

# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

## PROGRAM PRO NÁCVIK TECHNIKY RYCHLÉHO ČTENÍ

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

JAN PADYŠÁK

BRNO 2012



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**  
BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ**  
**ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ**

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

# **PROGRAM PRO NÁCVIK TECHNIKY RYCHLÉHO ČTENÍ**

APPLICATION FOR PRACTICING RAPID READING TECHNIQUES

**BAKALÁŘSKÁ PRÁCE**  
BACHELOR'S THESIS

**AUTOR PRÁCE**  
AUTHOR

**JAN PADYŠÁK**

**VEDOUcí PRÁCE**  
SUPERVISOR

**Doc. Ing. ADAM HEROUT, Ph.D.**

BRNO 2012

## Abstrakt

Cílem této práce je vytvoření programu pro nácvik techniky rychlého čtení. Ve vytvořené aplikaci jsou implementována cvičení pro trénink různých vlastností lidského zraku. U jednotlivých cvičení je možné zvolit obtížnost dle zkušeností čtenáře. Program byl vytvořen ve vývojovém prostředí NetBeans IDE 7.0.1 v jazyce Java. V této práci popisují tipy pro čtení textu a návrh grafického uživatelského rozhraní aplikace. Dále obsahuje popis vybraných cvičení a jejich implementace. V poslední části této práce se zaměřuji na testování aplikace.

## Abstract

The aim of this bachelor's thesis is to create a training program for rapid reading techniques. This created application is implemented for training exercises of various abilities of human eye. For each of these exercise, we can choose the difficulty according to the experience of the reader. The program was created in development environment NetBeans IDE 7.0.1 in Java. In this thesis I describe tips for reading text and a graphic user interface. It also contains a description of selected exercises and their implementation. In the last part of this work I focused on testing this application.

## Klíčová slova

Rychlé čtení, tipy pro čtení, grafické uživatelské rozhraní, nácvik rychlého čtení, oční gymnastika, fixace, reakce očí, periferní vidění, rychlost čtení.

## Keywords

Rapid reading, tips for reading, graphics user interface, practicing rapid reading, eye gymnastic, fixation, eye reaction, peripheral seeing, speed reading.

## Citace

Jan Padyšák: Program pro nácvik techniky rychlého čtení, bakalářská práce, Brno, FIT VUT v Brně, 2012

# Program pro nácvik techniky rychlého čtení

## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Doc. Ing. Adama Herouta Ph.D. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....

Jan Padyšák  
11. května 2012

## Poděkování

Děkuji panu Doc. Ing. Adamu Heroutovi Ph.D., vedoucímu mé práce, za poskytnuté rady a připomínky během práce na tomto projektu.

© Jan Padyšák, 2012.

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

# Obsah

<b>1 Úvod</b>	<b>2</b>
<b>2 Rychlé čtení</b>	<b>3</b>
2.1 Počátky rychločtení . . . . .	3
2.2 Výhody rychločtení . . . . .	4
2.3 Nejlepší čtenáři . . . . .	4
2.4 Čtecí návyky . . . . .	5
2.5 Existující nástroje pro nácvik rychlého čtení . . . . .	6
<b>3 Vývojové prostředí NetBeans IDE</b>	<b>10</b>
3.1 Historie NetBeans . . . . .	10
3.2 Pomocné nástroje v NetBeans IDE . . . . .	11
3.3 Tvorba uživatelských rozhraní v NetBeans IDE . . . . .	11
3.4 Java Web Start . . . . .	12
3.5 Synth . . . . .	13
<b>4 Návrh programu pro nácvik techniky rychlého čtení</b>	<b>14</b>
4.1 Vývoj uživatelského rozhraní . . . . .	14
4.2 Počáteční návrh uživatelského rozhraní . . . . .	15
4.3 Finální návrh uživatelského rozhraní programu . . . . .	15
4.4 Výběr technik rychlého čtení . . . . .	17
<b>5 Implementace programu</b>	<b>22</b>
5.1 Implementace technik rychlého čtení . . . . .	22
5.2 Statistiky . . . . .	24
5.3 Uchování dat . . . . .	25
5.4 Kompilace a spuštění programu . . . . .	26
5.5 Testování programu . . . . .	27
<b>6 Závěr</b>	<b>31</b>
<b>A Obsah CD</b>	<b>33</b>

# Kapitola 1

## Úvod

V dnešní době jsme zahrnováni informacemi ze všech stran. Ať už sledujeme televizi, anebo si čteme noviny či časopisy, přijímáme množství informací, které nejsme schopni dostatečně rychle zpracovat. Potřebujeme-li něco nastudovat, musíme většinou přečíst množství rozsáhlých publikací, jejichž přečtení nám zabere několik hodin. Právě pro tyto situace je vhodné rychločtení, díky kterému můžeme docílit zvýšení rychlosti čtení až o 300%.

V této bakalářské práci se zabývám technikou rychlého čtení a tvorbou počítačového programu sloužícího pro nácvik rychločtení. Hlavní částí této bakalářské práce je návrh a implementace programu pro nácvik této techniky.

Na začátku kapitoly **2** popisují vznik a historii rychlého čtení s výhodami, které přináší. Dále jsou v této kapitole zmíněni nejlepší čtenáři. Tato kapitola obsahuje také popis dobrých a špatných návyků, které nás ovlivňují během čtení. V závěru kapitoly najdete recenze několika programů, které jsou již vytvořeny pro nácvik rychločtení.

Následující kapitola **3** popisuje vlastnosti vývojového prostředí, ve kterém jsem se rozhodl tuto aplikaci implementovat.

V kapitole **4** popisují, pro jaké uživatele je aplikace určena. Také tato kapitola obsahuje návrh uživatelského rozhraní programu a popis vybraných technik pro trénink rychločtení.

Poté následuje kapitola **5**, ve které detailněji popisují implementaci jednotlivých cvičení a komponenty vytvořené pro tuto aplikaci. Konec této kapitoly je zaměřen na testování programu uživateli a vyhodnocení tohoto testování.

## Kapitola 2

# Rychlé čtení

Mluvíme-li o rychlém čtení, myslíme tím speciální techniku vnímání a porozumění textu, která umožňuje číst text rychleji a efektivněji. Většina průměrných čtenářů dosahuje rychlosti čtení kolem 200 slov za minutu. Tato rychlost odpovídá přibližně přečtení jedné stránky knihy za 1-2 minuty. Na základě tréninku rychločtení se čtenář zlepšuje průměrně až na rychlost kolem 600 slov za minutu. Zároveň přečteme-li text 3x rychleji, neznamená to automaticky, že si toho méně zapamatujeme. Rychločtení si klade za cíl kromě zvýšení rychlosti čtení, také zlepšovat i paměť čtenáře. I přes tyto výhody není vhodné používat rychločtení vždy, např. úřední dokumenty nebo smlouvy je vhodnější číst pomalu a pečlivě prostudovat každý detail.

### 2.1 Počátky rychločtení

Většina prvních kurzů zabývajících se rychločtením vycházela z neobvyklého zdroje informací, a to ze zkušeností královského letectva Velké Británie.

Na začátku první světové války nebyli piloti schopni v letu rozeznat dostatečně rychle typy letounů, které ve vzduchu míjeli. Tato situace byla v bojových akcích nebezpečná a okamžité rozpoznání nepřátelského letounu zachraňovalo pilotům životy. Proto vznikl speciální výcvik, ve kterém piloti trénovali rozpoznávání letadel. Během tohoto výcviku bylo pilotům promítáno několik snímků letounů na obrazovky během krátkého časového úseku pomocí tzv. tachistoskopu. Tento přístroj byl vytvořen právě pro tento výcvik. V pozdějších fázích tréninku se prodlevy mezi zobrazenými snímky zmenšovaly, čímž se zlepšovala reakce očí pilotů. Po absolvování tohoto výcviku zvládli zaznamenat čtyři snímky během jedné pěti-setiny sekundy a bezpečně rozeznat zobrazené typy letounů. Po první světové válce se zjistilo, že absolventi tohoto kurzu jsou nadprůměrní čtenáři, jelikož jejich zrak byl vytrénován pro pojmání mnoha informací během krátkého časového intervalu. Cvičení na principu tachistoskopu jsem implementoval také ve svém programu.

Za oficiální začátek rychločtení, jak ho známe dnes, můžeme považovat 20. léta 20. století, kdy vyšla kniha zabývající se technikou rychlého čtení [1], kterou napsal americký lektor filozofie a psychologie Walter B. Pitkin.

