


ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

Fakulta elektrotechnická

# Bakalářská práce



Jan Kubeš

Návrh diagnostického testu pro pacienty trpící  
Parkinsonovou poruchou

Katedra kybernetiky

Vedoucí projektu: Ing. Daniel Novák Ph.D

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

**Student:** Jan K u b e š

**Studijní program:** Kybernetika a robotika (bakalářský)

**Obor:** Robotika

**Název tématu:** Návrh diagnostického testu pro pacienty trpících Parkinsonovou poruchou

### Pokyny pro vypracování:


1. Seznamte se s problematikou diagnostiky pacientů trpících Parkinsonovou poruchou.
2. Naimplementujte vhodný diagnostický test na tablet či mobilní telefon.
3. Vytvořte jednoduchý webový systém pro zprávu pacientů
4. Ověřte funkčnost testu na vzorku deseti pacientů.

### Seznam odborné literatury:

- [1] Smith, D.; Friesen, J.: Android Recipes a Problem-Solution Approach, Apress, 2011
- [2] Růžička, E.: Parkinsonova nemoc. Čes. a slov. Neurol. Neurochir., 69/102, 2006, No. 4, p. 241–258.
- [3] Emre, M.; Aarsland, D.; Brown, R.; Burn, D. J.; Duyckaerts, C.; Mizuno, Y.; Brue, G. A.; Cummings, J.; Dickson, D. W.; Gauthier, S.; Goldman, J.; Goetz, C.; Korczyn, A.; Lees, A.; Levy, R.; Litvan, I.; McKeith, I.; Olanow, W.; Poewe, W.; Quinn, N.; Sampaio, C.; Tolosa, E.; Dubois, B.: Clinical Diagnostic Criteria for Dementia Associated with Parkinson's Disease. Mov Disord. 2007 Sep 15;22(12):1689-707; quiz 1837.

**Vedoucí bakalářské práce:** Ing. Daniel Novák, Ph.D.

**Platnost zadání:** do konce zimního semestru 2012/2013

  
prof. Ing. Vladimír Mařík, DrSc.  
vedoucí katedry



  
prof. Ing. Pavel Ripka, CSc.  
děkan

V Praze dne 9. 1. 2012

## Prohlášení

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

V Praze dne.....25.5. 2012.....

..........

## Poděkování

Chtěl bych poděkovat svému vedoucímu práce Ing. Danovi Novákovi Ph.D. za odborné vedení projektu a za pomoc při řešení problémů.

## Abstrakt

Cílem této bakalářské práce je vytvoření aplikace, která je schopná vyšetřit pacienty s esenciálním třesem. Inspiruje se testy typu tužka-papír, kde pacient kreslí do předtisknuté spirály a neměl by vybočit z předlohy.

Program je určen pro operační systém Android. Pacient je vyšetřen na základě obtažení šedivé spirály, výsledek je vyhodnocen aplikací. Jedním ze zajímavých ukazatelů je například celková úspěšnost tahu, důležité jsou i průměrná a maximální chyby, celkový čas tahu, ale zejména stanovení frekvence třesu. Tahy a jeho výsledky je možné uložit (formát JPEG). Konečný program byl vyzkoušen na dvou skupinách subjektů (starší, mladí) a výsledky byly statisticky zhodnoceny.

Po zhodnocení výsledků testovaných skupin byly zjištěny předpokládané signifikantní rozdíly mezi skupinami a zároveň byli odhaleny i některé nedostatky v programu, které způsobovaly nepřesnosti v měření. Pro uvedení do praxe je nezbytné zdokonalení aplikace, jež by mělo odstranit nepřesnosti v měření a zobjektivnit výsledky. Dalším potřebným krokem je stanovení průměrné hodnotící stupnice (například 1-5), čehož dosáhneme měřením širším vzorkem populace.

## Abstrakt

The goal of the thesis is implementation of application for essential tremor assessment. This project is inspired by paper diagnostic version where patient is follows spiral template..

The program is designed for Android's tablet. Patient is examine by go over the template spiral. The result is evaluate by application. One of interesting indicator is total succesful of draw, next important markers are average and maximum mistakes, total time of draw. Furthremore, one of the most important is frequency of tremor. Draws are able to save (format JPEG). The final program was tested on two groups of subjects (older, young) and the results weer evaluated by Kruskal Wallice test.

Presumed differences of significant value were founded after evaluation of results from test groups. Simultaneously, some imperfections in the application were found out, which created inaccuracy in measuring. In order to use the application in medical practices, it is necessary to upgrade it. This should remove imperfections in measuring and bring more objective results. The following necessity is to specify average rating scale (for example 1-5), which we will achieve by measuring larger sample of population.

# Obsah

<b>1</b>	<b>Úvod</b>	<b>1</b>
<b>2</b>	<b>Popis problému, specifikace cíle</b>	<b>2</b>
2.1	Parkinsonova nemoc . . . . .	2
2.2	Klinický obraz Parkinsonovy nemoci . . . . .	4
2.2.1	Základní příznaky PN . . . . .	4
2.2.2	Diagnóza Parkinsonovi nemoci . . . . .	6
<b>3</b>	<b>Analýza a návrh řešení</b>	<b>9</b>
3.1	Aplikace na mobilní telefony a tablet . . . . .	9
3.2	Lékařský test Pen&Paper . . . . .	12
3.3	Návrh řešení . . . . .	13
<b>4</b>	<b>Realizace</b>	<b>15</b>
4.1	Android a Asus EEE Pad Slider SL101 . . . . .	15
4.2	Vývoj programu . . . . .	17
4.2.1	Vizualizace . . . . .	17
4.2.2	Spirála a body . . . . .	21
4.2.3	Vyhodnocení bodů . . . . .	23
4.2.4	Vyhodnocení celého tahu . . . . .	27
4.2.5	Menu . . . . .	29
<b>5</b>	<b>Testování</b>	<b>31</b>
5.1	Skupina mladých lidí . . . . .	31
5.2	Skupina starých lidí . . . . .	34
5.3	Porovnání obou skupin . . . . .	38
<b>6</b>	<b>Závěr</b>	<b>40</b>
<b>7</b>	<b>Literatura</b>	<b>41</b>

## Seznam obrázků

1	Patologická stadia rozvoje Parkinsonovy nemoci [4] . . . . .	4
2	Ukázka aplikace TremorTracer [9] . . . . .	10
3	Ukázka aplikace Tremor-Meter [10] . . . . .	11
4	Ukázka aplikace Kinesia HomeView [11] . . . . .	12
5	Ukázka testu tužka-papír . . . . .	13
6	Asus EEE Pad Slider SL101 . . . . .	15
7	Příklad špatného překrývání tahu spirálou . . . . .	19
8	Šablona spirály . . . . .	23
9	Vyhodnocení vnitřku spirály . . . . .	26
10	Náhled na finální aplikaci . . . . .	30
11	Ukázka přetažení spirály . . . . .	33
12	Ukázka studenta číslo 14 . . . . .	34
13	Ukázka studenta číslo 1 . . . . .	34
14	Ukázka klientky s diagnostikovanou PN . . . . .	37
15	Ukázka klienta číslo 5 . . . . .	37
16	Chyba způsobená zápěstím u pokusu č. 13 . . . . .	38
17	Histogram pro procentuální úspěšnost . . . . .	44
18	Histogram pro maximální chybu . . . . .	44
19	Histogram pro průměrnou chybu . . . . .	45
20	Histogram pro čas . . . . .	45
21	Histogram pro třes . . . . .	46

## Seznam tabulek

1	Klinická diagnostická kritéria Parkinsonovy nemoci, UK Parkinson's Disease Society Brain Bank. . . . .	8
2	Tabulka výsledků mladých lidí . . . . .	32
3	Tabulka výsledků starých lidí . . . . .	36
4	Tabulka pro porovnání výsledků obou skupin . . . . .	38
5	Obsah přiloženého CD . . . . .	43



# 1 Úvod

S rychlým vývojem techniky se přemýšlí nad jejím užitím v různých odvětvích. S příchodem tabletů se diskutovalo o jejich možném využití a to jak v osobním životě, tak i v průmyslu a lékařství. Z počátku byli tyto přístroje velmi drahé a existovalo na ně jen málo aplikací. Dnes jsou již ceny přijatelnější a tablety získávají velké zastoupení.

Rozhodl jsem se, že má bakalářská práce se bude zabývat vyšetřením parkinsoniků, potažmo esenciálního třesu. Nástrojem pomáhajícím mi ve splnění tohoto cíle je tablet, který je ovládán dotykovým displayem. Program, který jsem vytvořil, má za úkol změřit a ohodnotit míru třesu a i na základě jiných parametrů pomoci k určení potřeby úplného vyšetření na nemoci projevujících se třesem.

Esenciální třes je nejčastějším druhem třesu, až 10x častější než Parkinsonova nemoc, a může se vyskytnout u lidí v každém věku. Nejvíce se však projevuje u lidí starších 40 let. Tato nemoc je většinou dědičná a probíhá velmi pomalu. Třes je statického rázu a jeho frekvence je asi dvakrát vyšší než u parkinsonického tremoru (tj. cca. 8-12 Hz). Často se projevuje třesem hlavy, někdy i hlasu (tremor hlasivek), ale hlavně třesem horních končetin [5, 7].

Parkinsonova nemoc většinou začíná mezi 50. a 60. rokem, asi 10% pacientů onemocní před 40. rokem života. Mezi hlavní příznaky této nemoci patří třes rukou i nohou (často se třes vyskytuje pouze na polovině těla), dále ztuhlost a zpomalenost. S rozvojem nemoci se objevují také psychiatrické (deprese), vegetativní poruchy (poruchy polykání a trávení, zvýšené slinění, zvýšené pocení). Příznaky Parkinsonovy nemoci jsou velmi dobře ovlivněny správnou léčbou. Tyto poruchy jsou pro pacienty nepříjemné a to především v lehké motorice, kdy mají problém s jídlem, oblékáním, hygienou. S postupem času i s řečí a mluvením [5].

Obě tyto nemoci se projevují hlavně třesem horních končetin, na což se má práce upínat.

## 2 Popis problému, specifikace cíle

V této kapitole se budu zabývat podrobnějšími popisy nemocí spojených s třesem, problémy, s kterými se nemocný setkává v běžném životě a co je potřeba k tomu, aby mohl být pacient vyšetřen, případně jak se dá pacientovi pomoci.

### 2.1 Parkinsonova nemoc

Jelikož nejsem doktor, tak se v otázkách definic a příznaků obrátím na různé články, kde se této nemoci věnují odborníci. Následující definice je ze článku Prof. MUDr. Evžena Růžičky, DrSc. o Parkinsonově nemoci a všechny text zabývající se touto nemocí je parafrází z tohoto článku [2].

„Parkinsonova nemoc (PN) je chronické progresivní onemocnění nervové soustavy. Vzniká na podkladu degenerativního zániku neuronů v pars compacta substantiae nigrae a v dalších pigmentovaných jádrech kmene mozku, v jehož důsledku dochází k nedostatku dopaminu (DA) a jiných neuromediátorů v bazálních gangliích mozku. PN se projevuje charakteristickou poruchou hybnosti, tzv. extrapyramidovým hypokineticko-rigidním (parkinsonským) syndromem skládajícím se z hypokineze, rigidity, třesu a posturální poruchy, odpovídajícím na dopaminergní léčbu. Dalšími projevy PN mohou být vegetativní, senzorické, afektivní a kognitivní poruchy.“ Prevalence<sup>1</sup> PN se pohybuje kolem jednoho až dvou případů na tisíc obyvatel. U skupiny lidí starších 60 let se zvyšuje až na jeden případ na sto obyvatel. Průměrný věk počátku onemocnění je kolem 60 let, ale asi 10% pacientů onemocní před 40. rokem a dalších 10% onemocní až po 75. roce života.

V posledních desetiletích výzkum a četné genetické a molekulárně biologické studie ukázali specifické genové defekty u různých forem familiárně vázané PN. Bylo objeveno několik genů a mutovaných proteinů, které zapříčiňují určitou dědičnost. Také objasňují různorodost nemoci co se týče začátku výskytu u pacienta, rychlosti, kterou se nemoc projevuje a charakteru příznaků. Proto se také předpokládá, že u části pacientů je nemoc geneticky vázané podmíněné onemocnění, ve kterém se hromadí vliv genetických faktorů s faktory ze zevního prostředí. Specifické mutace pro dopa-responzivní parkinsonismus byli identifikovány v několika různých genech (např. alfa-synuklein, parkin, UCH-L1, PINK, DJ1, LRRK2), jejich biochemické a molekulární vlastnosti svědčí pro patogenetický mechanismus PN. Důležitou úlohu v něm zaujímají změny v procesech skládání buněčných proteinů (folding) a v procesech jejich degradace v proteasomech po označení ubikvitinem <sup>2</sup>, procesy s mutovanými proteiny s abnormální konformací - nesložené (unfolded) nebo chybně složené (misfolded). Ty se shlukují, tudíž vážne jejich odstraňování z neuronů a tvoří se buněčné inkluze (odpad).

---

<sup>1</sup>Prevalence je podíl počtu jedinců trpících danou nemocí a počtu všech jedinců ve sledované populaci.

<sup>2</sup>Mechanismus zajišťující odstraňování „opotřebovaných“ buněčných proteinů

U pacientu se sporadickou PN se v substantia nigra <sup>3</sup> ukazuje hromadění oxidativně poškozených bílkovin zároveň s poruchou funkce proteasomů. Podobné testy se zkusily i s pacienty s geneticky vázanou PN. Ve shodě s těmito nálezy se ukazuje, že vadná proteolýza je jedním z klíčových prvků onemocnění a také odpovídá za jeho věkově vázaný výskyt.

Alfa-synuklein je například protein, který se zřejmě podílí na normální synaptické plasticitě. V důsledku mutace tohoto genu dochází k tvorbě defektní bílkoviny s tendencí k oligomerizaci a k agregaci. Tato vzniklá znečištění se hromadí v Lewyho tělískách<sup>4</sup>, jež se považují za typickou neuropatologickou známku zániku neuronů u PN. Některé práce ovšem dokazují, že Lewyho tělíska mohou ve skutečnosti být obranný mechanismus těla, kde se tělíska snaží nečistoty vtáhnout do sebe a ochránit tak přežívající neurony. Hromadění alfa-synukleinu a jeho fragmentů není výhradním ukazatelem u PN, nachází se i u jiných neurodegenerativních onemocnění, například u demence s Lewyho tělisky, Alzheimerovy nemoci a u Downova syndromu. Nejspíše kvůli jeho poměrně vysoké koncentraci v mozku a jeho schopnosti reagovat s mnoha bílkovinami, se alfa-synuklein objevuje u mnoha neurodegenerativních procesů, kde dochází k agregaci proteinů.