U nás se rychločtení začalo popularizovat až v 80. letech, kdy vyšel první překlad zahraniční knihy [2] věnující se této problematice. Poté se technice rychločtení začal věnovat v té době u nás populární časopis Mladý svět a zároveň vznikl televizní seriál *Kurs rychlého čtení*. V dnešní době mezi hlavní propagátory rychlého čtení v České Republice patří vysokoškolský učitel Ing. David Gruber, který vydal několik publikací o rychločtení, a také

pořádá od roku 1983 semináře na toto téma.

## 2.2 Výhody rychločtení

Technika rychlého čtení umožňuje číst průměrnému čtenáři text až trojnásobnou rychlostí. Cvičením této techniky se zlepšují vlastnosti zraku, jako je periferní vidění, reakce očí, a zmenšuje se počet fixací, které čtenář potřebuje pro přečtení určité části textu. Tato technika také umožňuje zapamatování si více informací z čteného textu a ulehčuje pozdější orientaci v přečteném textu. Pravidelným tréninkem této techniky se čtenář odnaučuje čtecím zlozvykům a učí se využívat co nejefektivněji svůj zrak.

## 2.3 Nejlepší čtenáři

Oproti průměrným čtenářům existují lidé, kteří jsou schopni číst až 100krát rychleji, a zároveň si většinu informací obsažených v textu zapamatují. V této kapitole jsou vzpomenu ti lidé, kteří svou rychlostí čtení ohromili svět.

### Howard Curtis Berg

Howard C. Berg se narodil v roce 1934, pracoval jako učitel biofyziky na Harvardově univerzitě. Od roku 1990 je zapsán v Guinnessově knize rekordů jako nejrychlejší čtenář na světě. Dokáže číst rychlostí až 25 000 slov za minutu. Vytvořil také 14 různých výukových kurzů pro trénink rychločtení. Jeho nejznámější publikace o rychlém čtení jsou [3] a [4].

### Sean Adam

Sean v mládí trpěl vážnou zrakovou vadou. Strávil několik let studiem zrakového systému s cílem zlepšit svůj zrak. Toto prováděl s obdivuhodným odhodláním a vytrvalostí.

V roce 1982 se mu konečně zrak zlepšil a začal pracovat na rychlosti svého čtení. Byl držitelem světového rekordu v rychlosti čtení 3 850 slov za minutu. Jako limit vlastních možností označil hranici kolem 4 550 slov za minutu.

### Vanda North

Vanda North se o rychločtení začala zajímat až jako prezidentka Mezinárodní společnosti pro zrychlené učení. Trénink rychločtení prováděla např. pomocí cviků popsanych v knize [6], jako jsou cviky pro zmenšení počtu fixací, zamezení regresi a rozšíření zrakového rozpětí. Po pár týdnech tréninku pomocí zmíněných cvičení již byla schopna číst rychlostí 3 000 slov za minutu.

### Prezident USA John F. Kennedy

John F. Kennedy nevynikal tak moc svou rychlostí, jako výše zmínění čtenáři, ale patří jistě k nejznámějším. Zpočátku jeho rychlost čtení nepřesahovala 300 slov za minutu, ale poté co začal studovat rychločtení, dosáhl rychlosti až 1 000 slov během jedné minuty.



## 2.4 Čtecí návyky

### Fixace

Čtení textu provádíme pomocí malých skoků. Jedná se o místo, kde se lidské oko zastaví a zaostří na určitou část textu. Text tedy plynule neprocházíme, ale skokově přeskakujeme po jeho jednotlivých částech, mluvíme o tzv. očních fixacích. Chceme-li číst rychleji, je důležité minimalizovat počet těchto fixací. Každá fixace stojí čas, nejdříve se oči musí přesunout na daný text, zaostřit na něj, a až poté jsme schopni čtení. Obyčejný čtenář, který nepraktikuje metodu rychlého čtení, využije na jeden řádek textu o šířce stránky A4 přibližně 7 fixací. Čtení obyčejného čtenáře můžeme vidět na obr. 2.1. Zobrazený text je rozdělen svislými čarami, které definují hranice jednotlivých fixací.

Lorem ipsum | dolor sit | amet, | consectetur | adipiscing elit, | sed do | eiusmod tempor | incididunt.

Obrázek 2.1: Fixace obyčejného čtenáře

Oproti tomu čtenáři trénující rychločtení mohou přečíst řádek přibližně na čtyři fixace, jak můžeme vidět na obr. 2.2.

Lorem ipsum dolor sit amet, | consectetur adipiscing elit, | sed do eiusmod tempor incididunt.

Obrázek 2.2: Fixace čtenáře trénujícího rychločtení

### Regrese

Další častou negativní vlastností čtenářů je vracení se v textu ke slovům, která právě přečetli, ale už si je nepamatují, a předpokládají, že obsahovala něco důležitého, a proto přečtou daný úsek textu znovu. Tuto vlastnost můžeme vidět na obr. 2.3. Regrese bývá způsobována nejčastěji nedostatečnou koncentrací na čtení.

Lorem ipsum | dolor sit | amet, | consectetur | adipiscing elit, | sed do | eiusmod tempor | incididunt.

Obrázek 2.3: Příklad regrese

### Pochopení textu

Číst text rychle automaticky neznamená zapamatovat si z něho méně informací. Čteme-li text pomalu, mozek nás nutí podvědomě číst pomaleji se stále menším pochopením. Čteme-li text rychleji, seskupujeme si přečtené informace do větších celků, a lépe pochopíme význam celého textu.

### Podmínky pro čtení

Chceme-li dosahovat skvělých čtecích výkonů a zároveň si chceme z textu hodně zapamatovat, měli bychom dbát na následující zásady, které zlepšují koncentraci čtenáře, a také mu

poskytují pohodlí. Naši koncentraci můžou narušovat okolní vlivy jako je vítr, prašné prostředí, suchý vzduch nebo hluk okolí. Je vhodné tyto nežádoucí prvky eliminovat. Proto, abychom vytvořili prostředí vhodné ke čtení, si rozebereme jistá doporučení, která nám vytvoří ideální prostor ke čtení.

## **Světlo**

Nejlepší pro čtení a studium je denní světlo. Je vhodné umisťovat pracovní stůl či křeslo, na kterém budeme číst, blízko okna. Pokud je noc nebo je zataženo, nezbyvá nám než číst pod umělým světlem. Osvětlení by mělo působit z opačné strany, než kterou rukou normálně píšeme, abychom si zbytečně nestínili. Zároveň by neměl být kontrast osvětlení knihy a okolní místnosti příliš veliký.

## **Pohodlí**

Budeme-li se nacházet v prostředí, v kterém nám bude až příliš pohodlně, může nás to vyvádět z koncentrace. Např. máme-li příliš měkké křeslo, do kterého se pohodlně usadíme, začne nás takovýto posed uspávat. Je vhodné tedy takovýmto případům předcházet a volit vždy takovou míru pohodlí, která nám není nepříjemná, a zároveň se nám nesnaží odvést myšlenky jinam.

## **Pracovní stůl a křeslo**

Dalším, neméně důležitým prvkem pro správné čtení, je volba vhodného stolu a křesla. Stůl by měl být průměrně vysoký okolo 70 - 80 cm. Čím větší stůl, tím více na něj můžeme dát věci, které nás budou rozptylovat. Důležitý je taky materiál stolu, např. z kovových či skleněných stolů nás mohou v chladnějších dnech studit ruce.

Špatná ergonomie židle může způsobovat i zdravotní poruchy. Vhodné je, aby bylo křeslo otáčecí, a rovněž mělo nastavitelnou výšku sedáku. Správné sezení lze dodržet pomocí pravidla pravých úhlů, kdy chodidla máme položená na zemi, či podložce. Mezi lýtky a stehny máme pravý úhel, tak jako mezi zády a stehny, a také mezi předloktím a pažemi.

## **Barva stěn**

Neméně důležitou částí je barva stěn, které obklopují prostor kolem nás. Je vhodné volit světlejší a decentnější barvy před kontrastními.

## **Vzdálenost očí od textu**

Optimální vzdálenost očí od textu je pro čtení velmi důležitá. Nachází-li se text moc blízko našim očím, tj. ve vzdálenosti kolem 20cm, naše zorné pole je velice omezené. Je vhodné být od textu ve vzdálenosti okolo 40cm, kdy náš zrak dokáže pokrýt celou šířku stránky. Přesnou vzdálenost, při které člověk vidí ostře co největší část textu si musí, ale každý najít sám.

## **2.5 Existující nástroje pro nácvik rychlého čtení**

Pro nácvik rychlého čtení lze využít počítačové programy nebo praktikovat cviky popsané v knihách o této problematice. Je známo, že čtení klasických papírových knih je o 10 %

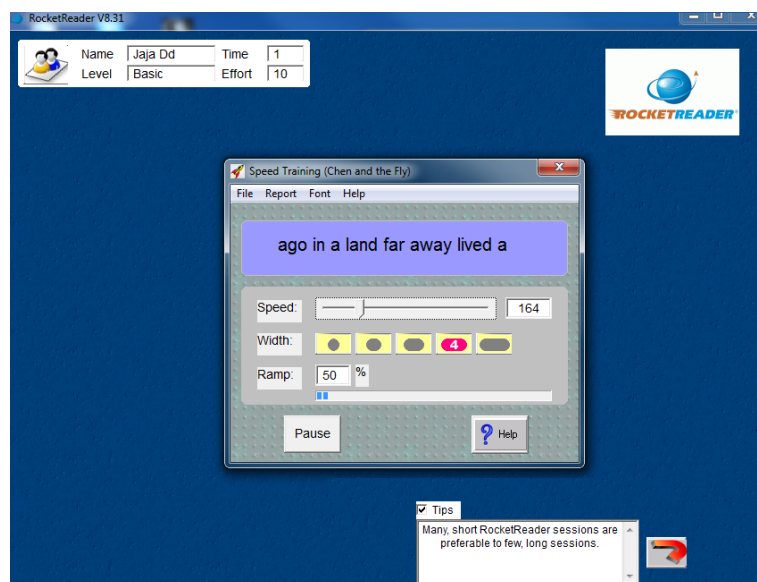
rychlejší, než přečtení stejné knihy v elektronické podobě. Také při čtení elektronické knihy jsou oči unavovány více vyzařováním světla z obrazovky, než při čtení klasické knihy. Proto při nábviku rychlého čtení mohou dát někteří čtenáři přednost cvičením s tištěným textem, ale je vhodnější zakomponovat do tréninku i programy k tomuto účelu vytvořené. Tyto programy oproti knihám obsahují různá interaktivní cvičení a rovněž zásoba cvičného textu je v podstatě neomezená, na rozdíl od knih, které obsahují jen několik textů pro různá cvičení.

## Programová řešení

Trénink rychlého čtení pomocí počítačové aplikace je prováděn pomocí sérií cvičení, které mohou být spuštěny uživatelem daného programu. Jednotlivá cvičení bývají zaměřena na některou z dovedností zraku, např. oční gymnastiku, periferní vidění, reakce očí nebo oční fixace. Aplikací vytvořených pro trénink rychlého čtení je mnoho. Tyto programy jsou ale pouze v placené verzi, a bohužel bez české lokalizace. Rozhodl jsem se detailněji popsat tři programy vytvořené pro nábviku techniky rychlého čtení, a to jmenovitě programy RocketReader, The Reader's Edge a AceReader Pro.

### RocketReader

Při prvním spuštění aplikace RocketReader je uživatel nucen vytvořit si vlastní profil, při jehož tvorbě je vyžadováno zadání příjmení i jména a zvolení hesla. Spuštěná aplikace je maximalizovaná, a nelze ji zmenšit ani přemístit. Chce-li uživatel spustit některé ze cvičení, je nucen projít přes nabídku, zda chce opravdu spustit cvičení, poté je zobrazena nápověda pro dané cvičení a poté až je možné začít provádět vybraný cvik. Zobrazený text v testech není rozdělen na stránky, ale uživatel musí text procházet pomocí posuvníku. Ovládání programu je velmi nepřehledné. Celkový dojem z aplikace nejvíce kazí neustále vyskakující informativní okna a různé nabídky, které musí uživatel potvrzovat. Tento program je možné stáhnout v demoverzi na stránkách [7].

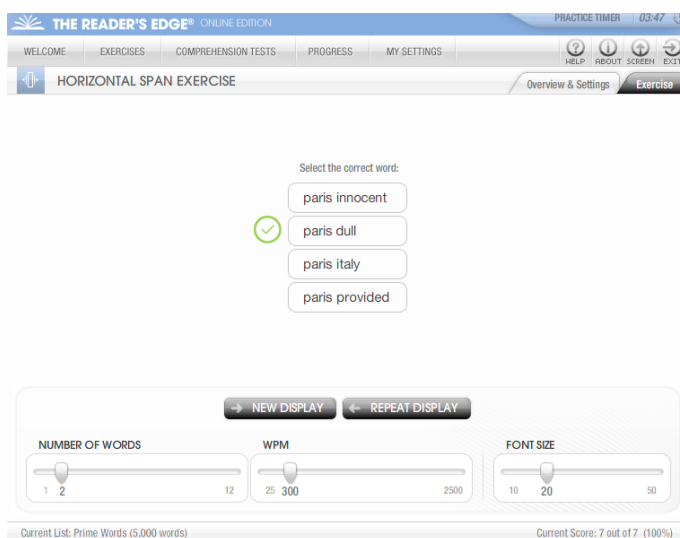


Obrázek 2.4: RocketReader

## The Reader's Edge

The Reader's Edge, oproti jiným programům, obsahuje kromě jednotlivých cvičení také výukové lekce, které jsou složeny z jednotlivých článků zaměřených na problematiku čtení textu.

Po spuštění aplikace je nám zobrazena úvodní obrazovka, která obsahuje nabídku pro rychlé spuštění nejdůležitějších částí této aplikace, a to jsou testy rychlosti, výukové lekce a jednotlivá cvičení. Bohužel, tento program neobsahuje cvičení na oční gymnastiku, která slouží jako rozcvička očí před delším čtením nebo tréninkem rychločtení. Jednotlivá cvičení lze spustit v předem definovaných obtížnostech a zároveň obsahují množství vlastností, které si může uživatel nastavit dle sebe. V programu se nachází také podrobná statistika obsahující informace o počtu spuštěných cvičení, dosažených výsledcích a celkové době tréninku. Také obsahuje vyhodnocení zlepšení uživatele ve formě grafů a výpisu průměrných dosažených hodnot ve cvičeních. Aplikaci je možné vyzkoušet zdarma online v omezené verzi na stránce [8]. Tato aplikace má nejpřívětivější uživatelské rozhraní z testovaných aplikací. Zároveň i přítomnost výukových lekcí dělá tento program zajímavým zdrojem nových informací.



Obrázek 2.5: The Reader's Edge

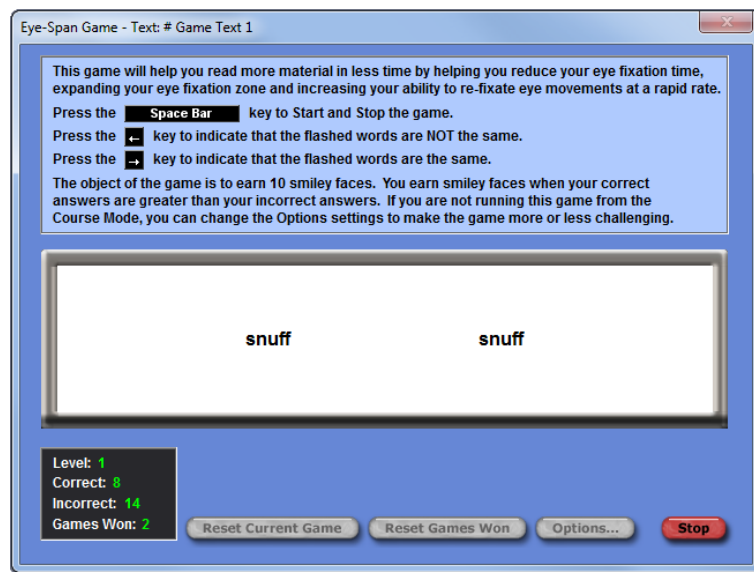
## AceReader Pro

Tento program je funkčně podobný jako The Reader's Edge.

Jeho nevýhody oproti výše zmíněnému programu jsou absence výukových textů, a také má nepřehledné uživatelské rozhraní. V AceReader Pro jsou všechny ovládací prvky formou textu, proto je i po delší době práce s tímto programem nepřehledná. Tato aplikace je dostupná v omezené verzi na stránkách [9].