Parkin, narozdíl od alfa-synukleinu, je bílkovina, jehož mutace nevykazuje tvorbu Lewyho tělísek ani jiných inkluzí. Přesto se i v její patogenezi ukazuje vadná funkce ubikvitin-proteasomového komplexu. Parkin a ubikvitin-C-hydroláza (UCH-L1) jsou na rozdíl od alfa-synukleinu protektivní faktory. Zajišťují odstraňování vadných proteinů, ale zároveň zabraňují nepřiměřené degradaci funkčních součástí neuronu. V poslední době se však objevily práce, kde se zásadním způsobem mění pohled na genetiku onemocnění. Jedná se o nově nalezené mutace genu pro enzym „leucine-rich repeat kinase 2“ (**LRRK2**), také často nazývaný dardarin. V posledních letech bylo popsáno asi 20 různých mutací tohoto genu a to jak u pacientů s dominantně dědičnou PN, tak i se sporadickou formou. Penetrance genu je věkově vázaná. LRRK2 je v neuronech v celém mozku a jeho rolí je regulace signálového přenosu zajišťujícího správnou funkci neuronů. LRRK2 se považuje za klíčový prvek neuronálního poškození u PN, nadřazeným ostatním dříve popsaným mutacím proteinů. Hypoteticky by mohly terapie zaměřené na LRRK2 zpomalit nebo zcela zastavit rozvoj neurodegenerativních procesů a to nejen u PN.

Důsledkem poškození neuronů různých typů je u PN velmi komplexní a tím se odlišuje od ostatních synukleinopatií<sup>5</sup>. Braak se spolupracovníky [4] stanovil šest stadií ve vzestupném pořadí (od prodloužené míchy až po neocortex) (Obrázek 1). V počáteční fázi (1 a 2) této škály je postiženo dorsální motorické jádro n. vagus, bulbus olfactorius<sup>6</sup> a příslušná část nucleus olfactorius

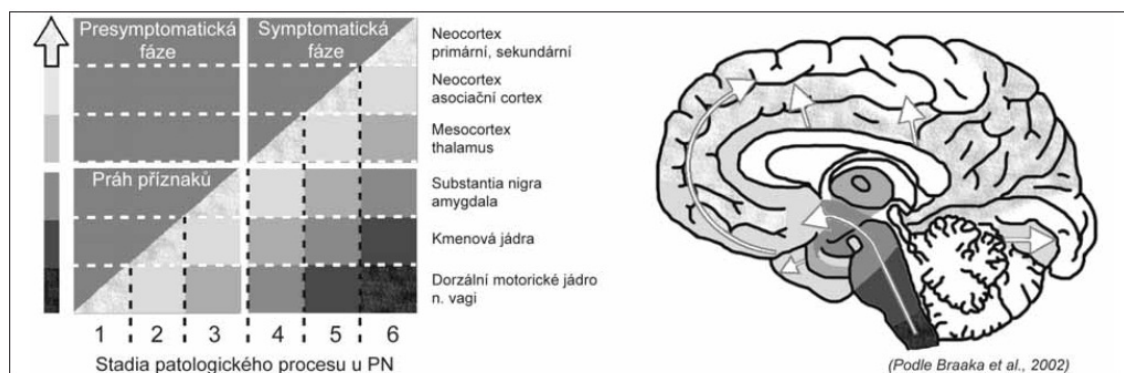
<sup>3</sup>Substantia nigra je párová struktura ve středním mozku, je součástí bazálních ganglií a hraje významnou roli v řízení pohybu

<sup>4</sup>Lewyho tělíska – histopatologický mikroskopický obraz vyskytující se v mozku u některých chorob, zejm. u Parkinsonovy nemoci a u Alzheimerovy nemoci v kůře mozkové.

<sup>5</sup>Synukleinopatie je onemocnění s agregací nerozpustného alfa-synukleinu v neuronech

<sup>6</sup>Bulbus olfactorius neboli čichový bulbus je část mozku obratlovců, který se vychyluje z přední strany hemisfér koncového mozku. Přichází do něj informace z čichových orgánů

anterior. Následně ve 3. stadiu jsou postiženy locus coeruleus<sup>7</sup> a substantia nigra. Ve 4. stadiu postupuje patologický proces dále do mesencefala a postihuje magnocelulární cholinergní jádra, dále je postižen i temporální mesocortex. Poslední fáze v 5. a 6. stupni je patologický proces plně vyvinut a postihuje i primární asociační oblasti neocortexu. PN je tedy plynule progresivní neurodegenerativní proces, postupně postihující řadu mozkových systémů. Braak tvrdí, že motorické i nonmotorické klinické projevy tohoto procesu (poruchy čichu, kognitivní poruchy, poruchy chování v REM spánku, autonomní disfunkce a pod.) jsou projevy progresu základního patologického procesu a nejen „pozdními komplikacemi“. Rychlost progresu a míra postižení je závislá na typu onemocnění. Pokud se tato teorie potvrdí, bude pohled na PN podstatně změněn a to hlavně s méně nadějnou vývojovou prognózou pro nemocné. Jelikož při diagnóze PN by při dostatečně dlouhém přežití pacienta nastaly i všechny výše zmíněné non-motorické projevy.



Obrázek 1: Patologická stadia rozvoje Parkinsonovy nemoci [4]

## 2.2 Klinický obraz Parkinsonovy nemoci

Klinický obraz jest souhrn příznaků a to jak subjektivních tak objektivních, které charakterizují určitou nemoc. Klinický obraz se dělí na typický a atypický. Typický klinický obraz popisuje nemoc tak, jak je pro ni charakteristické a atypický je-li průběh odlišný, například když má pouze některé příznaky shodné a jiné chybí. [8]

### 2.2.1 Základní příznaky PN

V počáteční fázi nemoci se PN projevuje jen velmi nepostřehnutelně. Toto období trvá až několik let. Podle Braakovy teorie, která jest zmíněna výše, zde probíhají preklinická stadia 1 a 2. Vyznačuje se převážně poruchou čichu a špatným REM spánkem, může za to postižení příslušných struktur bulbus olfactorius a jader dolního kmene. Jedním z dalších počátečních příznaků může být deprese, která velmi ztěžuje diagnózu PN. V následujících stádiích 3 a 4 již dochází k poškození

<sup>7</sup>Locus coeruleus je protáhlá skupina buněk ve stěně horní části IV. mozkové komory namodralého vzhledu

substantia nigra a k nedostatku dopaminu ve striatu. Zde je rozeznání choroby obtížné, neboť se nemoc projeví až tehdy, když počet dopaminergních neuronů a hladina dopaminu ve striatu poklesne pod určitou hranici. Většinou je tato hranice pod 50 %, ale některé práce tvrdí až pod 30 % normy. Poté obvykle nastávají necharakteristické somatické příznaky, jako kloubní a svalové bolesti. Až po čase se objeví typická motorická symptomatika, což je hypokineze, rigidita, popřípadě klidový třes a posturální poruchy.

Z těchto základních příznaku motoriky pacienta nejvíce omezuje hypokineze<sup>8</sup>, příbuzné projevy bradykineze<sup>9</sup> a akineze<sup>10</sup>. Hypokineze se projevuje alespoň zpočátku pouze jako jednostranná, nebo asymetrická. V důsledku těchto pohybových omezení má pacient problémy s řadou každodenních činností, jako je například jídlo, hygiena a oblékání, ale i nákup a otvírání dveří se stává problémem. Proto potřebují tito lidé pomoc od svých partnerů, či příbuzných. Dalšími časnými hypokinetickými projevy jsou mikrografie (zmenšení písma), hypomimie (otupělý výraz obličeje), hypofonie (mimořádně tichá řeč), aprosodie (nemelodická řeč), případně tachyfémie (zrychlení tempa řeči) a palilálie (opakování posledních slabik nebo celých slov).

Rigidita, neboli stuhlost, obvykle provází hypokinezi. Může se objevit v počátku PN jako svalové bolesti a ztuhlost končetin. Rigidita zasahuje axiální svalstvo a flexory, což postihuje držení šíje, trupu a končetin. Rigidita ustupuje ve spánku. Zvýšení klidového napětí a ztuhlost svalů kladou odpor v aktivním i pasivním pohybu, často se přirovnává ke kladení odporu ohýbané trubice z měkkého kovu (fenomén olovené trubice), spolu se svalovými náskoky a zárazy v průběhu pasivního pohybu (fenomén ozubeného kola).

Jeden z nejtypičtějších příznaku PN je třes. Ten je klidového rázu a převážně postihuje končetiny s frekvencí 4-6 Hz, který mizí při volném pohybu a při spánku. Třes se zvyšuje, pokud je pacient vystaven stresu, mentálnímu úsilí a únavě. Pokud je přítomen klidový asymetrický třes ruky, je pravděpodobná diagnóza PN. Nepřítomnost třesu diagnózu nevylučuje. Třes nepostihuje hlavu, třes dolní čelisti, jazyka či rtů je výjimkou.

Tíže postižení nemocného a jeho kvalita života závisí na posturálních poruchách (flekční držení trupu, nejistota ve stoji, šouravá chůze drobnými kroky, zárazy). Objevuje se zde i „akinetický freezing“, který se projevuje náhlými pohybovými bloádami při chůzi, zárazy v úzkých prostorech a změně směru chůze. Na počátku chůze je hesitace (váhání, neschopnost vykročit). V průběhu chůze se může stát, že pacient zrychlí a zkrátí krok, čemuž se říká festinace.

V této podkapitole jsem se zabýval příznaky PN a důsledky, které každý pacient zažívá. V následující části se zaměřím na diagnózu nemoci a chybám při diagnóze.

---

<sup>8</sup>Hypokineze - omezení rozsahu pohybu

<sup>9</sup>Bradykineze - pohybové zpomalení

<sup>10</sup>Akineze - porucha startu pohybu

### 2.2.2 Diagnóza Parkinsonovi nemoci

Jak je výše uvedeno, PN je charakterizována třemi hlavními příznaky. Jsou to hypokineze, rigidita a klidový třes. PN je klinicky pravděpodobná, pokud jsou přítomny nejméně dva ze tří těchto hlavních příznaků. Nemoc má plynule progresivní průběh, takže pokud nejsou nalezeny žádné další příznaky, které jsou obvyklé pro jiné nemoci a pokud není prokázána jiná choroba, která by mohla vyvolat parkinsonský syndrom, je velmi pravděpodobné, že pacient opravdu trpí PN.

Jedním z hlavních testů pro PN je ověření odpovídavosti na dopaminergní podnět. Pokud dojde k ústupu motorických příznaků po dávce levodopy nebo agonisty dopaminu, je tím potvrzena zachovávaná reaktivita dopaminových receptorů ve striatu a tím pádem má porucha hybnosti presynaptickou příčinu a diagnóza PN je téměř jistá. Za úspěšný test bereme zlepšení motorického skóre o 25 % nebo více. Pokud pacient nereaguje na dopaminergní podnět, svědčí to pro postsynaptické postižení striata a jedná se o jinou nemoc, která se projevuje parkinsonským syndromem. Pokud nedojde ke zlepšení příznaků po čtyřtýdenním podávání levodopy, je nutné přemýšlet o jiné diagnóze než je PN.

V případech, kde zhodnocení příznaků a odpovídavosti na dopaminergní podnět nestačí k potvrzení či úplného vyloučení PN, se používá nukleárně medicínské vyšetření jednofotonovou emisní termografií (SPECT). Radioaktivní látka se váže na presynaptické dopaminové transportéry. Toto vyšetření potvrdí presynaptický dopaminergní deficit ve striatu, nemůže ovšem odlišit PN od jiných nemocí, u kterých je také postižen presynaptický a postsynaptický oddíl dopaminergního systému. Definitivní diagnózu PN potvrdí až pitevní nález, kde se makroskopicky ukazuje depigmentace v substantia nigra. Postup při diagnostice je naznačen níže v tabulce (Tabulka 1).

PN je nejčastěji příčinou parkinsonského syndromu (PS) a to asi v 80 % případu. PS vzniká při různých chorobných stavech (tzv. sekundární PS) a může být součástí klinického obrazu jiných neurodegenerativních onemocnění (Tabulka 1). U PS bývá test na dopaminergní odpovídavost negativní.

Chyby a omyly v diagnóze PN se dají dělit do dvou kategorií. Na případy, kdy pacient trpí PN je chybně diagnostikován a na případy, kdy pacient trpí jinou chorobou je diagnostikován jako parkinsonik. Nejčastější chybou je pokládat každý třes za projev PN. Je zajímavé, že nejčastějším třesem není PN, ale esenciální třes (ET). Ten je na rozdíl od PN třesem staticko-kinetický (je možné, že i klidová složka může být přítomná), jež je izolovaný a není doprovázen hypokinezí, rigiditou ani jinými typickými příznaky PN. Často bývá i u ET přítomen třes hlavy či hlasivek, který se u PN téměř nevyskytuje. Esenciální tremor je relativně benigní choroba, proto je špatná diagnóza PN pro pacienta zbytečným traumatem a případná zbytečná léčba dopaminem nevylepší třes a navíc může způsobit kolísání krevního tlaku a další nežádoucí účinky.

V některých případech je velmi těžké se chybné diagnóze vyhnout, neboť u sekundárních par-

kinsonských syndromů a i u jiných neurodegenerativních onemocnění může dopaminergní léčba trochu pomoci, ale je chybou komplexně podávat antiparkinsonika, aniž by byl jejich efekt ověřen. Obvykle rozhodne objevení nových symptomů v rozvoji nemoci.

Druhým častým případem je, když pacientovi s PN je přiřazená jiná, chybná diagnóza. Nejčastěji se tak stává u pacientů, kteří onemocní před 40. rokem života (asi 10 % případů). U takto mladých pacientů se na možnost PN vůbec nepředpokládá a proto trvá několik let, než se stanoví adekvátní diagnóza. V našich podmínkách je nejčastější chybná diagnóza arterisklerotického či parkinsonského syndromu. U těchto nemocí se nepředpisuje dopaminergní léčba, ale zcela špatně dostávají vasoaktivní léčbu nebo dokonce cinnarizin, který je pro svůj antidopaminergní účinek u PN přísně kontraindikován. Proto by měl být u všech pacientů s parkinsonským syndromem nasazen L-DOPA po dobu 4-6 týdnů z důvodu ověření jeho neúčinnosti.

### Krok 1: Diagnóza parkinsonského syndromu

- **bradykineze** (zpomalení iniciace volního pohybu a postupné snižování rychlosti a amplitudy pohybu při repetitivních činnostech)
- *a nejméně jeden příznak z následujících:*
  - **svalová rigidita**
  - **klidový třes 4-6 Hz**
  - **posturální instabilita** nezpůsobená primární dysfunkcí zrakovou, vestibulární, mozečkovou nebo proprioceptivní

### Krok 2: Zpochybňující a vylučující kritéria Parkinsonovy nemoci

- opakované úrazy hlavy
- prodělaná encefalitida
- léčba neuroleptiky v době začátku obtíží
- v rodině více než jeden další podobný případ
- výlučně jednostranné postižení po více než 3 letech trvání obtíží
- časná těžká demence s poruchami mnestickými, fatickými a praxickými
- nádor mozku nebo komunikující hydrocefalus na CT mozku
- chybění odpovědi na silné dávky L-DOPA

### Krok 3: Podpůrná prospektivní pozitivní kritéria Parkinsonovy nemoci

- jednostranný začátek
- přítomnost klidového třesu
- progresivní průběh
- přetrvávající asymetrie s těžším postižením na straně začátku
- výtečná odpověď na L-DOPA (90-100 % zlepšení)
- klinické trvání 10 let a více

Tabulka 1: Klinická diagnostická kritéria Parkinsonovy nemoci, UK Parkinson's Disease Society Brain Bank.