## Jiná řešení

Nechceme-li nebo nemáme-li prostředky pro použití programů pro nácvik rychlého čtení, můžeme cvičit tuto techniku dle cviků popsanych v publikacích zabývající se rychločtením, nebo se můžeme přihlásit na intenzivní kurz rychločtení, který je veden lektorem.



Obrázek 2.6: AceReader Pro

Následuje popis cvičení, které slouží k odnaučení regrese, a není k němu potřeba víc než jen tužka a papír.

### Cvičení na odnaučení regrese

Cílem tohoto cvičení je odnaučit se regresi. K tomuto cvičení nám stačí vypsát náhodná čísla do sloupečku pod sebou. Dále budeme potřebovat kousek papíru, kterým budeme jednotlivá čísla zakrývat a tužku, kterou budeme tato čísla přepisovat. Cvičení se provádí tak, že čteme jednotlivá čísla na řádce a zakrýváme je ihned kouskem papíru. Zároveň toto číslo přepisujeme vedle něj. Jelikož je zobrazené číslo přikryto papírem, nemůžeme se podvědomě vracet a přečíst si ho podruhé. Takto postupujeme, až dojdeme na poslední číslo ve sloupci. Obtížnost se postupně zvyšuje zakrýváním stále větších čísel, která jsme nuceni si zapamatovat. Ukázkou tohoto cvičení můžeme vidět na obr. 2.7.

222.....  
 295.....  
 2889.....  
 94829.....  
 29832.....  
 09834.....

Obrázek 2.7: Odnaučení regrese

## Kapitola 3

# Vývojové prostředí NetBeans IDE

V této práci bylo cílem vytvořit počítačovou aplikaci. Musel jsem si proto vybrat programovací jazyk a vývojové prostředí, které bych pro implementaci programu použil. Rozhodoval jsem se mezi jazykem C++ a jazykem Java. V jazyce C++ je známé vývojové prostředí Qt Creator. Jedná se o multiplatformní vývojové prostředí, které obsahuje integrovaný grafický návrhář, který znatelně ulehčuje tvorbu desktopových aplikací.

Mezi nejznámější vývojová prostředí pro jazyk Java patří Eclipse a NetBeans IDE. Tato vývojová prostředí jsou velmi podobná. Eclipse je vhodný pro větší projekty, protože obsahuje větší množství pluginů než zmiňovaný NetBeans. Pro můj program jsem ale tyto knihovny nepotřeboval, a výhodnější by pro mě bylo vývojové prostředí NetBeans, které oproti Eclipse obsahuje zabudovaný grafický návrhář uživatelského rozhraní. Eclipse sice dovoluje dohrát podobný návrhář pomocí pluginů, ten ale nedosahuje kvalit zabudovaného návrháře v NetBeans.

Jazyk Java je interpretovaný jazyk, který je kompilován na tzv. bytecode, který je interpretován virtuálním strojem (JVM). Umožňuje snadnější tvorbu programů než jazyk C++, zároveň aplikace vytvořené v tomto jazyce jsou multiplatformní, pokud nevyužíváme specifické knihovny napsané v jiném jazyce. Další výhodou je technologie Java Web Start, kterou lze použít ke snadné distribuci programu. Tato technologie je detailněji popsána v kapitole 3.4. Rychlá tvorba nových aplikací v jazyce Java a snadná distribuce programu pomocí technologie Java Web Start byla hlavním důvodem pro implementaci programu v jazyce Java.

### 3.1 Historie NetBeans

Začátky vývojového prostředí NetBeans sahají až do roku 1996, kdy vznikl na Matematicko-fyzikální fakultě Univerzity Karlovy jako školní projekt pod názvem Xelfi. V roce 1997 vznikla společnost, která vytvářela NetBeans IDE jako komerční verzi, kterou v roce 1999 koupila firma Sun Microsystems a v následujícím roce změnila NetBeans IDE z komerčního softwaru na Open Source.

Vývojové prostředí NetBeans je Open Source a celé je napsáno v jazyce Java, díky tomu může být spuštěno na všech platformách, které podporují JVM (Java Virtual Machine), a to např. MS Windows, Mac OS, Linux a další. Vývojové prostředí slouží ke zvětšení efektivity programátora, který nemusí používat jednotlivé individuální vývojové nástroje, ale má toto vše dostupné v jednom programu. Hlavním sponzorem projektu NetBeans je firma Sun Microsystems. NetBeans je vytvářen komunitou, která se neustále rozrůstá, a

také NetBeans disponuje téměř 100 partnery po celém světě.

## 3.2 Pomocné nástroje v NetBeans IDE

Vývojové prostředí NetBeans IDE obsahuje řadu pomocných nástrojů, které usnadňují vývoj aplikace. Následující popsané nástroje jsou pouze těmi základními, které jsou dostupné ihned po instalaci NetBeans IDE, a které jsem využíval při tvorbě programu pro nácvik rychločtení. Dále existuje množství dalších nástrojů, které lze vložit do vývojového prostředí pomocí pluginů, a tím ho rozšířit o další funkce.

### Grafický návrhář UI

Ulehčuje a urychluje tvorbu uživatelských rozhraní skládáním a propojováním jednotlivých grafických prvků programu. Současně dochází k automatickému generování kódu, který tyto jednotlivé závislosti popisuje.

### Debugger

NetBeans IDE obsahuje kvalitní debugger, který umožňuje umisťovat zarážky na různá místa ve zdrojovém kódu a umí postupně procházet zdrojovým kódem, vstupovat a přeska-  
kovat implementované funkce atd. Také po najetí kurzoru na jakoukoliv proměnnou zobrazí její aktuální hodnotu. Dále umožňuje ladění více vláknových programů. V tomto případě se v okně editoru objeví pomocný panel se seznamem aktuálně probíhajících vláken, na kterém se můžeme mezi těmito vlákny přepínat.

### Správa verzí programu

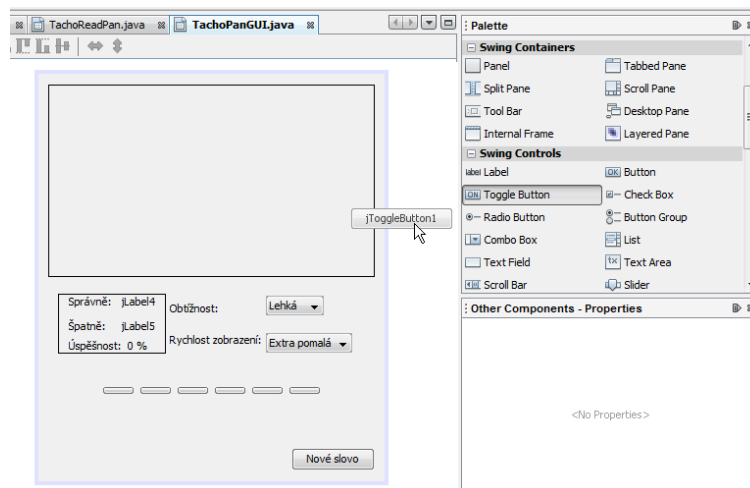
Užitečný nástroj, který automaticky uchovává veškeré verze zdrojových kódů programu, které bývají v tomto prostředí psány. Uživatel může procházet minulé verze kódů a může se lehce navrátit k předešlým verzím zdrojového kódu.

## 3.3 Tvorba uživatelských rozhraní v NetBeans IDE

Uživatelské rozhraní může programátor vytvořit pomocí psaní kódu v programovacím ja-  
zyce, anebo využije nástroj pro vizuální programování, a to tzv. grafický návrhář UI, který je v NetBeans IDE dostupný.