### 3 Analýza a návrh řešení

Nyní je čas na probrání způsobů různých druhů vyšetření pro pacienty trpící PN. Nesmíme ovšem zapomínat, že diagnostika nemoci je velmi obtížná, jelikož se hlavní příznaky projevují až po několika letech od počátku nemoci. A proto i samo vyšetření je pouze prvním ukazatelem, které má pomoci k určení dalších kroků.

První fáze nemoci jsou neprokazatelné a proto lékař sleduje vývoj nemoci a na tom stanovuje veškerou diagnózu. Až po příchodu třesu lze lépe poznat zda jde o parkinsonovu nemoc nebo pouze o esenciální třes, či něco jiného. A právě na měření míry třesu se obrací většina vyšetření a aplikací. V následujících podkapitolách se budu věnovat různým aplikacím i skutečnému lékařskému testu na vyšetření třesu.

#### 3.1 Aplikace na mobilní telefony a tablet

Všechny nové mobilní telefony mají mnoho věcí společných a nezáleží na tom, jaký je v nich operační systém (OS). Neustále se vyrábí nové aplikace na přesnější změření polohy prstu na obrazovce nebo citlivější senzory pohybu. V současnosti se prodávají telefony se třemi hlavními druhy OS. Jsou to například Windows, které se vyskytují zejména na mobilech firmy Nokia. Další velkou skupinou jsou mobilní telefony iPhone od firmy Apple, které jsou známé svým user-friendly přístupem a spoustou aplikací a příslušenstvím. Poslední velkou skupinou jsou mobily s OS Android, na které existuje mnoho aplikací. Jelikož jsou volně přístupné všechny příkazy, s kterými tento systém pracuje, využití tohoto OS se stále rozšiřuje.

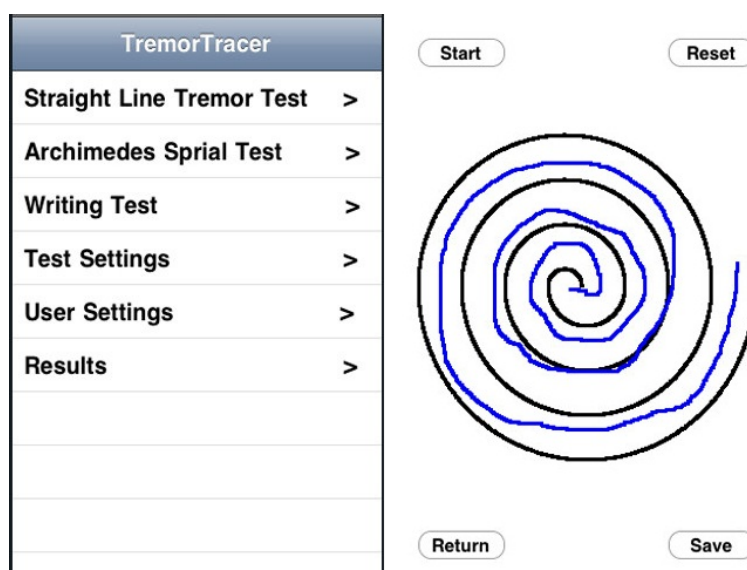
Všechny tyto mobilní telefony mají společné dvě základní věci, kterých můžeme u vyšetření třesu využít. Jedná se o dotykovou obrazovku, na které lze relativně dobře kreslit prsty a o senzory pohybu a natočení (akcelometry a gyroskopy), které neustále počítají natočení a pohyb. Tím se naskýtají dvě možnosti vyšetření.

S využitím dotykového displaye se dá měřit třes horních končetin tak, že pacient má za úkol obkreslit různé tvary zobrazené na obrazovce. Z porovnání předložené šablony a tahu, který pacient vykonal, se určují různé parametry, které ohodnotí kondici pacienta. Jako šablona se nejčastěji vyskytuje spirála, neboť při jejím kreslení je zapojeno nejvíce svalů, takže její nakreslení má největší vypovídající hodnotu. Někdy se ovšem může vyskytnout i test na nakreslení rovné čáry nebo na test opsání zadaného textu.

Druhou možností jest měření třesu pomocí akcelometrů v mobilním telefonu. Pacient je instruován k držení mobilu v ruce v různých polohách, například v natažené ruce, či v pokrčené ruce v pravém úhlu. Mobilu snímá třes pacienta po určitý časový okamžik a následně jej pak vyhodnotí. Tato metoda má výhodu hlavně v tom, že s její pomocí můžeme měřit klidový třes u PN, jehož vypovídající hodnota je lépe změřitelná.

Nyní bych představil dvě aplikace na měření třesu, které obě jsou již v prodeji na iTunes, což je aplikace, nebo lépe řečeno služba, přes kterou si mohou lidé vlastníci iPhone nebo počítač s OS Mac, stahovat hudbu, knihy a aplikace. Tyto aplikace jsou uzpůsobeny právě na iPhone, ale samozřejmě existují podobné i na OS Android.

První z nich se jmenuje TremorTracer [9], je zaměřená na dotykovou obrazovku přístroje. Aplikace má tři druhy testu. Test rovné čáry, test obkreslení spirály a test psaní. Bohužel jsem sám tuto aplikaci nevyzkoušel, proto neznám její přesnější funkčnost a parametry, které měří. Nicméně její náhled je znázorněn níže (Obrázek 2). Je vidět že aplikace má i záložku s výsledky, do kterých zřejmě archivuje jednotlivé testy.



Obrázek 2: Ukázka aplikace TremorTracer [9]

Druhá aplikace naopak využívá gyroskop v mobilním přístroji a měří všechny otřesy a rotace, což je zajímavé, neboť většina podobných aplikací používá pouze akcelometry. Aplikace se jmenuje Tremor-Meter [10] a měří třes a amplitudu horní končetiny. Dále zobrazuje četnost jednotlivých frekvencí od 1 do 16 Hz. Další transparentní a užitečný údaj je pohyb ve všech třech osách, ze kterých se počítá celkový třes a amplituda (graf v dolní části obrazovky). Jednou z dalších výhod je pěkný design aplikace (Obrázek 3), který působí solidně a může mít vliv na upřednostnění tohoto programu před jinými.



Obrázek 3: Ukázka aplikace Tremor-Meter [10]

Jednou ze zajímavých aplikací měřících třes u PN na přístroji tablet ukáží nyní. Jedná se o bezdrátové a bezdotykové měření třesu pomocí kamery tabletu a nasazovacího senzoru. Tato aplikace se jmenuje Kinesia HomeView [11] a jedná se o projekt Great Lakes NeuroTechnologies. Jde o domácí monitorování, kde se pacient přes tablet připojí na stránky této aplikace, přihlásí se na svůj účet a vyplní datum a čas požití levodopy. V průběhu dne dělá různé zadané úkoly, které jsou nahrávány kamerou. Pacient si na prst nasadí senzor snímající jeho motorické funkce v jednotlivých úkolech. Úkoly mohou být různé, například dotknutí se prstem špičky nosu nebo různé cviky s prsty. Data ze senzoru i s nahraným záběrem jsou následně nahrány na server, kde se tvoří tabulka výsledků.

Na webu této aplikace si pacient, ale i jeho lékař může prohlédnout výsledky nebo si přehrát záznam cvičení. Výhodou je denní, a pokud to je zapotřebí, i hodinová kontrola motorických funkcí pacienta, podle které lze optimalizovat medikaci.



Obrázek 4: Ukázka aplikace Kinesia HomeView [11]

### 3.2 Lékařský test Pen&Paper

Výše zmíněná aplikace TremorTracer (Obrázek 2) se hodně inspiruje testem, který se v našem zdravotnictví často využívá. Jedná se o test na papír, kde pacient má za úkol tužkou opsat jednu zadanou větu (Obrázek 5). Tím se zjišťuje úhlednost písma, ale i mentální stav pacienta. Další úkol je napsat správné datum. Následuje úkol obtížnější a to spojit dva body, kde první je uvnitř spirály a druhý na jejím konci. Opět je spirála zvolena proto, aby pacient musel namáhat co nejvíce svalů a třes se mohl prosadit. Poslední úkol je spojit dva body mezi dvěma linkami, což je test rovné čáry. Tyto úkoly se provádějí levou rukou i pravou. Je důležité vědět, jaká ruka je pro pacienta dominantní.

Další částí testu je procvičení prostorové orientace, kde se pacient musí dotknout prstem nosu a to jak levou rukou, tak i pravou. Podobné cvičení se dělá i pro dolní končetiny, kde se pacient musí dotknout patou jedné nohy kolena druhé. Lékař u každé nohy a ruky vyhodnotí míru třesu pohledem a zapíše do připravených kolonek.

Poslední částí testu jest zkouška se šálky s tekutinou, kdy pacient musí přelít vodu z jednoho šálku do druhého. Tímto celý test končí a doktor vyhodnocuje závěry. V průběhu nemoci se tyto testy případně opakují a archivují, tím se zmapuje vývoj nemoci.

**VYŠETŘENÍ TŘESU:**

Jméno pacienta: \_\_\_\_\_ Datum: \_\_\_\_\_

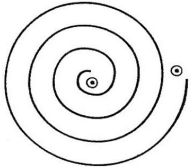
Diagnóza: \_\_\_\_\_ Pravák ☐ Levák ☐

**Pravá ruka:**


Napište: »Toto je ukázka mého nejlepšího písma« \_\_\_\_\_

Podepište se a napište dnešní datum: \_\_\_\_\_

Nakreslete spirálu spojením bodů mezi vyznačenou čarou (ruka není položena na papíře):



Nakreslete linku mezi vyznačenými body (ruka není položena na papíře):

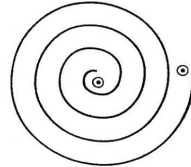


**Levá ruka:**


Napište: »Toto je ukázka mého nejlepšího písma« \_\_\_\_\_

Podepište se a napište dnešní datum: \_\_\_\_\_

Nakreslete spirálu spojením bodů mezi vyznačenou čarou (ruka není položena na papíře):



Nakreslete linku mezi vyznačenými body (ruka není položena na papíře):



**VYŠETŘENÍ TŘESU:**

**Zkouška prst-nos pro HK, pata-koleno pro DK**

0. žádný třes

1. nepatrný, stěží postřehnutelný, může být intermitentní

2. mírný, amplituda do 2 cm, může být intermitentní

3. zřetelný, amplituda 2-4cm

4. velmi výrazný, amplituda nad 4 cm

☐ PHK ☐ LHK ☐ PDK ☐ LDK

**Přelévání: použít šálky naplněné vodou 1 cm od horního okraje**

0. normální

1. mnohem opatrnější než osoba bez třesu, žádná voda nerozlita

2. rozlito malé množství vody (do 10% celkového množství)

3. rozlito zřetelné množství vody (10-50%)

4. neschopný přelévát, aniž by rozlil většinu vody

☐ PHK ☐ LHK

Neurologická klinika 1.LF UK a VFN, Kateřinská 32, 128 00, Praha 2, Flektnost: Prof. MUDr. Eduard Štěpánek, DrSc.

Extrapyramidové centrum, Doc. MUDr. Jan Roth, CSc. Tel.: 224 965 539, Fax: 224 965 078

Upřesnění školicího testu pro hodnocení třesu: Fahn, S., Tolosa, E., Marra, C. Clinical rating scale for tremor.

In Jankovic, J., Tolosa, E. (Eds.), Parkinson's disease and Movement Disorders. Baltimore: Williams and Wilkins, 1993, p. 273-280.

Neurologická klinika 1.LF UK a VFN, Kateřinská 32, 128 00, Praha 2, Flektnost: Prof. MUDr. Eduard Štěpánek, DrSc.

Extrapyramidové centrum, Doc. MUDr. Jan Roth, CSc. Tel.: 224 965 539, Fax: 224 965 078

Upřesnění školicího testu pro hodnocení třesu: Fahn, S., Tolosa, E., Marra, C. Clinical rating scale for tremor.

In Jankovic, J., Tolosa, E. (Eds.), Parkinson's disease and Movement Disorders. Baltimore: Williams and Wilkins, 1993, p. 273-280.

Obrázek 5: Ukázka testu tužka-papír

### 3.3 Návrh řešení

Nyní, když jsem probral již existující aplikace na vyšetření třesu i lékařskou variantu s tužkou a papírem, začnu se zabírat metodou, kterou budu vyvíjet a stanovovat parametry, jež jsou potřeba měřit.

Prvním a základním krokem je určit, co vlastně chceme u pacienta měřit. Parkinson se nejvíce a nejprokazatelněji projevuje právě třesem horních končetin a na to se také zaměřuje má bakalářská práce. Rozhodl jsem se použít dotykovou obrazovku přístroje pro obkreslení spirály. Jak jsem již mnohokrát zmiňoval, spirála je nejvhodnější právě proto, že se při jejím kreslení zapojí nejvíce svalů. Jako snímací způsob sběru dat jsem upřednostnil tablet před mobilním telefonem a to z důvodů větší obrazovky. Starší lidé si s moderní technikou příliš nerozumí a malý display mobilního telefonu by mohl značně zkreslit výslednou vypovídací hodnotu měření. A proto pro tyto lidi bude příjemnější používat a kreslit na větší obrazovku tabletu.

Na rozdíl od lékařského spirálového testu, kdy má pacient za úkol spojit dva body na začátku

a konci spirály, jsem zvolil raději šablonu spirály, kterou má pacient obkreslit, přičemž je každá výchylka zaznamenána a vyhodnocena.

Rozeznáváním chybných bodů a dobrých bodů v celkovém tahu dává možnost stanovení procentuální úspěšnosti, což je první důležitý parametr, který můj program stanovuje. Po dokončení tahu se pacient i lékař hned dozví, jak přesný a úspěšný pokus byl. Dalším důležitým údajem je maximální chyba celkového tahu, která stanovuje a aktualizuje maximální odchylku od spirály při tahu. Známe-li již maximální chybu, je dobré znát i chybu průměrnou, což tento program také dělá. Důležitým údajem je i čas potřebný k obtažení spirály. Tento parametr je důležitý k vyhodnocení celkového výsledku. Čím pomaleji člověk táhne, tím méně chyb udělá a tím lepší má úspěšnost. Posledním měřeným parametrem je stanovení třesu, který musíme ovšem brát v úvahu pouze někdy, jelikož při málo chybách je tento údaj zcela nekompetentní.

Aby byl program funkční, je důležité mít možnost program vypnout, po jednom tahu vymazat, popřípadě měření opakovat od začátku. Důležité je také výsledné tahy ukládat a ani to v programu nesmí nechybět. Výsledky se ukládají na SD kartu do složky Parkinson ve formátu JPG. Všechny tyto možnosti budou v menu programu.

Tímto jsem shrnul všechny dostupné informace, které se týkají nemoci, projevů a diagnóz i již existujících aplikací na vyšetření třesu. Teď je čas se přesunout k samotné tvorbě programu.

## 4 Realizace

V této kapitole popíši postup vývoje aplikace, problémům, které se při tvorbě vyskytly a základní principy a ukázky kódu. Začnu ovšem s přístrojem, na kterém jsem svoji práci vytvářel a s operačním systémem Android.

### 4.1 Android a Asus EEE Pad Slider SL101

K vývoji aplikace na změření třesu je jedním z vhodných přístrojů právě platforma tablet, je samozřejmě možné podobnou aplikaci vyvíjet pro mobilní telefony, ale jak jsem již zmiňoval, větší dotyková obrazovka je lepší a komfortnější pro pacienta. Za tímto účelem mi byl školou zapůjčen tablet Asus EEE Pad Slider SL101 (Obrázek 6), který se od běžných tabletů liší klávesnicí, která je skrytá pod displayem a lze ji vyklopit. Tato klávesnice je pro mou aplikaci naprosto nepotřebná, ale pro většinu lidí je tato forma tabletu více vyhovující.



Obrázek 6: Asus EEE Pad Slider SL101

Tablet je dále vybaven desetipalcovou dotykovou LED obrazovkou, která je chráněna tvrzeným sklem. Dosažitelné rozlišení je 1280x800 a obraz se automaticky přetáčí podle potřeby. Dále je přístroj vybaven dvěma kamerami na přední a zadní straně, WiFi přijímačem, USB, mini

HDMI rozhraním a čtečkou micro SD karet. O výkon přístroje se stará procesor NVIDIA Tegra 2 pracující na frekvenci 1 GHz spolu s operační pamětí 1 GB. Na tabletu je operační systém Google Android 3.2.

Android je operační systém vytvořený zejména pro inteligentní mobilní telefony. Operační systém je založený na jádru Linux. Jeho historie začíná v roce 2003 když byla společnost Android Inc. založena Andym Rubinem, Richem Minerem, Nickem Searsem a Chrisem Whitem. V roce 2005 odkoupil tuto společnost Google Inc. a udělal jí svou dceřinou společností. Zakladatelský tým pod vedením Googlu, v čele s Andym Rubinem, vyvinul platformu na Linuxovém jádře a v roce 2007 získali několik partnerů z oblastí mobilních telefonů [12].

5. listopadu roku 2007 bylo založeno uskupení Open Handset Alliance. Konsorcium velkých i malých výrobců mobilních telefonů (např. Google, HTC, Intel, LG, Motorola, NVidia, Qualcomm, Samsung, Texas Instruments a dalších 25 společností), kteří se dohodli na vývinu otevřeného standardu pro výrobu mobilních zařízení. O týden později bylo vydáno první Android SDK pro vývojáře pod licencí open-source [13].

Architektura OS Android se dělí na pět základních částí, které se vzájemně protínají. Nejnížší vrstvou je jádro operačního systému, odděluje používání hardwaru a softwaru. Jelikož je tento OS vybudován na jádru Linux, je zde využito jeho mnoha vlastností, jako například podpora správy paměti, správa sítí, zabudované ovladače nebo správy procesů. Další vrstvou jsou knihovny, které jsou poskytnuty prostřednictvím Android Application Framework a jsou psány v C/C++ kódu. Třetí vrstvu tvoří Android Runtime, který obsahuje virtuální stroj, jenž je potřeba pro práci s pamětí a vlákny. Tato vrstva také umožňuje používat příkazy z programovacího jazyka Java. Nejdůležitější vrstvou pro vývojáře je ovšem vrstva Application framework, která poskytuje přístup ke službám používaných přímo v programech. K těmto prvkům patří zejména sada prvků View (seznamy, textové pole, tlačítka, checkboxy), Content providers (přístup k obsahům), Resource manager (řetězce a grafika), Notification manager (povoluje všem aplikacím zobrazit vlastní upozornění ve stavovém řádku), Activity manager (řídí "život" aplikací a pořadí v zásobníku). Nejvyšší vrstvou systému jsou základní aplikace, které jsou využívány běžnými uživateli [14].

Aplikací, která nám zpřístupňuje funkce a příkazy pro programování na přístroje s OS Android se nazývá Android Software Development Kit (SDK). Je přístupný pro všechny druhy počítačů (Windows, Linux i Mac). Přes tuto aplikaci je možné si stáhnout příkazy a dokumentaci na určitou verzi Androidu a také USB ovladače pro komunikaci s přístrojem. Jedním z potřebných nástrojů je i SDK tools, který zpřístupňuje debuggování, analýzu grafického layoutu a další potřebné programy. Neméně důležitou částí, kterou nám SDK zpřístupňuje je Emulátor, virtuální zařízení,



pomocí kterého můžeme testovat vytvořené aplikace bez fyzického zařízení.

## 4.2 Vývoj programu

Prvním potřebným krokem k vývoji mé aplikace bylo zprovoznění vývojového prostředí tak, abych mohl programovat na zařízení se systémem Android. V tomto účelu je nejlepší vývojová platforma Eclipse, která je oficiálně podporovanou platformou pro práci s Android projekty. To umožňuje instalaci SDK.

Po stažení a implementaci SDK do vývojové platformy Eclipse jsem musel nainstalovat příslušnou verzi Android příkazů pro tablet, spolu s tím jsem zprovoznil i komunikaci mezi stolním počítačem, na kterém jsem programoval a tabletem, na kterém jsem si aplikaci zkoušel.

Správnost nainstalování jsem si ověřil prostřednictvím jednoduchého programu „Hello World“, který vypíše na obrazovku nějaký text. Tento program jsem spustil na virtuálním emulátoru i na přístroji samotném. K tomu, aby se program spouštěl přes počítač na tablet je nutné nastavit jak na přístroji, tak i v programu *Debuggable* na *True*. V zařízení se tato úloha nastavuje v hlavním menu přístroje a v programu se musí nastavit tento parametr v souboru *AndroidManifest.xml*. Tento soubor musí mít každá aplikace psaná pro Android a musí být umístěna v kořenovém adresáři všech aplikací. Obsahuje informace o programu potřebné ještě před spuštěním programu.

Když již mám fungující vývojovou platformu a funkční komunikaci mezi tabletem a počítačem, mohu se dát do práce s tvorbou mé aplikace.

### 4.2.1 Vizualizace

Jádrem mé práce je vizualizace, neboť právě snímáním pacientova prstu a čarou, kterou jím kreslí, se bude vyhodnocovat třes. Takže prvním krokem bylo zprovoznit funkci kreslení a tu pak modifikovat na formu, kterou potřebuji.

Po spuštění programu se jako první věc vyhodnocují příkazy, které jsou ve funkci *onCreate* (Ukázka kódu 1).

```
public void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(new MyView(this)); // Zavola tridu MyView  
    setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_NOSENSOR);  
}
```

Ukázka kódu 1: Základní třída *onCreate*

Tato funkce odkazuje na třídu *MyView*, ve které se řeší všechny operace kreslení. Jedním z prvních problémů, které jsem u kreslení musel řešit, bylo vymazání nakreslených tvarů po přetočení obrazovky. Řešením bylo zablokování přetáčení obrazu v aplikaci. Příkaz *setRequestedOrientation* ovládá automatické přetáčení. Jelikož úkolem bude obkreslit spirálu, nevadí mi, když se nebude přepínat orientace podle natočení tabletu.

Třída *MyView* obsahuje mnoho funkcí potřebných ke kreslení. První z důležitých funkcí je *onDraw* (Ukázka kódu 1), která se volá stále dokola po dokončeném tahu, ale i při spuštění. Deklaruje se zde styl a barvy kreslených objektů, ale také barva pozadí. Stanovuje se pořadí objektů co a čím se bude překreslovat.

```
public void onDraw(Canvas canvas) {
    myCanvas = new Canvas(myBitmap); // Stanovi, ze objekty kreslene
                                     // pres myCanvas kresli primo do
                                     // myBitmap
    if (helpCounter == 0) { // Hlidani pro vykresleni spiraly, pouze na
                           // pocatku
        spiral(canvas); //Vola funkci pro vypsani spiraly
        helpCounter = 1;
    }
    canvas.drawBitmap(myBitmap, 0, 0, myPaint); // Nastaveni pro
                                                // zobrazeni a zapsani
                                                // bitmapy
    canvas.drawPath(myPath, myPaint);
    invalidate(); // Vykonani a zobrazeni vseh zmen (aktualizace) Nutne
                 // pro spravnou funkci Erase
}
```

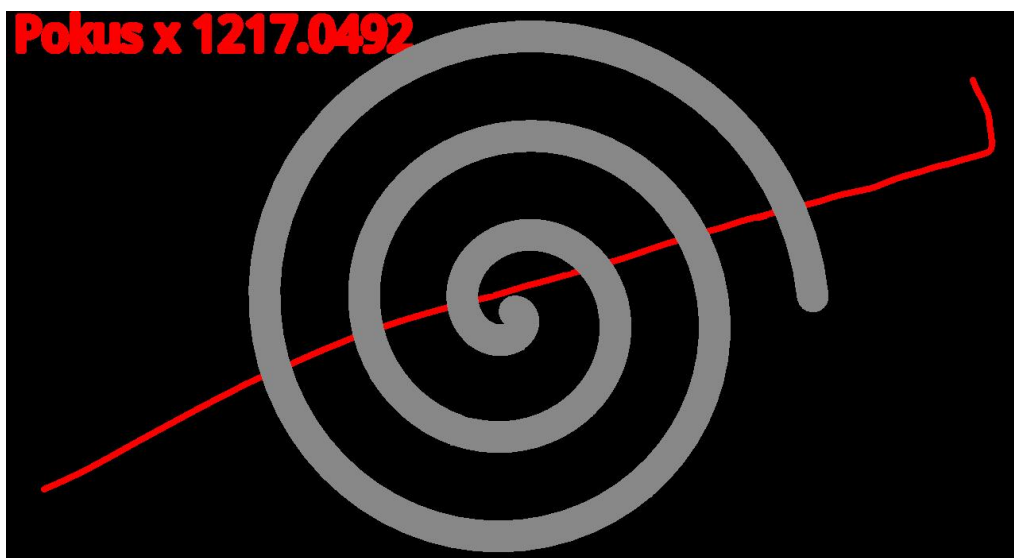
#### Ukázka kódu 2: Funkce onDraw

V originálním kódu je i nastavení barev, velikostí pro šablonu spirály a kreslení prstem, ale jelikož se jedná o mnoho řádků, pro lepší přehlednost jsem tyto ukázky smazal. V tomto úseku kódu se deklaruje zapisování do bitmapy, což si můžeme představit jako souřadnicové pole všech pixelů obrazu. V pozdější fázi programu ukládám bitmapu jako obrázek a proto je důležité zapisování přímo do bitmapy a ne jen překládání několika objektů přes sebe.

Další nezbytnou částí je volání funkce pro vykreslení spirály jakožto šablony, kterou má být za úkol obkreslit. Tato funkce je podmíněna pouze pro první průchod programem, takže se zavolá pouze po spuštění programu. Jelikož se funkce *onDraw* volá stále dokola i spirála by se kreslila opakovaně a to by mělo za důsledek překrývání tahu, který pacient vykonal. Na obrázku číslo 7,

který jsem pořizoval při vývoji programu, je vidět, jak se tah uživatele (červená čára) překrývá šablonou spirály (šedá čára). Tento obrázek zachycuje chvíli, kdy mi ještě nefungovalo správně zapisování do bitmapy, kterou jsem následně ukládal jako fotku. Tento problém jsem zkoušel vyřešit různými způsoby, ale nakonec jsem zvolil ten nejjednodušší, kde se porovnává proměnná *helpCounter* - jestli se rovná nule, pak se vykreslí spirála. Po vykreslení spirály se tato proměnná nastaví na jedničku a při dalších průchodech se kreslení spirály přeskočí.

Následující příkazy *drawBitmap* a *drawPath* jsou důležité pro kreslení na bitmapu, aby se tahy pacienta překryly přes barvu pozadí a šablonu. Poslední příkaz *invalidate()* vykonává a překresluje celou bitmapu podle příkazů, které se ještě nevykonávaly. Tento příkaz je nutný pro funkci *Erase*, která smaže všechny tahy. Bez tohoto příkazu bylo nutné kliknout na obrazovku, až pak se obrazovka vyčistila.



Obrázek 7: Příklad špatného překrývání tahu spirálou

Kreslení samotné je odkázáno na dotyku prstu na obrazovku. Tento fakt vyvolá událost (*event*), kterou program zachytí a dále se ji snaží vyhodnotit. Nyní ukáži kousek kódu, který zachytí a vyhodnotí dotyk a dále odkazuje na kreslení samotných čar (Ukázka kódu 3).

```
public boolean onTouchEvent(MotionEvent event) {  
    float x = event.getX();  
    float y = event.getY();  
    switch (event.getAction()) {  
    case MotionEvent.ACTION_DOWN:  
        touch_start(x, y);  
        invalidate();  
    }
```

```

        break;
    case MotionEvent.ACTION_MOVE:
        touch_move(x, y);
        invalidate();
        break;
    case MotionEvent.ACTION_UP:
        touch_up();
        invalidate();
        break;
    }
    return true;
}

```

### Ukázka kódu 3: Funkce onTouchEvent

Při každé události se zapamatují souřadnice  $x$  a  $y$  a podle toho, zda jde o první dotyk, probíhající tah nebo poslední dotyk dokončeného tahu, funkce zavolá příslušnou funkci pro kreslení. Po vykonání jedné z funkcí se program opět vrátí zpět do tohoto vyhodnocení, kde se příkazem *invalidate()* opět překreslí celá bitmapa.

Jak již bylo řečeno, jsou možné tři akce při zjištění dotyku. Pro první dotyk se do lokálních proměnných zapíše souřadnice dotyku a nastaví se počáteční poloha pro připravovaný tah. Při dalším pohybu se vypočítává vzdálenost mezi jednotlivými body. Po překročí tolerance, která se v kódu nastavuje, se příslušná část dráhy vykreslí. Při posledním dotyku na konci tahu se pouze dokreslí poslední část čáry a poloha se vymaže, tím se připraví na další tah.

Ukázky kódu pro všechny tři funkce neukáží, neboť jsou velmi obdobné, ale už pouze sama skutečnost rozdělení tahu do tří částí je velmi důležitá pro další tvorbu programu. Více o tom však bude řečeno později (Kapitola 4.2.3 Vyhodnocení bodů). Pro přehled ovšem ukáží a osvětlím alespoň část pro vykreslení čáry uprostřed tahu (Ukázka kódu 4).

```

private void touch_move(float x, float y) {
    float dx = Math.abs(x - mX);
    float dy = Math.abs(y - mY);
    if (dx >= TOUCHTOLERANCE || dy >= TOUCHTOLERANCE) {
        myPath.quadTo(mX, mY, (x + mX) / 2, (y + mY) / 2);
        mX = x;
        mY = y;
        Point send = new Point(mX, mY); // Vytvoreni objektu bod pro

```

```

// odeslání na vyhodnocení
vzdalenost(send); // Funkce pro zjištění vlastnosti bodu
}
}

```

Ukázka kódu 4: Funkce `touch_move` řídící kresbu při tahu

Z předešlého bodu a bodu nového se vypočítá vzdálenost, která se následně porovnává s nastavenou tolerancí, pokud je odchylka vyšší, nakreslí se příslušná část čáry. Pak se přepíše lokální proměnné na novou hodnotu a tento nový bod se pošle na vyhodnocení, viz kapitola 4.2.3 Vyhodnocení tahu.