### Vizuální programování

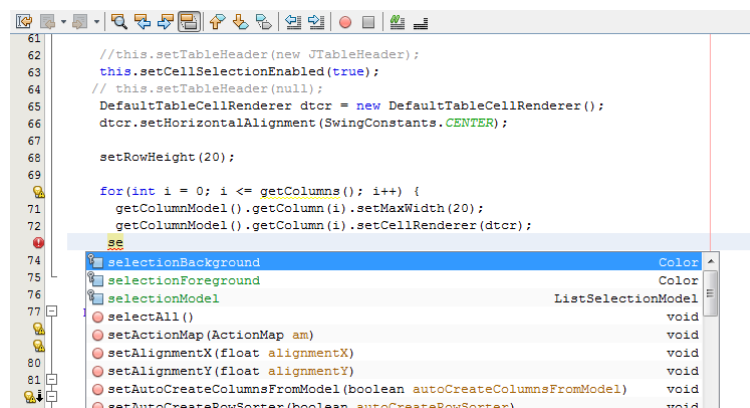
V dnešní době se přikládá větší důraz na vizuální programování, při kterém programátor vy-  
užívá grafický návrhář uživatelského rozhraní. Pro vytvoření aplikace v grafickém návrháři stačí uživateli jen přemístit a propojit jednotlivé prvky vybrané ze seznamu komponent do-  
stupných v návrháři. Zároveň dochází k automatickému generování zdrojového kódu, který si může programátor upravit dle svých specifických požadavků. Práci s grafickým návr-  
hářem můžeme vidět na obr. 3.1, na kterém je zobrazena tvorba cvičení pro zlepšení reakce  
očí, které je použito v této aplikaci. Tento grafický návrhář jsem využíval často, ať už při  
implementaci jednotlivých cvičení nebo při tvorbě uživatelské rozhraní celého programu.



Obrázek 3.1: Grafický návrhář UI v NetBeans IDE

## Psaní kódu

Pro psaní zdrojového kódu je v NetBeans IDE integrován vlastní editor, který můžeme vidět na obr. 3.2. Psaní kódu ulehčuje nabídka doplňování zdrojového kódu, která bývá automaticky zobrazována při psaní kódu. Tuto nabídku můžeme vidět na obr. 3.2.



Obrázek 3.2: Automatické doplňování zdrojového kódu v NetBeans IDE

## 3.4 Java Web Start

Java Web Start je technologie sloužící k spuštění aplikace přes internet pomocí jednoho kliknutí ve vašem prohlížeči. Zároveň obsahuje i bezpečnostní prvky, takže integrita dat souborů není ohrožena.

S technologií Java Web Start můžete spustit aplikaci kliknutím na odkaz umístěný na webové stránce. Pokud se aplikace nenachází na vašem počítači, automaticky stáhne potřebné soubory z URL adresy definované v JNLP souboru a aplikaci spustí pomocí JRE.



## Kompilace JWS

Pro spuštění aplikace pomocí JWS je nutné nastavit tuto informaci v překladači. Takto zkompileovaný program je poté nahrán na server, z kterého si ho můžou uživatelé stáhnout a spustit.

## Soubor JNLP

Soubor JNLP (Java Network Launching Protocol and API) je XML dokument, který slouží k popisu aplikace šířené pomocí JWS. Detailní popis souboru JNLP můžeme nalézt v knize [5].

## 3.5 Synth

V jazyce Java je vzhled aplikace definován pomocí třídy Look and Feel. Nejsme-li spokojeni se základním vzhledem aplikace, který nám nabízí prostředí Look and Feel, můžeme si vytvořit vzhled vlastní. Máme dva způsoby vytvoření vlastního vzhledu. Můžeme implementovat celý balík tříd, který bude obsahovat popis vzhledu v programovacím jazyce, nebo můžeme využít Synth, který umožňuje uživatelsky přívětivější definování vzhledu aplikace pomocí XML souboru, který slouží k inicializaci vlastního Java Look and Feel prostředí. Tento způsob nastavení grafického vzhledu aplikace jsem použil hlavně z důvodu oddělení vzhledu celé aplikace a umístění ho do jediného souboru. Takto definovaný vzhled aplikace lze jednoduše a rychle změnit. Např. pomocí přepsání jednoho řádku v tomto souboru můžeme změnit vzhled všech tlačítek v programu.

XML soubor definující vzhled aplikace může vypadat následovně:

```
<!-- Style that all regions will use -->
<style id="backingStyle">
  <!-- Make all the regions opaque-->
  <opaque value="TRUE"/>
  <font name="Dialog" size="14"/>
  <state>
    <color type="BACKGROUND" value="RED"/>
    <color type="FOREGROUND" value="BLUE"/>
  </state>
</style>
<bind style="backingStyle" type="region" key=".*"/>
```

V tomto XML souboru definujeme vzhled všech komponent, a to určením typu fontu a jeho velikostí. Dále se každé komponentě nastaví daná barva popředí a pozadí.

## Kapitola 4

# Návrh programu pro nácvik techniky rychlého čtení

Cílem aplikace je, aby uživatel pravidelně prováděl jednotlivá cvičení, a tím zlepšoval své vlastnosti zraku a rychlost čtení.

Při návrhu programu jsem se musel zabývat nejen otázkami, které techniky tréninku rychlého čtení bude vhodné zahrnout do mé aplikace, ale musel jsem také vytvořit vhodné uživatelské rozhraní programu. Aplikace by měla být určena všem věkovým kategoriím uživatelů. Z těchto důvodů jsem kladl důraz na jednoduché a srozumitelné ovládání, které bude jasné i méně zkušeným uživatelům. Trénink jednotlivých cvičení se provádí sledováním pohybujícího se textu nebo různých objektů. Proto by bylo vhodné, aby bylo cvičení zobrazené v co největší možné ploše.

### 4.1 Vývoj uživatelského rozhraní

Uživatelská rozhraní se rychle vyvíjejí. Pryč jsou doby, kdy při spuštění aplikace viděl uživatel množství různých ovládacích tlačítek, a po spuštění programu se zobrazovalo několik různých oken. V dnešní době se přikláníme k co nejjednoduššímu řešení, čím méně ovládacích prvků, tím lépe.

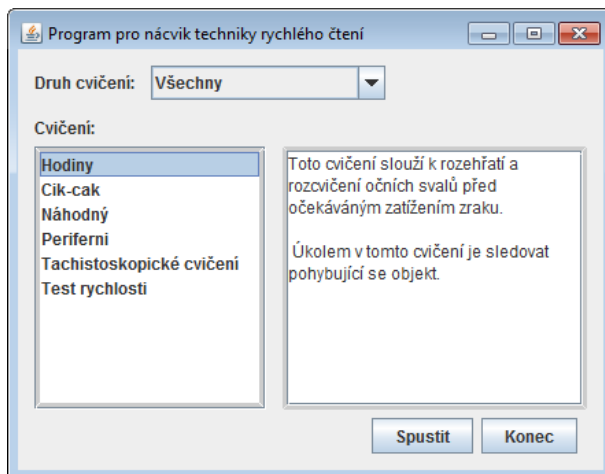
Cílem moderních uživatelských programů je vytvořit program, ve kterém se uživatel lehce zorientuje, a zároveň mu bude umožňovat lehké ovládání při splnění všech funkcí. Uživatelé mají rádi jednoduché ovládání. Nechtějí se zabývat složitým hledáním, kde se v nepřehledném menu skrývá funkce, kterou chtějí použít. Správné uživatelské rozhraní se chová přesně podle očekávání uživatele. Není vhodné vyžadovat po uživateli neustále různá nastavení nebo zobrazovat různá vyskakující dialogová okna.

Další velmi důležitý bod je zajistit, aby uživatel měl snadnou možnost navrátit různá nastavení, která mohl omylem změnit, a zapříčinil tím např. nezobrazení ovládací nabídky. Pokud uživatel není schopen tuto nabídku lehce obnovit zpět, často se může přiklonit k ukončení používání takového programu a nalezení alternativního řešení, nebo může zkusit tento program přeinstalovat s očekáváním, že se tato nabídka zobrazí.

Uživatelé chtějí pracovat s programem, ve kterém jsou ovládací prvky rozmístěné jako v jiných běžně používaných programech, jako jsou programy obsažené v kancelářském balíku Microsoft Office a webových prohlížečích jako Internet Explorer nebo Mozilla Firefox. Rozmístěním ovládacích prvků na podobná místa jako ve výše zmíněných programech zajistíme uživateli lehčí orientaci v aplikaci.

## 4.2 Počáteční návrh uživatelského rozhraní

Ze začátku jsem se zabýval spíše funkční otázkou, větší důraz jsem kladl na implementaci jednotlivých cvičení. Uživatelské rozhraní používané ze začátku tvorby této aplikace můžeme vidět na obr. 4.1. Toto rozhraní sloužilo převážně pro testování jednotlivých cvičení, jelikož nedodržovalo prvky, které již byly popsány v této kapitole pro návrh uživatelských rozhraní.



Obrázek 4.1: Počáteční návrh GUI programu

V tomto návrhu hlavní okno programu obsahovalo v levé části seznam cvičení a vpravo byl zobrazen textový popis vybraného cvičení. Ve vrchní části okna byl k dispozici comboBox pro detailnější výběr typu cvičení.

Z obr. 4.1 můžeme vidět, že toto rozhraní obsahovalo pouze komponenty obsahující prostý text a nebyly v něm žádné zachytivé orientační body v podobě ikon. Ke spuštění cvičení musel uživatel vybrat cvičení ze seznamu obsahující názvy implementovaných cvičení, a poté musel stisknout tlačítko Spustit. Po spuštění se zobrazilo nové okno s daným cvičením. Statistiky a dosažené skóre se v tomto návrhu zobrazovaly také v novém okně. Díky stále nově zobrazovaným oknům působila aplikace dosti nepřehledně, proto jsem se rozhodl toto uživatelské rozhraní přepracovat a vytvořit přívětivější a přehlednější.

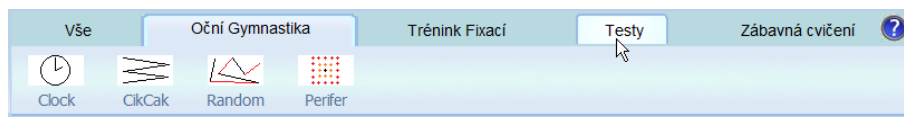
## 4.3 Finální návrh uživatelského rozhraní programu

Definováním počátečních požadavků na aplikaci a také na základě testování uživatelských rozhraní programů pro nácvik techniky rychlého čtení popsaných v kapitole 2.5, jsem navrhl uživatelské rozhraní, které splňovalo dané požadavky. Také jsem se při návrhu inspiroval prvky programů, se kterými pracují běžní uživatelé nejčastěji, a to programy z kancelářských balíků a internetovými prohlížeči.

Na základě získaných informací jsem se rozhodl vytvořit uživatelské rozhraní, které bude složeno pouze z jednoho hlavního okna, které bude obsahovat ve vrchní části přehlednou nabídku jednotlivých cvičení, zbylá část bude použita pro zobrazení jednotlivých cvičení a zobrazení statistik, popisujících rychlost čtení uživatele programu.

## Horní nabídka

Tato nabídka je zobrazena ve vrchní části okna aplikace. Obsahuje tlačítka ke spuštění jednotlivých cvičení, která jsou zároveň rozdělena do skupin podle toho, kterou oblast rychločtení procvičují. Zároveň se v této nabídce zobrazují různá nastavení pro vybraná cvičení. Tuto nabídku můžeme vidět zobrazenou na obr. 4.2.

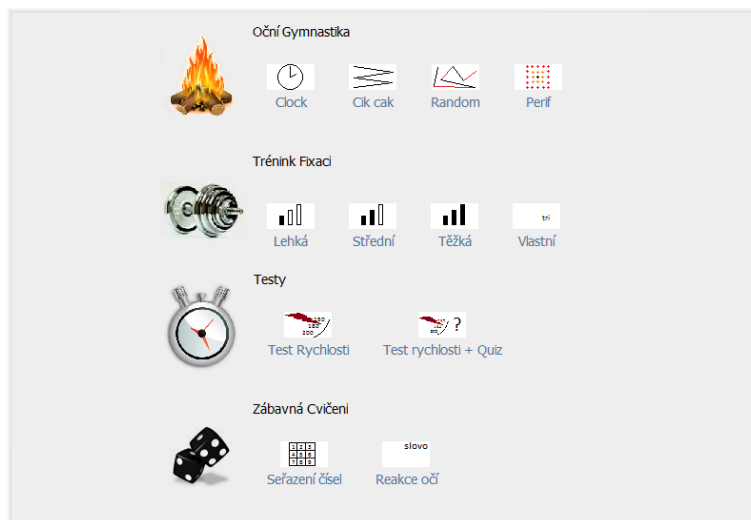


Obrázek 4.2: Horní nabídka

## Hlavní zobrazovací panel

Pod hlavní nabídkou najdeme hlavní zobrazovací panel, který vyplňuje zbytek části zobrazeného okna.

Při spuštění programu, nebo při výběru *Vše* v horní nabídce, dojde k zobrazení přehledného menu, které můžeme vidět na obr. 4.3. Toto menu je složeno z jednotlivých tlačítek sloužících k rychlému spuštění daného cvičení. Pro větší přehlednost jsou jednotlivá cvičení na stejném řádku zaměřena na jednu problematiku rychlého čtení. V levé části tohoto menu jsou zobrazeny ikony, které tématicky patří ke cvičením na daném řádku, a ulehčují orientaci v nabídce hlavního menu.



Obrázek 4.3: Hlavní nabídka

Při spuštění jakéhokoliv cvičení dojde k jeho zobrazení na místě hlavního menu. Tento panel je vycentrován vzhledem k celému oknu aplikace a má omezenou šířku, aby mohlo být simulované čtení o šířce klasického listu na obrazovce.

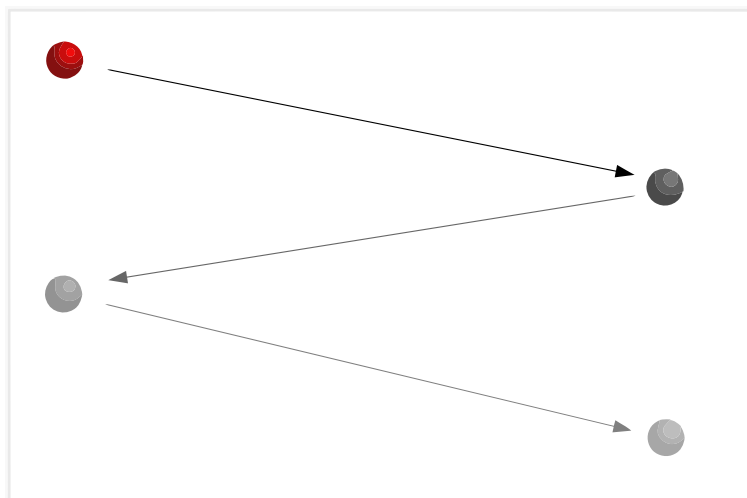
## 4.4 Výběr technik rychlého čtení

Na základě knih [3] a [4] jsem vybral oblasti dovedností zraku, které je vhodné trénovat pro zlepšení v rychločtení. Zároveň jsem se v některých cvičeních inspiroval ze cvičení v programech popsanych v kapitole 2.5. Zároveň se tyto cviky navzájem ovlivňují. Nelze tedy trénovat pouze určitou vlastnost a očekávat velká zlepšení. Chceme-li efektivně číst, musíme postupně zlepšovat všechny oblasti, na které je zaměřeno jednotlivé cvičení. V mém programu jsem implementoval cviky zaměřené na trénink následujících vlastností lidského zraku:

- Oční gymnastika
- Oční fixace
- Periferní vidění
- Reakce očí

### Oční gymnastika

Jednotlivé cviky oční gymnastiky slouží jako rozcvičení před očekávanou zátěží očí, a to např. chceme-li číst delší text nebo trénovat rychločtení. V těchto cvičeních máme za úkol sledovat pohybující se objekt pouze očima bez pohybu hlavy. Provedením dvou cvičení z oční gymnastiky se naše oči dostatečně připraví na danou zátěž.



Obrázek 4.4: Cvičení na oční gymnastiku

Pro oční gymnastiku jsem vybral 4 cvičení. V prvních třech je úkolem sledovat pohybující se objekt pouhým zrakem bez pohnutí hlavy. Pro každé z těchto cvičení platí jiná sekvence pohybů objektu. Jedno z těchto cvičení můžeme vidět na obr. 4.4. Čtvrté cvičení je zaměřeno na rozcvičení periferního vidění. V tomto cvičení máme za úkol sledovat nehybný bod uprostřed obrazovky a současně vnímat periferně vzdalující se okolní objekty.

## Oční fixace

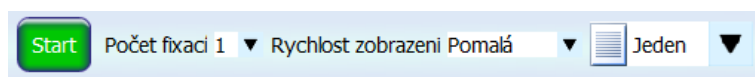
Trénink očních fixací patří mezi nejdůležitější ze všech cviků. Zároveň ale praktikování pouze tohoto cviku nám nepomůže k rychlejšímu čtení. Prováděním tohoto cvičení se postupným tréninkem zmenšuje počet fixací, které potřebujeme při čtení textu. Toto cvičení závisí z větší části také na periferním vidění, které umožňuje pojmout větší část textu.