#### 4.2.2 Spirála a body

Další část programu je vykreslení šablony spirály při spuštění. To je možné pouze po prvním spuštění aplikace kvůli překrývání tahu a šablony. To jsem již vysvětloval výše u probírání funkce *onDraw*. Při vykreslení šablony jsou důležité parametry obrazovky. Je žádoucí mít střed spirály uprostřed displaye a velikost spirály nesmí být příliš velká, aby nevyčnívala přes obrazovku.

Nyní ukáži celou funkci vykreslení spirály a vysvětlím jednotlivé příkazy (Ukázka kódu 5).

```

public void spiral(Canvas canvas) {
    canvas = new Canvas(myBitmap); // canvas zapisuje do myBitmap
    canvas.drawColor(0xFFAAAAAA); // Nastavení barvy pozadí
    int a = 0, pom = 0; // Pomocné parametry
    float x = 0, y = 0;
    float startx = (float) myBitmap.getWidth() / 2;
    float starty = (float) myBitmap.getHeight() / 2 + 30;
    float xold = startx, yold = starty; // nutnost pro kreslení spirály
    if (myBitmap.getHeight() > myBitmap.getWidth()) {
        a = myBitmap.getWidth() / 35;
    } else {
        a = myBitmap.getHeight() / 35;
    }
    for (double i = 0; i < 18.9; i += 0.1) {
        x = (float) (startx + a * Math.cos(i) * i); // Spirála X
        y = (float) (starty + a * Math.sin(i) * i); // Spirála Y
        canvas.drawLine(xold, yold, x, y, sablona); // Čára
        points[pom] = new Point(xold, yold); // Zapamatování
    }
}

```

```

pom++; // Dodržování pořadí v poli
xold = x; // Nastavení nových "starých" bodu
yold = y;
}
points[189] = new Point(xold, yold); // Dodání posledního bodu
}

```

#### Ukázka kódu 5: Funkce kreslící spirálu

Z počátku funkce je důležité stanovit opět správné vpisování do bitmapy. Druhý příkaz `canvas.drawColor(0xFFAAAAAA)` zbarví pozadí celé bitmapy na bílo. Dále je nutné deklarovat všechny proměnné pro výpočet jednotlivých souřadnic spirály. Proměnná *pom* je využita pro hlídání pořadí pro pole jednotlivých bodů spirály. Proměnné *startx* a *starty* určují počátek spirály a jsou středy maximální šířky a výšky bitmapy. Výška je posunutá o 35 pixelů směrem dolů, neboť vykreslená šroubovice je na tabletu trochu větší směrem vzhůru. *Xold* a *yold* jsou koncové souřadnice pro kreslení čáry s novými body. Zpočátku jsou nastaveny na *startx* a *starty*. Parametr *a* stanovuje velikost spirály, pro její určení je však potřeba znát rozložení obrazovky, neboť jsou rozdíly mezi displayi tabletu a telefonu. Tablet má větší šířku než výšku, mobil naopak. Proto se parametr *a* nastavuje v závislosti na tom, která souřadnice je menší tak, aby byla na obrazovce vidět celá spirála.

Nyní se dostáváme k samotnému cyklu vytvářejícímu spirálu. Tento cyklus se řídí proměnnou *i*, která nabývá postupně hodnoty od nuly do 18,9 a inkrementuje se po jedné desetinné při průchodu. Toto číslo se v maximu skoro rovná  $6\pi$  a určuje počet otáček ve spirále. Jednotlivé body spirály se pak tvoří narůstající vzdáleností od středu spirály za pomoci funkcí sinus a cosinus. Pro pravotočivou spirálu pak platí následující vzorce.

$$x = startx + a \cdot i \cdot \cos(i) \quad (1)$$

$$y = starty + a \cdot i \cdot \sin(i) \quad (2)$$

Se zvyšujícím se *i* se jednotlivé body vzdalují svojí absolutní vzdáleností od středu spirály a díky kombinaci sinu a cosinu se tyto body mění v kruh. Tímto se tvoří nové body spirály a ze znalosti bodů předešlých můžeme spirálu postupně vykreslovat následujícím příkazem `canvas.drawLine(xold, yold, x, y, sablona)`. Poslední parametr v tomto příkazu je deklarace třída *Paint*, která stanovuje vlastnosti čáry jako je barva, tloušťka a podobně. Tyto vlastnosti jsou již deklarovány ve funkci *onDraw*.

Pro vyhodnocení pacientova tahu je důležité pamatovat si jednotlivé body spirály. Proto jsem vytvořil objekt *Point*, který nese informaci o souřadnicích *x* a *y* jednotlivých bodů. Tyto body

spirály se ukládají do pole *points*[ ]. Nejdříve jsem se pokusil vytvořit *ArrayList*, ale při nahrání programu do tabletu se program přerušoval a hlásil chybu. Tuto chybu se mi nepodařilo opravit a proto jsem zvolil uchovávání bodů v poli, s kterým přístroj neměl problémy. Na konci každého cyklu se aktualizují nové „staré“ souřadnice pro nakreslení dalšího kousku čáry. Po skončení cyklu je nutné ještě přiřadit poslední objekt do pole.

Po průběhu této funkce je na bitmapu nakreslena šedivá spirála na bílém pozadí(Obrázek 8) a je možné začít testování na jejím obkreslení. Jelikož mám uložené všechny body spirály a dokážu rozpoznat jednotlivé body kreslené pacientem, jsem schopen tyto tahy i vyhodnotit.



Obrázek 8: Šablona spirály

#### 4.2.3 Vyhodnocení bodů

Když jsem program tvořil, vytvářel jsem měřené parametry postupně a tak se postupně modifikoval i samotný program a pomocné proměnné. Prvním krokem pro vyhodnocení bylo připravení všech bodů ze spirály. To mám však zmíněno již v minulé kapitole a ukládají se přímo při tvoření spirály.

V prvním kroku je nutné určit, kdy je nakreslený bod uvnitř a kdy vně spirály. Proto je každý bod, který se zakresluje poslán do funkce *vzdálenost()* (Ukázka kódu 6), kde se bod vyhodnocuje a kde se střádají informace o chybách. Následná ukázka kódu je celá funkce *vzdálenost()*. Nyní ji ukáži a následně vysvětlím principy rozhodování při běhu programu.

```
public void vzdálenost(Point point) {  
    float abs = 0; // Deklarace pomocných promenných pro výpočty  
    boolean llast;  
    min = 300000;
```

```

llast = last; // llast je predchozi vysledek a last je novy
for (int i = 0; i < points.length; i++) {
    abs = (float) Math.sqrt((point.getX() - points[i].getX())
        * (point.getX() - points[i].getX())
        + (point.getY() - points[i].getY())
        * (point.getY() - points[i].getY()));
    if (abs < 22) {
        min = 0;
        last = true;
        break;
    } else if (min > abs) {
        min = abs;
        last = false;
    }
}

if (last == false && llast == true) {
    startOfTremor = timer.getElapsedTime();
} else if (last == true && llast == false) {
    endOfTremor = timer.getElapsedTime();
    tremorCounter++;
    avrTremor += endOfTremor - startOfTremor;
}

if (absMax < min) {
    absMax = min;
}

count++;
if (last == false) {
    prumer = prumer + min - 22;
    bad++;
} else if (last == true) {
    good++;
}
}

```

Ukázka kódu 6: Hodnotící funkce vzdalenost



Na začátku funkce se deklarují pomocné proměnné, konkrétně je proměnná *abs* použita při výpočtu absolutní vzdálenosti nově započteného bodu od všech bodů ze spirály, je typu *float*, neboť i souřadnice vyhodnocení z události doteku jsou tohoto typu. Proměnná *last* v sobě nese informaci o nově nakresleném bodu a pokud je *true*, je bod uvnitř spirály a pokud *false*, je vně spirály a hodnotí se jako chyba. Proměnná *llast* je bod, který předchází novému bodu. Proměnná *min* je pokaždé nastavena na velké číslo a je potřeba pro hledání správné chyby ve vyhodnocování, ale o principu tohoto rozhodování napíši o chvíli později.

Po deklarování proměnných je první důležitý příkaz aktualizování předchozího bodu, zda byl dobrý nebo špatný. Je to potřeba kvůli zjištění přechodu ze stavu *bad* do stavu *good* a následně se z toho vyhodnocuje míra třesu.

Každý nově příchozí bod je vyhodnocován cyklem, ve kterém se počítá jeho absolutní vzdálenost od všech bodu spirály, která jsou uchovány jednorozměrným polem *points*[ ].

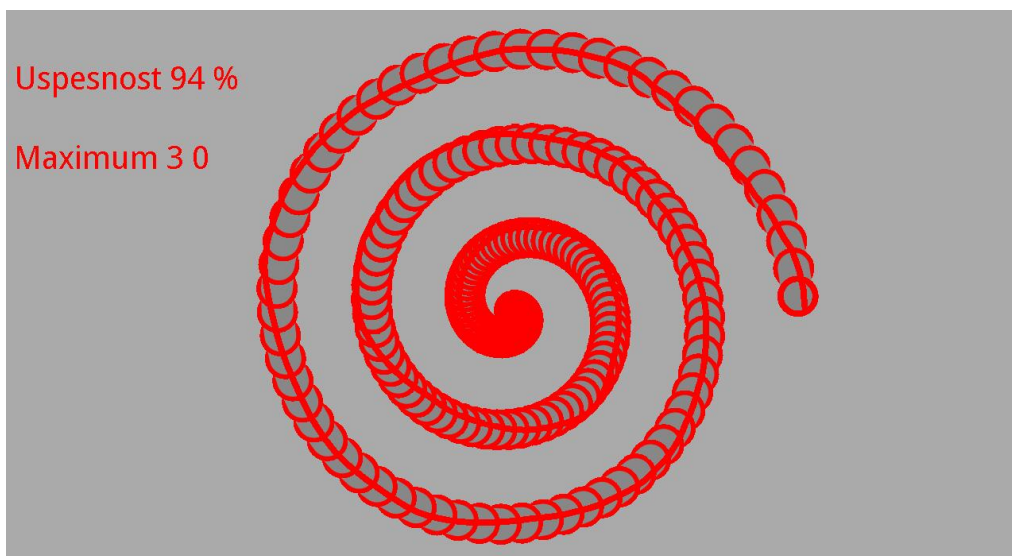
$$abs = \sqrt{(x - points[i].getX())^2 + (y - points[i].getY())^2} \quad (3)$$

Tato vzdálenost musí být porovnána s velikostí odpovídající šířce šablonové spirály. Tu jsem zjišťoval postupným vykreslováním kruhů v každém z bodů spirály a změnou poloměru těchto kruhů (Obrázek 9). Tímto způsobem jsem zjistil poloměr o velikosti 22 a oproti tomuto číslu se porovnávala vypočítaná absolutní velikost.

Pokud zjištěná absolutní hodnota je menší než 22, pak se nově nakreslený bod nachází uvnitř spirály (alespoň pro nějaký bod spirály) a je vyhodnocen jako bod správný (*last = true*) a chybná vzdálenost je 0, pak je celý cyklus přerušen příkazem *break* a vyhodnocování pokračuje.

Jestliže je absolutní hodnota větší než 22, jedná se o chybu, ale jelikož procházíme celý cyklus, mění se nám absolutní velikosti vůči všem bodům a nás zajímá vzdálenost od nejbližšího z bodů spirály. Proto je ještě před začátkem cyklu nastavena pomocná proměnná *min* na velké kladné číslo a každá absolutní vzdálenost je porovnána z tímto velkým číslem. Pokud je *abs < min*, proměnná *min* se nastaví na hodnotu *abs* a v dalších cyklech se vzdálenosti jednotlivých bodů porovnávají s touto hodnotou. Samozřejmě se nastaví i proměnná *last* na hodnotu *false*, což nám říká, že je bod chybný. Za předpokladu chybného bodu se projdou všechny body spirály a jejich vzdálenosti od nakresleného, v proměnné *min* je nejmenší chybná vzdálenost k některému z bodů spirály a *last* je *false*.

Ted' se dostáváme do bodu, kdy je cyklus dokončen a my víme, zda je bod správný, či chybný a je-li chybný, znám jeho chybnou vzdálenost. Nyní je třeba vyřešit otázku třesu a chyb. Abych mohl stanovit průměrnou míru třesu, potřebuji zjistit průměrnou dobu stavu, kdy se body nachází vně spirály. Za tímto účelem jsem použil třídu *StopWatch()*, která odečítá systémový čas zařízení. Tato třída má možnost začít měření času, skončit odměřování, ale i zjistit čas při probíhající



Obrázek 9: Vyhodnocení vnitřku spirály

měření.

Jeden z měřených parametrů je celkový čas vykonání tahu a proto při přiložení prstu na obrazovku se přes událost *onTouchEvent* zavolá funkce *touch\_start* ve které se použije odměřování času příkazem *timer.start()*. Po dokončení tahu se opět přes *onTouchEvent* zavolá funkce *touch\_up*, kde se měření času vypne příkazem *timer.stop()*. Zmiňuji to nejen pro měření parametru celkového času tahu, ale je toho využito i pro stanovení třesu.

Po skončení cyklu známe vlastnosti nového bodu, ale i bodu, který tomuto bodu předcházela. Z této znalosti jsem schopen rozpoznat, kdy jde tah ze stavu *good* do stavu *bad* a tím se dostáváme do chybné části tahu. Do proměnné *startOfTremor* se nahraje čas vzniku této chyby. Následuje sequence bodů, které jsou všechny špatné a při změně z *bad* do *good* a opět odečte čas přechodu do proměnné *endOfTremor*. Rozdílem začátku a konce chyby dostanu čas, který pacientův tah strávil vně spirály. Tento rozdíl se sčítá s ostatními chybnými časy do lokální proměnné *avrTremor* a do jiné proměnné *tremorCounter* se inkrementuje počet všech chybných tahů. Vydělením *avrTremor* ku *tremorCounter* dostanu průměrný čas strávený vně spirály, ale toto vyhodnocení je zpracováno až ve funkci *evaluate()*, kterou popíši později (Kapitola 4.2.4).

Dalším z měřených parametrů je maximální chyba, kterou pacient v tahu udělal. Tu mi zachytí podmínka, kdy se porovnává chyba vzniklá novým bodem s lokální proměnnou *absMax*, do které je vždy uložena maximální chyba tahů předchozích. Při chybě větší než té, která bylo doposud zjištěna, se proměnná *absMax* aktualizuje.

Poslední částí funkce je incrementace jednotlivých bodů do lokálních proměnných. Pokud je nový bod deklarován jako správný, inkrementuje se proměnná *good*, jestli-že je bod chybný, pak se inkrementuje proměnná *bad*. Ať byl bod dobrý nebo špatný, započítává se do proměnné *count*.