K přesnějšímu nastavení tohoto cvičení můžeme využít panel s nastavením, který můžeme vidět na obr. 4.5, který se skládá ze tří comboBoxů, které slouží k nastavení následujících vlastností:

- Počet fixací - tímto nastavujeme, na kolik částí bude zobrazen jeden řádek textu. Zvolíme-li si jednu fixaci, zobrazí se nám celý řádek najednou. Při volbě tří fixací se rozdělí řádek na tři části, které budou postupně zobrazovány ve zvoleném časovém intervalu.
- Rychlost zobrazení - slouží k nastavení rychlosti, s jakou se bude zobrazený text měnit. Čím větší rychlost zobrazení, tím je nám text zobrazen kratší dobu.
- Počet sloupců - poslední prvek v nastavení slouží k rozdělení textu na sloupce. Výběrem určitého množství sloupců je stránka s textem rozdělena na daný počet sloupců a text je postupně zobrazován po jednotlivých sloupcích.

Pro spuštění cvičení slouží výrazné zelené tlačítko *Start* v panelu s nastavením, anebo stisknutím klávesy *mezerník* na klávesnici.

V tomto cvičení se nám zobrazují postupně části textu v určitém časovém intervalu. Provádění cvičení můžeme vidět na obr. 4.6. Na obrázku je ukázka cvičení s nastavením pro tři fixace, kdy je řádek rozdělen na tři podobně velké části, které jsou postupně zobrazeny.



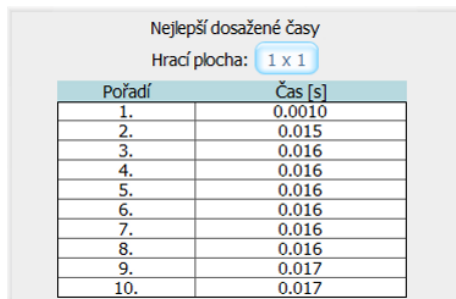
Obrázek 4.5: Nastavení cvičení pro trénink očních fixací

Karel IV., křtěný jménem Václav, se narodil v Praze jako syn Elišky Přemyslovny a Jana  
⇩  
Karel IV., křtěný jménem Václav, se narodil v Praze jako syn Elišky Přemyslovny a Jana  
⇩  
Karel IV., křtěný jménem Václav, se narodil v Praze jako syn Elišky Přemyslovny a Jana

Obrázek 4.6: Trénink fixací

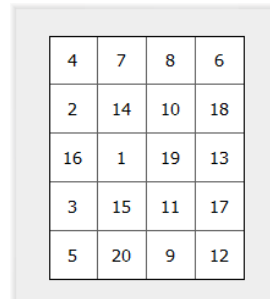
## Periferní vidění

Dalším důležitým cvičením pro trénink rychlého čtení je cvičení pro rozšíření periferního vidění. Trénink této dovednosti se promítne při čtení textu, kdy dokážeme pojmout větší množství textu najednou, a tím omezíme množství fixací nutných k přečtení textu.



Pořadí	Čas [s]
1.	0.0010
2.	0.015
3.	0.016
4.	0.016
5.	0.016
6.	0.016
7.	0.016
8.	0.016
9.	0.017
10.	0.017

Obrázek 4.7: Tabulka nejlepších časů

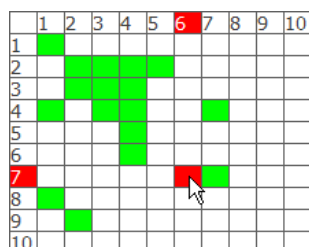


4	7	8	6
2	14	10	18
16	1	19	13
3	15	11	17
5	20	9	12

Obrázek 4.8: Seřazení čísel

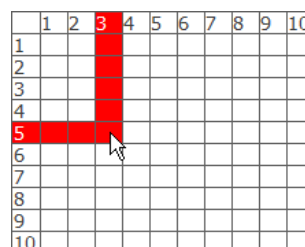
Cílem tohoto cvičení je seřadit zobrazená čísla od 1 do N v co nejkratším čase. Úkolem uživatele není zaměřit se pouze na jedno zobrazené číslo, ale musí neustále vnímat i okolní hodnoty. Před spuštěním samotného cvičení se zobrazuje tabulka s 10 nejlepšími časy, dosaženými pro jednotlivé velikosti hracího pole. Vzhled této tabulky můžeme vidět na obr. 4.7. Velikost hracího pole, pro které se má zobrazit skóre, zvolíme kliknutím na tlačítko, které se nachází nad tabulkou s výsledky nejlepších dosažených časů. Po stisknutí tohoto tlačítka se nám zobrazí nabídka, ze které si vybíráme velikost hrací plochy, pro které chceme zobrazit skóre. Tento ovládací prvek byl původně typu comboBox, ale při větším počtu hracích ploch začal být výběr nepřehledný. Proto jsem navrhl vlastní přehlednější komponentu, která je zobrazena na obr. 4.9. Zelenou barvou jsou zobrazeny velikosti hracích ploch, pro které již jsou dostupné dosažené výsledky. Bílou barvou je naopak vyznačeno hrací pole, pro které nejsou ještě k dispozici žádné zaznamenané hodnoty.

Před spuštěním samotného cvičení si ještě uživatel zvolí velikost hracího pole, které udává množství zobrazených čísel. Toto se provádí kliknutím na tlačítko *Hrací Pole*, nacházejícího se v horním menu vedle tlačítka *Start*. Výběr hracího pole je zobrazen na obr. 4.10.



	1	2	3	4	5	6	7	8	9	10
1										
2										
3										
4										
5										
6										
7										
8										
9										
10										

Obrázek 4.9: Výběr skóre pro určité hrací pole



	1	2	3	4	5	6	7	8	9	10
1										
2										
3										
4										
5										
6										
7										
8										
9										
10										

Obrázek 4.10: Výběr hracího pole

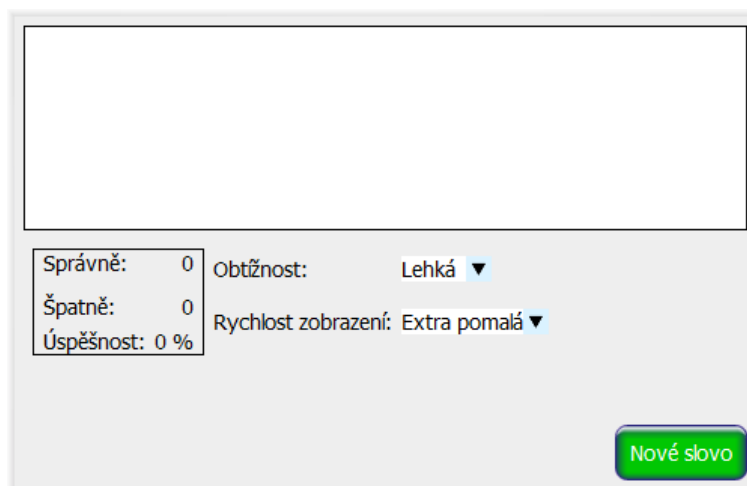
Ke spuštění cvičení dojde stiskem zeleného tlačítka *Start*, anebo stisknutím klávesy *mezerník* na klávesnici. Při spuštění se zobrazí tabulka o vybrané velikosti, která obsahuje náhodně rozmístěná čísla. Tuto tabulku můžeme vidět na obr. 4.8. Po seřazení všech zobrazených čísel je uživateli vypsán dosažený čas a zobrazeno skóre pro danou velikost hracího

pole.

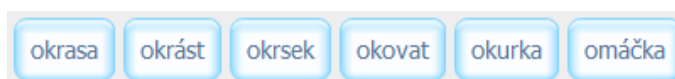
## Reakce očí

Reakce očí ovlivňuje dobu, kterou náš zrak potřebuje k zaostření na viděný objekt. Tréninkem této dovednosti budeme schopni sledovat rychleji měnící se text a přecházet z jedné fixace na další. Toto cvičení vychází z tzv. tachistoskopu. Trénink této dovednosti se provádí zobrazením slova po velmi krátký časový interval a jeho následným uhodnutím.