Jelikož mne zajímá i průměrná chyba, tak za podmínky chybného bodu se počítá průměr všech chyb, respektive se všechny zjištěné chyby sčítají dohromady a až při vyhodnocování se dopočítá průměr. Od každé nové chyby z proměnné *min* se musí ještě odečíst šířka šablony o velikosti 22, tím je tato chyba měřena od jejího okraje.

Výsledkem této funkce je ohodnocení každého bodu, který pacient vykonal a připravení všech pomocných proměnných na konečné zpracování celkového tahu. Tím se dostávám do poslední fáze k celkovému vyhodnocení tahu.

#### 4.2.4 Vyhodnocení celého tahu

Finální vyhodnocení tahu je uděláno ve funkci *evaluate()* (Ukázka kódu 7), který shrnuje výsledky z funkce *vzdalenost()*. Obsahuje některé finální dopočítání průměrných hodnot, vypisování a aktualizaci výsledku. Tato funkce je zavolána ve chvíli, kdy pacient dokončí tah a pokud pacient obkresluje spirálu postupně nebo když na chvíli tah přeruší, toto vyhodnocení se sčítá s předchozím. Dále následuje ukázka kódu a samozřejmě i jeho vysvětlení.

```
public void evaluate() {
    double prumerChyba = 0;
    DecimalFormat df = new DecimalFormat("#.###");
    double tremor = 0;

    double MAX = absMax - 22;
    if (MAX < 0) {
        MAX = 0;
    }

    sablona.setColor(0xFFAAAAAA);
    sablona.setStyle(Style.FILL);
    myCanvas.drawRect(10, 70, 320, 110, sablona);
    myCanvas.drawRect(10, 205, 300, 170, sablona);
    myCanvas.drawRect(10, 305, 300, 270, sablona);
    myCanvas.drawRect(10, 405, 200, 270, sablona);
    myCanvas.drawRect(10, 505, 300, 270, sablona);

    if (bad > 0) {
        prumerChyba = prumer / bad;
    }
```

```

score = (int) 100 * good / count;
if ((double) timer.getElapsedTime() / 1000 >= 0.001) {
    pomTime += (double) timer.getElapsedTime() / 1000;
}
tremor = 1 / (avrTremor / 1000 / tremorCounter);

myPaint.setStrokeWidth(2); // nastaveni pro vypis vysledku
myPaint.setStyle(Paint.Style.FILL);

myCanvas.drawText("Uspesnost " + score + " %", 10, 100, myPaint);
myCanvas.drawText("Maximum " + (int) MAX, 10, 200, myPaint);
myCanvas.drawText("Prumer " + (int) prumerChyba, 10, 300, myPaint);
myCanvas.drawText("Cas " + df.format(pomTime), 10, 400, myPaint);
myCanvas.drawText("Tres " + df.format(tremor) + " Hz", 10, 500, myPaint);

myPaint.setStrokeWidth(8);
myPaint.setStyle(Paint.Style.STROKE);
myPath.reset();
}

```

#### Ukázka kódu 7: Shrnutí výsledků

Jako vždy se na počátku funkce deklarují pomocné proměnné pro dopočítání výsledků. Zmíním tu především proměnnou *df*, která je použita při samotném vypisování výsledků a zaokrouhluje čísla na tři desetinná místa.

Další v pořadí této funkce se počítá maximální chyba celého tahu, opět je nutné od ní odečíst hodnotu 22, což je šířka šablony spirály. Ihned následuje podmínka, která se ptá, není-li maximální chyba záporná. To se provádí v případě, že pacient neudělal jedinou chybu, pak se po odečtu 22 od nuly proměnná *MAX* stane zápornou a tato hodnota se vypíše do výsledků. Právě proto tato podmínka změní maximum na nulu a tím tento problém obchází.

Jeden z problémů, co jsem řešil, bylo jak vypisovat výsledky. Zkoušel jsem vytvořit *Layout* s textovým panelem, do kterého bych výsledky zapisoval a aktualizoval. Vznikaly ovšem komplikace s prováděním tohoto *Layoutu* s celým programem. Nakonec jsem tento problém vyřešil funkcí kreslení. Nejen, že je možné kreslit přes tuto třídu čáry, geometrické tvary jako obdélník a kružnice, ale jde vypisovat i text přímo na bitmapu. Tuto funkci dělá příkaz *myCanvas.drawText("text",x,y,Paint)*, kde se do uvozek napíše požadovaný text, souřadnice *x* a *y* jsou souřadnice počátku textu a *Paint* v sobě nese nastavení barvy, velikosti a tloušťky textu. Tento způsob zapisování výsledků má však

jednu nevýhodu, jedná se o aktualizování, kde se jeden text překrývá jiným. Proto se musí při aktualizaci překreslit starý text obdélníkem stejné barvy jako je pozadí. Toho je docíleno příkazem *myCanvas.drawRect*, který nakreslí obdélník. U příkazu se nastavují x a y souřadnice dvou rohů, obdélníku a opět třída *Paint*, nesoucí informace o barvě a podobně. Jelikož vypisují pět výsledků, je zapotřebí pěti překrývajících obdélníků.

Následuje vypočítání průměrů, procentuální úspěšnosti, času a frekvence třesu. První se spočítá průměrná chyba vydělením proměnných *prumer*, ve které jsou sečtené všechny velikosti chyb, jež pacient udělal a vydělí se počtem všech chyb v proměnné *bad*.

Procentuální skóre se počítá také velmi jednoduše  $skore = 100 \cdot \frac{good}{count}$ . Další je na řadě vypočtení celkového času. To je chráněno podmínkou, která nedovoluje započtení velmi malých časů (vznikají například u klepnutí do obrazovky), což způsobovalo problémy u zaokrouhlování. Čas ze všech provedených tahů je postupně čítán do proměnné *pomTime*, ale jelikož je měřen v mikrosekundách, je nutné toto číslo vydělit tisícem.

Jako poslední se dopočítává frekvence třesu. Podle definice je frekvence rovna převrácené hodnotě časové periody a proto je nutné vypočítat nejprve průměrný čas všech chyb. Jelikož ho dostaneme v mikrosekundách, musím tento čas ještě vydělit tisícem a tímto výsledkem ve vteřinách vydělím jedničku.

$$tremor = \frac{1}{\frac{avrTremor}{tremorCounter} \cdot \frac{1}{1000}} \quad (4)$$

Poslední část již pouze nastavuje šířku písma a pět řádků vypsání výsledků. U některých hodnot je nutné ještě zaokrouhlit na tři desetinná místa a po vypsání ještě znovu přenastavit šířku kreslené části.

#### 4.2.5 Menu

Poslední důležitou částí je vytvoření menu, kterým se ovládá běh programu. To se nachází v pravém horním rohu a obsahuje čtyři možnosti. První je možnost kreslení *Paint*, která pouze nastavuje šířku tahu na velikost 12. Druhá možnost *Erase* dovoluje smazání tahu a zresetování všech proměnných potřebných k novému vyhodnocení následujícího pokusu. Třetí položka *Save* obstarává ukládání celé malby a výsledků do souboru formátu jpg. Čtvrtá a poslední možnost *Exit* je na vypnutí programu.

Všechny tyto položky jsou v podstatě jednoduché, výjimkou je ovšem ukládání, kde se musí nastavit složka pro ukládání jednotlivých souborů. Ty se automaticky ukládají na SD kartu přístroje do složky „Parkinson“ a pokud se tato složka na kartě nenachází, vytvoří ji.

Když již máme vytvořenou složku, jednotlivé soubory se ukládají jako test0000.jpg, kde se kontroluje, zda-li se již soubor se stejným číslem nenachází. Pokud ano, číslo se mění od 0000 do 9999, což by mělo bohatě stačit. Program zjistí nejmenší možné číslo a uloží obrázek.

```

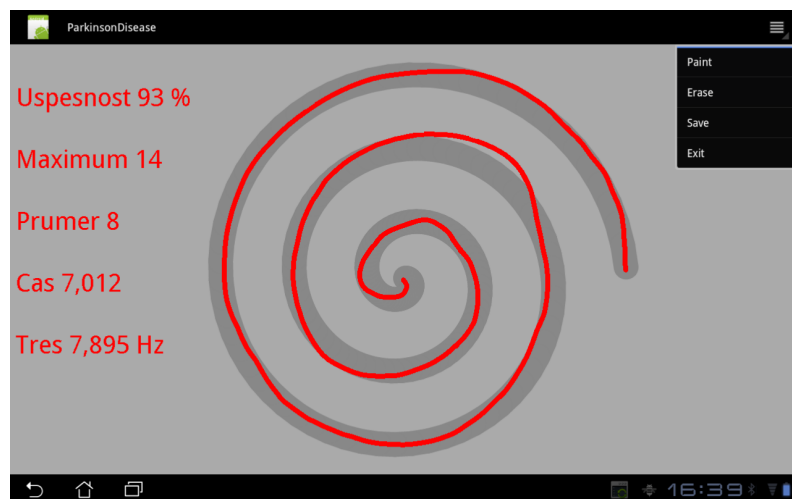
FileOutputStream fos = new FileOutputStream(dirname+ "test"
        + l + "" + k + "" + j + "" + i + ".jpg");
myBitmap.compress(CompressFormat.JPEG, 75, fos);
fos.flush();
fos.close();

```

#### Ukázka kódu 8: Příkazy pro zapsání do souboru

V této ukázce je znázorněn princip ukládání do souboru (Ukázka kódu 8). Je to již po části kódu, kde se zjistí hodnoty  $i, j, k$  a  $l$  tak, aby nové číslo bylo nejmenší neobsazené. Do proměnné  $fos$  je uloženo jméno souboru, do kterého se bude obrázek následně ukládat. Poté se celá bitmapa a vše, co je přímo do ní vkreslené zkompreseje do formátu .JPG a uloží do příslušného souboru. Pak je nutné soubor a zápis přerušit a ukončit.

Nyní již ukáží náhled celého programu včetně menu (Obrázek 10). A je čas se přesunout k otestování mé aplikace.



Obrázek 10: Náhled na finální aplikaci

## 5 Testování

Testování mé aplikace jsem měl v úmyslu provést přímo na pacientech s Parkinsonovou nemocí, ale z časových důvodů potřebných k vyhotovení projektu, který by byl nutný k přístupu k pacientům s PN, jsem se rozhodl vydat jinou cestou.

Samozřejmě byla i možnost nechat tablet s aplikací na pohotovosti, kde po domluvě s lékaři by pacienti s třesem program vyzkoušeli. Problém tohoto řešení je ovšem v nejistotě dostatku vzorků a nemohl bych zajistit svou osobní přítomnost při jejich sběru.

Nakonec jsem se rozhodl vyzkoušet mou práci na dvou skupinách lidí a poté tyto výsledky porovnat. První kontrolní skupinu tvořila třída z Pražské vysoké školy Psychosociálních studií v Hekrově ulici a druhou byli klienti z domu s pečovatelskou službou v Praze 10, Janovská 486.

U první skupiny předpokládám, že se jedná o lidi mladé, zdravé a kteří mají alespoň základní znalosti práce s počítačem a proto by jim zacházení s tabletem nemělo dělat starosti. Druhou skupinou jsou naopak lidé staří, u kterých se mohou projevat některé nemoci postihující motoriku a lze také předpokládat, že ne všichni budou umět zacházet s obyčejným počítačem, natož s tabletem ovládaným dotykovým displayem.

Pro obě skupiny byl vypočten aritmetický průměr  $\phi$  a jeho směrodatnou odchylku  $s$  pro každý jednotlivý parametr. Aritmetický průměr není nic jiného, než součet určitého parametru od všech pokusů vyděleného jejich počtem (5) a směrodatná odchylka určuje průměrný rozdíl mezi pokusy a průměrem (6). Výsledek se pak zapisuje ve tvaru  $x = \phi \pm s$ , viz tabulky 2,3, 4.

$$\phi = \frac{1}{N} \sum_{i=1}^N x_i \quad (5)$$

$$s = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (x_i - \phi)^2} \quad (6)$$

### 5.1 Skupina mladých lidí

Jak již bylo zmíněno, data od zdravých, mladých lidí jsem sbíral od třídy z Pražské vysoké školy Psychosociálních studií. Studentům jsem vysvětlil účel programu, jaké parametry aplikace měří. Studenti postupně přicházeli a každý si obkreslil spirálu, následně jsem celou malbu i s výsledky jsem ukládal pro budoucí vyhodnocení.

V den měření bylo přítomno 23 studentů a všechny jejich výsledky jsou uloženy i na CD v příloze, která je součástí tohoto dokumentu. Soubory se nachází ve složce \testing\26.4. 2012\. Podle předpokladu neměli studenti s obkroužením spirály moc problémů, jejich výsledky jsou sepsány v tabulce níže (Tabulka 2). Všichni studenti si mohli vybrat jakou rukou budou kreslit a proto předpokládám ze si zvolili svoji dominantní ruku.

Každému studentovi jsem ještě před začátkem testu zopakoval některé důležité informace a

případně zodpověděl jejich další dotazy, ale jelikož dotazů nebylo moc a test samotný trval jen krátce, otestování celé třídy zabralo asi třicet minut.

ID	Úspěšnost	Max	Průměr	Čas	Třes
[-]	[%]	[pixel]	[pixel]	[s]	[Hz]
1	95	3	2	5,75	3,39
2	99	0	0	16,43	11,63
3	100	0	0	17,91	0
4	89	39	5	7,69	8,60
5	84	17	8	4,31	3,64
6	96	27	9	16,67	23,67
7	92	6	2	8,79	9,25
8	100	0	0	19,04	0
9	100	0	0	10,18	0
10	98	19	10	10,35	0
11	97	1	1	11,07	11,63
12	91	33	9	4,52	17,75
13	100	0	0	15,80	0
14	81	14	5	7,69	3,49
15	97	4	2	13,49	11,68
16	89	10	5	5,45	5,09
17	100	0	0	22,76	0
18	97	20	12	16,70	0
19	86	13	6	5,03	5,56
20	95	10	5	4,05	9,43
21	97	4	1	19,82	4,73
22	99	0	0	13,39	23,81
23	97	4	3	21,21	1,82
$\phi \pm s$	$94,74 \pm 5,56$	$9,74 \pm 11,41$	$3,70 \pm 3,78$	$12,09 \pm 6,00$	$6,75 \pm 7,29$

Tabulka 2: Tabulka výsledků mladých lidí

Z tabulky výsledků je vidět, že studenti, kteří táhli pomalu a obezřetně dosahují velmi dobrých výsledků. Všichni studenti, kteří dosáhli úspěšnosti 96% a víc, trval tah déle jak 10 vteřin. Kdybych chtěl vytvořit funkci hodnotící úspěšnost vzhledem k trvání tahu, tak je zřejmé, že s narůstajícím časem by mělo hodnocení klesat, ale i nároky na úspěšnost by se měly zvyšovat.

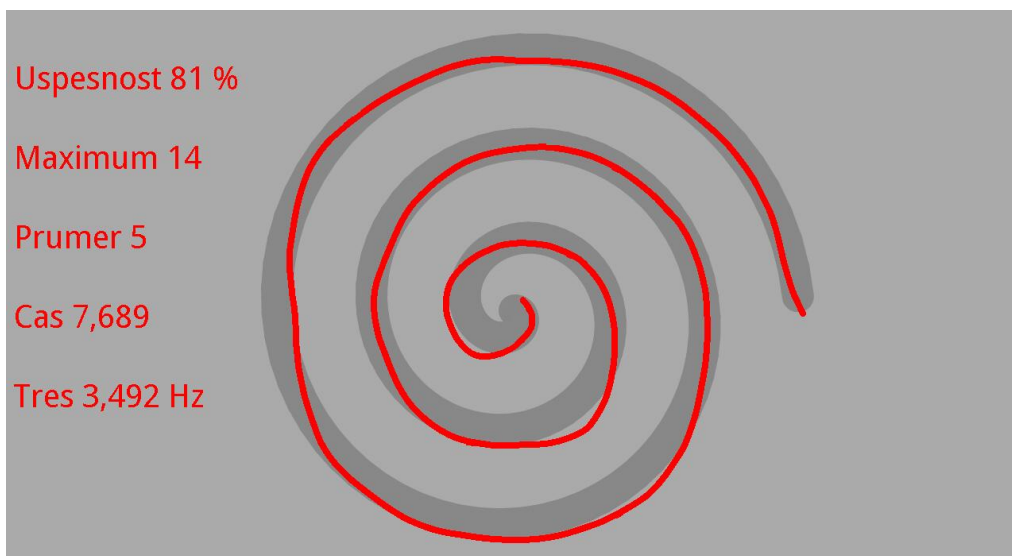


Vysoké frekvence třesu, které jsou například u pokusů číslo 2,6,12 a 22 jsou způsobeny tím, že student udělal pouze pár chyb a velmi rychle se z nich dokázal vzpamatovat. Protože je frekvence převrácená hodnota času, po ztrávení malé doby vně spirály velmi naroste právě frekvence. Proto je důležité u výsledků hledět na procentuální úspěšnost. Pokud je úspěšnost velmi dobrá, nemá cenu hledět na hodnotu třesu. Jako příklad, kdy vzniká velká míra třesu je i ta, kdy student při dokončování tahu přetáhl mimo spirálu a pak rychle prst zvedl. Toto se stalo například u pokusu číslo 12 (Obrázek 11).



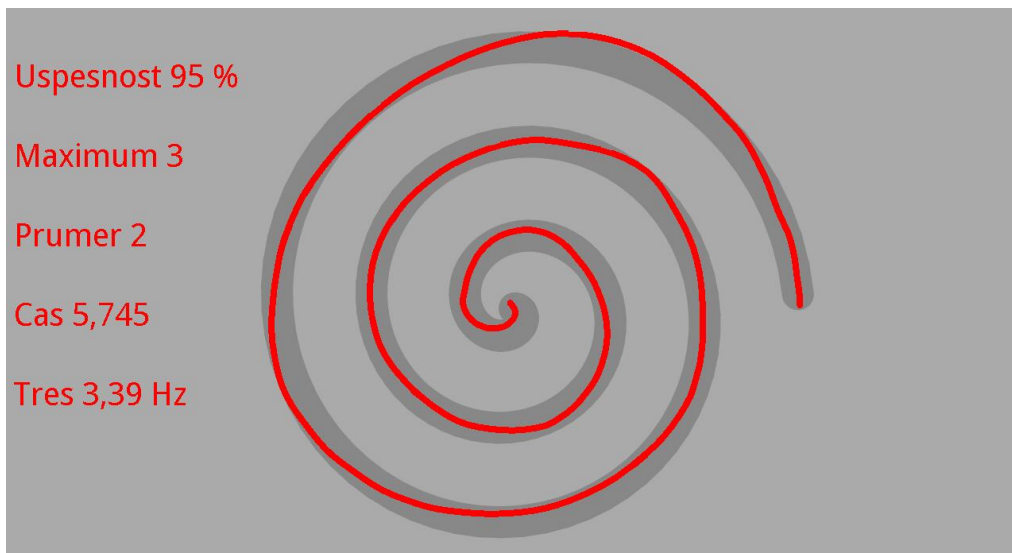
Obrázek 11: Ukázka přetažení spirály

Z tabulky je vidět, že nejhorší pokus udělal student číslo 14 (Obrázek 12). Tento pokus patří mezi ty rychlejší (čas cca. 7,7s) a je udělán jedním plynulým tahem, což navzdory špatnému skóre nenasvědčuje o možné PN studenta. Ten udělal chybu vždy u spodních částí spirály a mohla být způsobena například tím, že student přes svou vlastní ruku neviděl šablonu.



Obrázek 12: Ukázka studenta číslo 14

Nejlepší pokusem bych zvolil buď pokus číslo 1 a nebo pokus číslo 20, neboť oba studenti udělali tah velmi rychle a zároveň udělali jen minimum chyb. Ale díky menším chybám bych vyhlásil nejlepším pokusem právě pokus číslo 1 (Obrázek 13).



Obrázek 13: Ukázka studenta číslo 1

## 5.2 Skupina starých lidí

Ke skupině starých lidí jsem se dostal přes dům s pečovatelskou službou v ulici Janovské. Dům s pečovatelskou službou jsem si vybral proto, že se v něm samozřejmě nachází staří lidé, ale také

proto, že tito lidé si rádi zkusí něco nového.

Po několika rozhovorech se zástupci tohoto podniku jsem byl pozván ke klientům. Všechna naměřená data jsou na CD ve složce \testing\10.5. 2012\. Nejprve jsem ráno navštívil klienty s omezenou hybností na jejich pokojích. Po druhé jsem přišel odpoledne, protože se klienti schází ve společenské místnosti na kávu. Ve společenské místnosti jsem pohovořil o cílech programu, způsobu, jakým budou ohodnoceni. Tento den jsem odměřil nakonec 20 klientů a jejich výsledky jsou shrnuty níže (Tabulka 3). Opět jsem nechal jednotlivé klienty vybrat, jakou rukou budou kreslit.

Kreslení na tabletu nedělalo zkoušeným lidem problémy, po vysvětlení co je jejich úkolem byli všichni schopní kresbu absolvovat. Nejmladšímu klientovi bylo 68 let a nejstarší klientce bylo 92.

ID	Úspěšnost	Max	Průměr	Čas	Třes
[-]	[%]	[pixel]	[pixel]	[s]	[Hz]
1	93	41	13	30,01	10,89
2	51	37	12	7,73	1,36
3	81	16	7	14,74	3,49
4	85	14	4	26,99	1,65
5	45	72	17	7,08	1,82
6	78	20	9	6,34	4,37
7	97	4	2	13,52	13,09
8	53	34	9	17,57	1,01
9	93	9	4	12,72	3,75
10	90	15	6	9,04	4,50
11	94	6	2	10,08	10,00
12	40	37	15	14,84	1,75
13	70	247	59	6,59	3,31
14	89	5	2	12,72	6,84
15	95	6	2	9,66	11,82
16	95	7	2	11,87	5,91
17	67	12	6	5,53	3,54
18	99	4	3	16,81	26,32
19	86	11	5	10,35	3,70
20	87	69	13	6,42	3,10
$\phi_1 \pm s_1$	$79,4 \pm 18,62$	$33,3 \pm 54,27$	$9,6 \pm 12,57$	$12,53 \pm 6,55$	$6,11 \pm 5,99$
$\phi_2 \pm s_2$	$79,89 \pm 19,00$	$22,05 \pm 23,90$	$7,00 \pm 5,59$	$12,84 \pm 6,58$	$6,26 \pm 6,12$

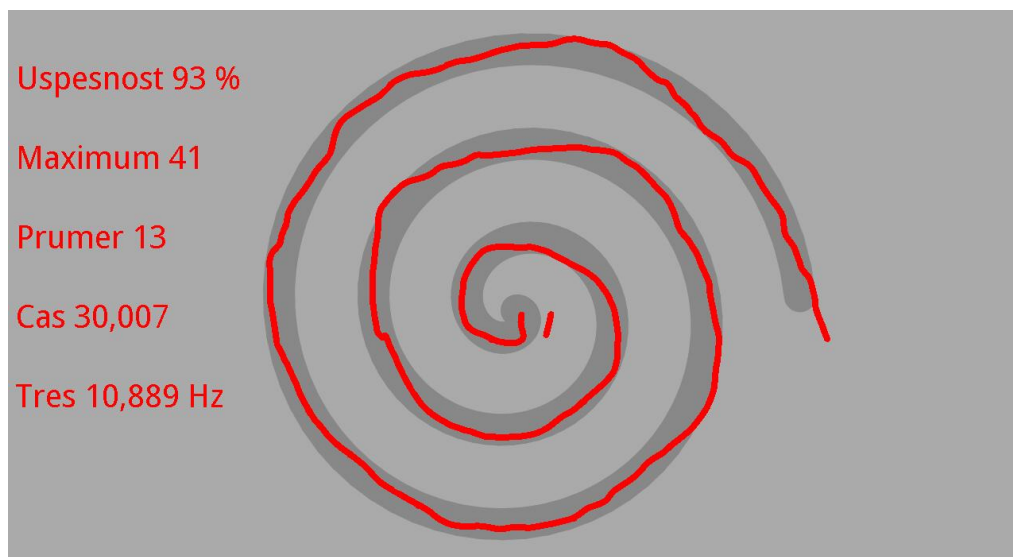
Údaje  $\phi_2 \pm s_2$  jsou vypočteny po odstranění 13 pokusu, který má chybnou maximální a průměrnou chybu

Tabulka 3: Tabulka výsledků starých lidí

Oproti studentům nemají staří lidé tak dobrou motoriku a i u pokusů, které trvali delší dobu není zaručena dobrá úspěšnost. Také se tu oproti studentům vyskytují i pokusy, které byli udělány na více tahů.

Hned první klientka, u které jsem byl, měla diagnostikovanou PN od lékaře a proto tento případ rozeberu trochu více podrobně. Na paní bylo vidět, že má problémy s pohybem a měla i viditelný klidový třes. Na obrázku (Obrázek 14) je vidět, že paní měla problém trefit se na počátek středu spirály. Nejprve udělala menší tah mimo šablonu a až na druhý pokus se trefila. Také čas tohoto pokusu byl extrémně dlouhý, nejvyšší ze všech mnou měřených pokusů a to 30 vteřin. Na samotné

trajektorii je vidět trhavý pohyb, nicméně co se týče úspěšnosti, je na tom velmi dobře. Při kreslení byl znát třes ruky, ale paní si ruku stabilizovala tím, že kreslícím prstem hodně tlačila na display, spolu s pomalým pohybem docílila jen malých cílených posuvů správným směrem. Na konci tahu je vidět opět přetažení mimo šablonu, ale i tak mě celý tento pokus velice překvapil.



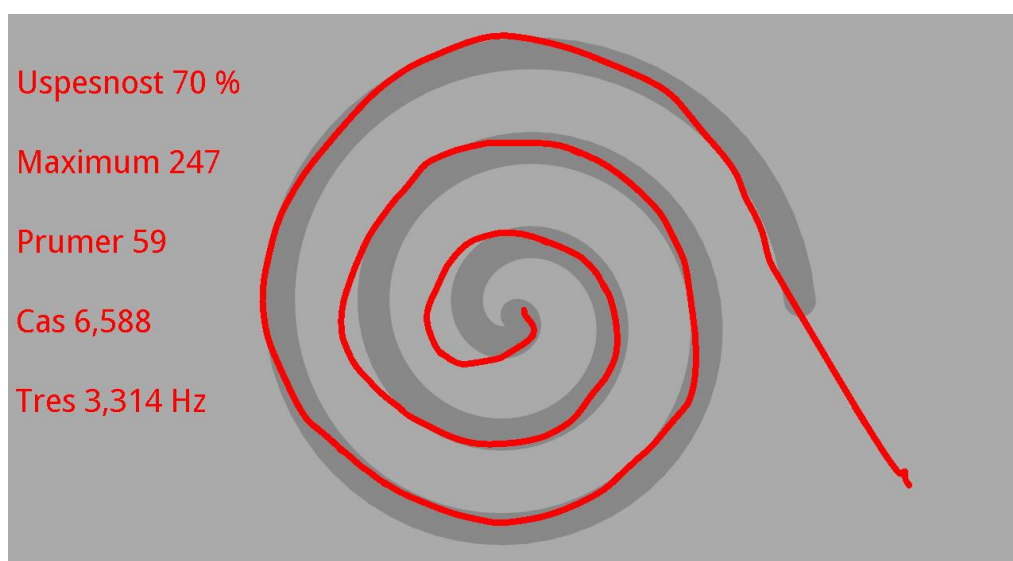
Obrázek 14: Ukázka klientky s diagnostikovanou PN

U této skupiny lidí se ukazuje horší procentuální úspěšnost. Je tu více pokusů, které mají úspěšnost kolem 50% a dokonce i méně. Stávalo se například to, že někteří lidé vůbec nebyli schopni udržet tah uvnitř předkreslené spirály a to dost výrazně (Obrázek 15), konkrétně pokus číslo 5 a 12.



Obrázek 15: Ukázka klienta číslo 5

Další pokus, který bych zde rád zmínil je číslo 13 (Obrázek 16). Zde se objevila velmi velká chyba, která byla způsobena omylem. Klient, který obkresloval spirálu z vnějšku dovnitř se ještě před dotykem prstu dotkl i částí zápěstí a pak teprve prstem. Po oddělení zápěstí z obrazovky program nakreslil přímkou od zápěstí k prstu. To způsobilo, že část tahu udělaný zápěstím ovlivnil hlavně maximální a průměrnou chybu. Tento markantní rozdíl mi ovlivnil i průměry a odchylky vypočtené v dolní části tabulky. Proto jsem vypracoval průměry jak pro všechna data ( $\phi_1 \pm s_1$  z Tabulky 3), tak pro data bez tohoto pokusu ( $\phi_2 \pm s_2$  z Tabulky 3). Jsou patrné znatelné změny u průměrů maximální a průměrné chyby a jejich odchylek. Naopak úspěšnost, čas a třes nejsou skoro ovlivněny.



Obrázek 16: Chyba způsobená zápěstím u pokusu č. 13

### 5.3 Porovnání obou skupin

Po shlédnutí výsledků zpracovaných tabulek (Tabulka 2, Tabulka 3) jsou na první pohled vidět rozdíly mezi oběma skupinami, pro větší přehlednost jsem výsledky dal do jedné tabulky (Tabulka 4). Dalo se očekávat, že mladí budou mít lepší úspěšnost a menší chyby.

Skupiny	Úspěšnost	Max	Průměr	Čas	Třes
[-]	[%]	[pixel]	[pixel]	[s]	[Hz]
$\phi_{mladi} \pm s_{mladi}$	$94,74 \pm 5,56$	$9,74 \pm 11,41$	$3,70 \pm 3,78$	$12,09 \pm 6,00$	$6,75 \pm 7,29$
$\phi_{2stari} \pm s_{2stari}$	$79,89 \pm 19,00$	$22,05 \pm 23,90$	$7,00 \pm 5,59$	$12,84 \pm 6,58$	$6,26 \pm 6,12$

Tabulka 4: Tabulka pro porovnání výsledků obou skupin

To se také potvrdilo na mnou měřených skupinách. Průměrná úspěšnost i s odchylkou mladých je  $\phi_{mladi} \pm s_{mladi} = 94,74 \pm 5,56$  [%], celkově je velmi dobrá a má i malou odchylku. Oproti tomu mají starší lidé úspěšnost  $\phi_{2stari} \pm s_{2stari} = 79,89 \pm 19,00$  [%], což je znatelně horší než u studentů a i odchylka je velmi velká, což ukazuje na výrazné rozdíly mezi jednotlivými pokusy.

Stejně to je i pro chyby maximální a průměrnou, kde mladí dosáhli více jak o polovinu lepších výsledků oproti starým v maximální chybě a skoro poloviční průměrné chyby. Odchylky opět ukazují více stabilní výkony mladších studentů před starými lidmi.

Tyto výsledky jsou pochopitelné a proto musíme brát v úvahu věk testovaných pacientů a v závislosti na tom ohodnotit celkový test. Co mě ovšem nejvíc překvapilo, jsou podobné hodnoty pro průměry i odchylky obou skupin u času a třesu. Ovšem když si vzpomenu na některé chybně velké frekvence třesu u mladých studentů, tak si uvědomuji jejich ovlivnění pro průměr i odchylku. U třesu starých lidí nejsou tak vysoké frekvence a proto se i výsledný průměr více blíží realitě.

Všechna data byla dále testována na normalitu. Výsledky se nachází v příloze B. Dále byla data zpracována statisticky podle výsledku normality. Protože se nepotvrdila normálnost dat, byla použita Kruskal–Wallisova metoda. Výsledkem testu byly signifikantní rozdíly mezi oběma skupinami při použití hladiny významnosti 5

## 6 Závěr

Otestováním mé aplikace na vzorku mladých a starších osob jsem vyzkoušel její funkčnost a zjistil závislost na věku jednotlivých lidí. Pro lepší stanovení diagnózy pro Parkinsonovu nemoc by bylo potřeba udělat hodnotící funkci, která by všechny parametry změřené aplikací ohodnotila určitou váhou v závislosti na věku testovaného člověka. Musel by být proměřen větší vzorek a použity složitější modely, jako například ANCOVA, kde by byl věk jedním z kovariantů. Tato funkce by musela brát v úvahu celkový čas tahu a podle něj stanovit hranice pro ostatní parametry. Výsledná hranice by oddělovala pacienty zdravé od nemocných například jako ve škole známkami 1 až 5.

Pro vypracování podobné hodnotící funkce by však bylo nutné provést více testování na skupinách lidí různého věku a zdravotní způsobilosti. Tím by se zjistily závislosti jednotlivých měřených parametrů, pro různé věkové skupiny.

Pokud bych v budoucnu prováděl nějaké podrobnější testování, zavedl bych komplikovanější podmínky provedení testu. Například by se nesmělo kreslícím prstem tolik tlačit na obrazovku tabletu, tento poznatek jsem si uvědomil u starší klientky domu s pečovatelskou službou, která měla lékařsky diagnostikovanou PN, kde se zvýšením tlaku na display stabilizoval třes. U tahu kresleném lehce by se měl projevit více třes. Dále bych odměřil výsledky z pravé i levé ruky, kde by pro dominantní ruku byly jiné podmínky než pro ruku druhou. Program samotný by bylo potřeba také trochu pozměnit, aby zaregistroval a ignoroval malé chybné časy ztrávené mimo spirálu, ty způsobovaly chybně velké frekvence.

Tento program může být ovšem užitečný pro samostatné vyšetření pacientů v jejich domovech a denním opakováním zjišťovat zhoršení, či zlepšení jemné motoriky. Po neustálém zhoršování výsledků by mohl být pacient předán do rukou odborných lékařů.



## 7 Literatura

### Reference

- [1] D. Smith, J. Friesen: *Android Recipes a Problem-Solution Approach*, Apress. 2011.
- [2] E. Růžicka: *Parkinsonova nemoc*. MINIMONOGRFIE ČSSN 4/06, Čes. a slov. Neurol. Neurochir., 69/102, 2006, No. 4, p. 241–258..
- [3] Emre M, Aarsland D, Brown R, Burn DJ, Duyckaerts C, Mizuno Y, Brue GA, Cummings J, Dickson DW, Gauthier S, Goldman J, Goetz C, Korczyn A, Lees A, Levy R, Litvan I, McKeith I, Olanow W, Poewe W, Quinn N, Sampaio C, Tolosa E, Dubois B: *Clinical Diagnostic Criteria for Dementia Associated with Parkinson's Disease*. 2007.
- [4] H. Braak, K. Del Tredici, H. Bratzke: *et al. Staging of the intracerebral inclusion body pathology associated with idiopathic Parkinson's disease (preclinical and clinical stages)*. 2002. 249 (Suppl. 3), p. 1301-1304.
- [5] P. Vondráček: *Parkinsonova nemoc*. (2008).  
<http://www.lecit.eu/pro-dospELE/parkinsonova-nemoc/>
- [6] E. Růžicka, J. Roth: *Parkinsonova nemoc*. (říjen 2006).  
<http://www.zdn.cz/clanek/postgradualni-medicina/parkinsonova-nemoc-271473>
- [7] Z. Mlčoch: *Esenciální třes, tremor - příznaky, projevy, příčina, léčba, genetická zátěž*. (prosinec 2008).  
<http://www.zbynekmlcoch.cz/informace/medicina/neurologie-nemoci-vysetreni/>
- [8] Lékařský slovník . (2008).  
<http://www.lekarske.slovniky.cz>
- [9] iTunes *Tremor Tracker*. (2011).  
<http://itunes.apple.com/us/app/tremortracer/id355455233?mt=8>
- [10] iTunes *Tremor Meter*. (2011).  
<http://itunes.apple.com/ca/app/tremor-meter/id447804073?mt=8>
- [11] Mobihealthnew *Wireless sensors, tablet computer measure Parkinson's tremors*. (2011).  
<http://mobihealthnews.com/13812/wireless-sensors-tablet-computer-measure-parkinsons-tremor>
- [12] Webcitation *Google Buys Android for Its Mobile Arsenal*. (2005).  
<http://www.webcitation.org/5wk7sIvVb/>

- [13]    Openhandsetalliance *Industry Leaders Announce Open Platform for Mobile Devices* .  
         (2007).  
         [http://www.openhandsetalliance.com/press\\_110507.html](http://www.openhandsetalliance.com/press_110507.html)
  
- [14]    Developer Android *What is Android?*. (2007).  
         <http://developer.android.com/guide/basics/what-is-android.html>

## Příloha A: Obsah CD

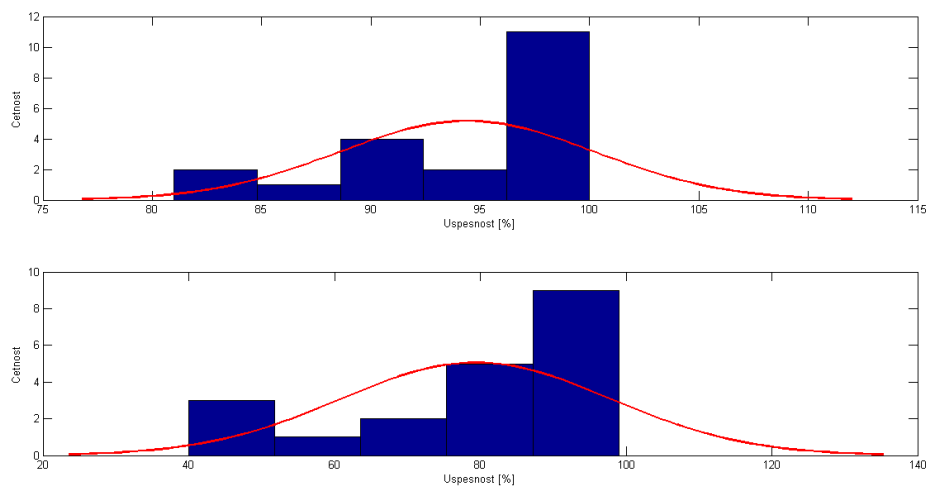
Zde je seznam souborů a složek nacházejících se na přiloženém CD.

<b>Zápis</b>	<b>Popis</b>
ParkinsonDisease	Složka s celým programem
Programing	Složka s obrázky z vývoje aplikace
Testing	Složka s obrázky obou testovaných skupin
BP_Jan_Kubeš.pdf	Dokument bakalářské práce ve formátu PDF
ParkinsonDisease.apk	Aplikace pro instalaci na tablet

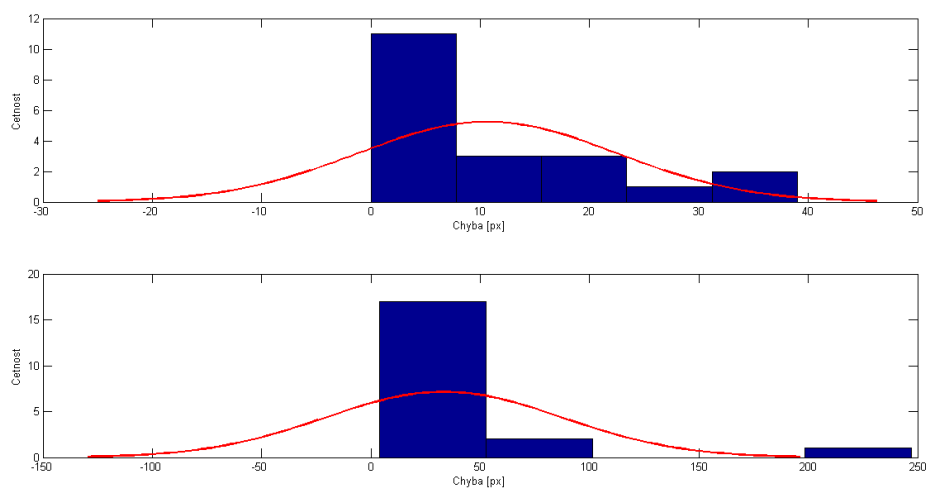
Tabulka 5: Obsah přiloženého CD

## Příloha B: Histogramy

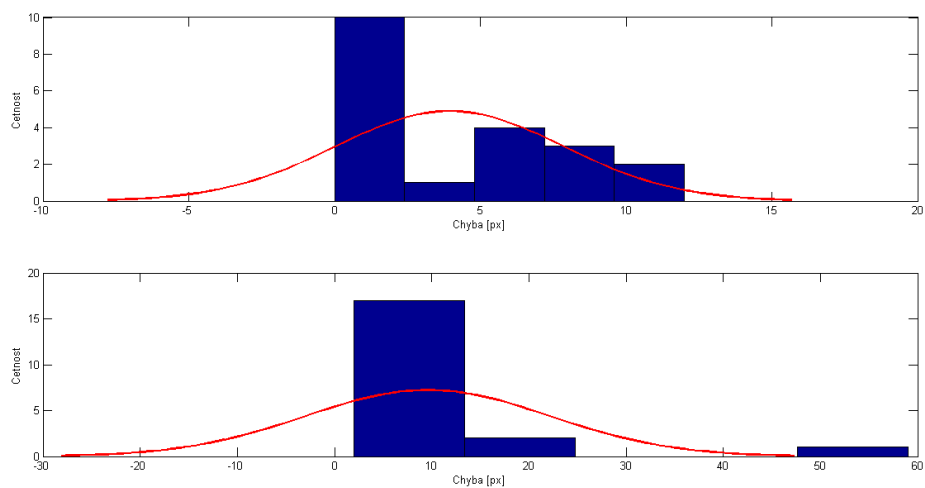
V horní části jsou výsledky od skupiny mladých a v dolní části jsou vždy výsledky od skupiny starých lidí. Každý histogram je proložen Gaussovskou křivkou metodou největší věrohodnosti.



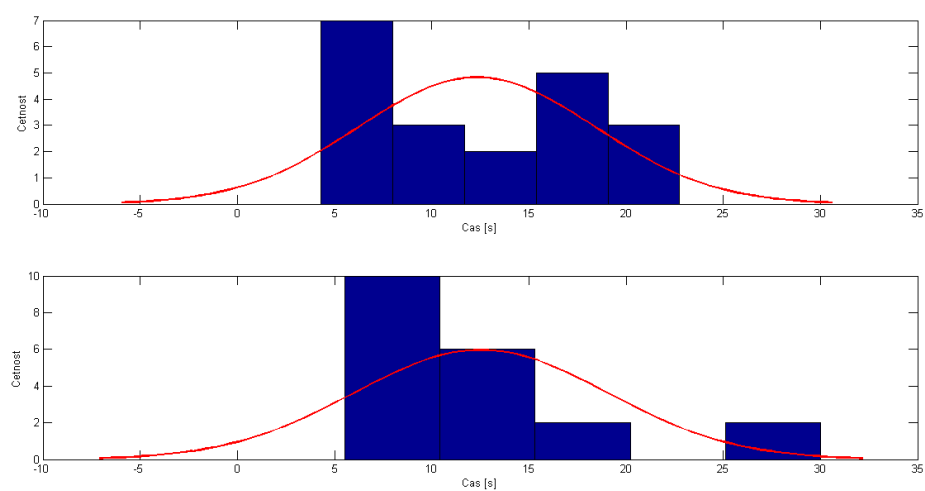
Obrázek 17: Histogram pro procentuální úspěšnost



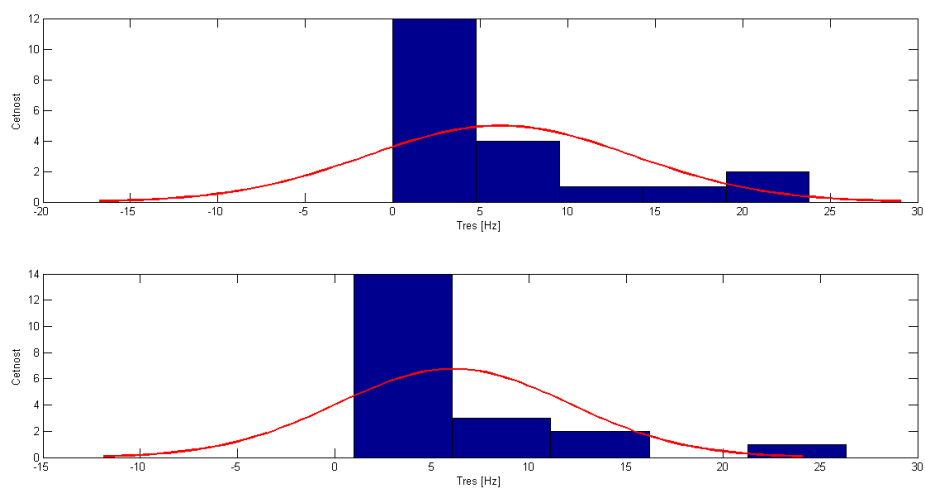
Obrázek 18: Histogram pro maximální chybu



Obrázek 19: Histogram pro průměrnou chybu



Obrázek 20: Histogram pro čas



Obrázek 21: Histogram pro třes