Grafické rozhraní tohoto cvičení obsahuje ve vrchní části panel, ve kterém se budou zobrazovat jednotlivá slova. Pod ním vlevo nalezneme statistiky udávající úspěšné a neúspěšné odpovědi. Napravo od statistik se nachází dva comboBoxy. Prvním nastavujeme velikost oblasti, ve které se bude dané slovo zobrazovat, a také délku zobrazených slov. U lehčí obtížnosti se zobrazují kratší slova v menším prostoru. U těžších obtížností se zobrazují dlouhá slova po celé šíři panelu. Druhý comboBox slouží k nastavení doby, po kterou bude zobrazené slovo viditelné. Čím větší rychlost zobrazení, tím kratší dobu bude slovo viditelné. V pravém dolním rohu se nachází zelené tlačítko *Nové Slovo*, které slouží k vygenerování a zobrazení nového slova v TachoPanelu. Vygenerování nového slova můžeme vyvolat také stiskem klávesy *mezerník* na klávesnici. Celkový vzhled cvičení můžeme vidět na obr. 4.11. Po ukázání náhodného slova (např. slova *okovat*) se zobrazí uživateli nabídka složená z pěti tlačítek, z nichž jsou 4 popsány podobnými slovy jako vygenerované slovo a páté tlačítko je popsáno zobrazeným slovem. Cílem uživatele je kliknout na správné tlačítko, aby se započítala správná odpověď. Zobrazené možnosti pro vygenerované slovo *okovat* můžeme vidět na obr. 4.12.



Obrázek 4.11: Cvičení pro trénink reakce očí



Obrázek 4.12: Možnosti pro vygenerované slovo „okovat“

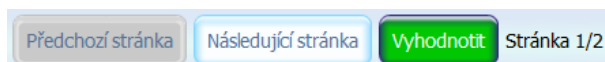
V tomto cvičení je nutná vysoká koncentrace, protože předem nevíme, na kterém místě panelu se slovo zobrazí. Zároveň máme na zareagování málo času, jelikož vygenerované slovo pouze problikne na obrazovce.



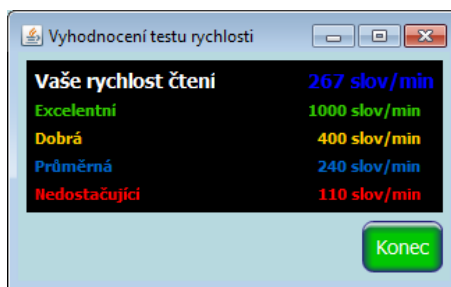
## Test rychlosti čtení a porozumění

Testy rychlosti čtení, jak už z názvu vyplývá, slouží ke změření rychlosti čtení uživatele. Zároveň může uživatel podstoupit test porozumění, ten obsahuje otázky zaměřené na čtený text. Na základě testu porozumění se může uživatel přesvědčit, kolik si toho zapamatoval. Úspěšnost průměrných čtenářů se pohybuje okolo 60-ti procent.

Při spuštění pouze testu rychlosti čtení se nám zobrazí v hlavním panelu text a začne se nám měřit čas. V horní nabídce se nachází ovládání pro toto cvičení, které obsahuje tlačítka pro otáčení stránek, a také obsahuje tlačítka pro vyhodnocení cvičení. Ovládací prvky můžeme vidět na obr. 4.13. Otáčení mezi jednotlivými stránkami můžeme simulovat také stiskem levé a pravé šipky umístěné na klávesnici. Po kliknutí na tlačítko *Vyhodnotit* se uživateli zobrazí informace o jeho dosažené rychlosti čtení, jak můžeme vidět na obr. 4.14.

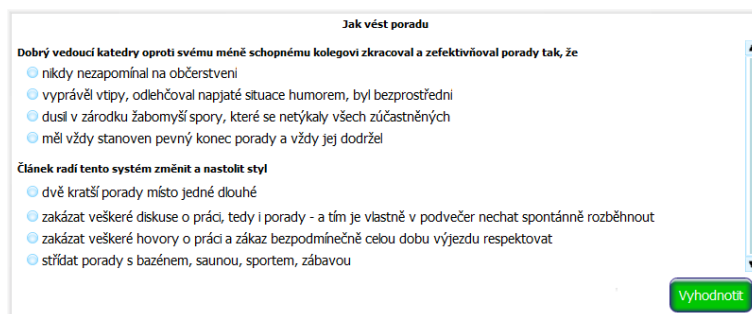


Obrázek 4.13: Ovládací tlačítka pro test rychlosti čtení



Obrázek 4.14: Vyhodnocení rychlosti čtení

Při spuštění testu rychlosti a porozumění dojde po přečtení textu místo vyhodnocení textu k zobrazení otázek na čtený text, na které má uživatel odpovědět. Vzhled tohoto testu můžeme vidět na obr. 4.15. Po zodpovězení těchto otázek a kliknutí na tlačítko *Vyhodnotit* je následně uživateli zobrazeno okno obsahující podobné informace jako na obr. 4.14, doplněné o počet a procentuální úspěšnost zodpovězených otázek.



Obrázek 4.15: Test porozumění textu

## Kapitola 5

# Implementace programu

Při tvorbě programu jsem se nejdříve zaměřil na implementaci jednotlivých cvičení, které jsem vybral pro tuto aplikaci. Výběr jednotlivých technik popisují v kapitole 4.4. Tato cvičení jsem implementoval do komponent třídy *JPanel*. Poté jsem se zaměřil na vylepšení uživatelského rozhraní celé aplikace. Cvičení vytvořená v jednotlivých panelech usnadňovala jejich začlenění do programu, kdy stačilo jenom přemístit jednotlivé panely na požadované místo v aplikaci.

### 5.1 Implementace technik rychlého čtení

Pro každou implementovanou techniku rychlého čtení jsem vytvořil vlastní třídu, která vychází z komponenty *JPanel*, a pro každý tento panel jsem implementoval funkčnost daného cvičení.

#### Oční gymnastika

Cvičení na oční gymnastiku jsou implementována v balíčku *eyegym* a jsou tvořena následujícími třídami: *Clock*, *CikCak*, *RandomMove* a *Perifer*. Každá z těchto tříd obsahuje jedno cvičení na oční gymnastiku.

Cvičení implementovaná ve třídách *Clock*, *CikCak* a *RandomMove* zobrazují jediný grafický objekt a každá z těchto tříd implementuje rozdílný pohyb tohoto objektu. Vykreslování se provádí pomocí funkce *paintComponent()*, ve které jsou vypočítávány nové souřadnice zobrazených objektů, a zároveň se tyto souřadnice kontrolují, aby nedocházelo k vykreslení objektu do části panelu, která není pro uživatele viditelná. Oproti tomu ve cvičení třídy *Perifer* je zobrazen jeden objekt ve středu obrazovky a dalších 8 objektů se od tohoto středu vzdaluje.

Pohyb objektů je simulován pomocí vlákna třídy *Thread*, které v určitém časovém intervalu volá funkci *repaint()*, která vyvolává překreslení grafického obsahu panelu voláním zmíněné funkce *paintComponent()*.

#### Trénink očních fixací

Cvičení pro trénink očních fixací je implementováno pomocí třídy *FixaceReadPan* v balíčku *tacho*. Uživatel může toto cvičení spustit v definovaných obtížnostech, anebo může zvolit vlastní nastavení. Při zvolení druhé varianty se v horní nabídce zobrazí panel s nastavením, který je implementován ve třídě *FixaceControl*.

Po spuštění tohoto cvičení, dojde nejdříve k vypočítání maximálního počtu znaků, které může textová oblast obsahovat. Následně pomocí funkce *getNameTachoFile()* třídy *XMLparser* získá název souboru, ze kterého bude načítat zobrazovaný text. Po získání názvu souboru dojde k načtení daného množství textu funkcí *readFile()* třídy *FixaceReadPan*. Při načítání posledního slova, které může být zobrazeno v textové oblasti, dochází ke kontrole, zda je načteno celé. Pokud se celé slovo do textové oblasti nevejde, je načítání textu ukončeno před tímto slovem. Také se ukládá informace o offsetu, který určuje, na jaké pozici v souboru bude funkce *readFile()* pokračovat v načítání. Následně je načtený text vložen do textové oblasti funkcí *reText()*.

Poté dojde k aktivaci vlákna třídy *Thread*, které postupně zobrazuje části jednotlivých řádků podle počtu zvolených fixací. Zvýrazňují se celá slova a délka jednotlivých fixací zobrazených pro jeden řádek je zvolena jako poměr celkového počtu znaků obsažených na řádku a počtu fixací, na které se má řádek rozdělit. Zvýraznění fixací obsluhuje funkce *setLineOffset()*, která vybírá z textové oblasti jednotlivé řádky. Ve funkci *setLineOffset()*, dochází k volání funkce *setSelectionOffsets()*, která již zobrazuje jednotlivé fixace.

Při zobrazení poslední fixace obsažené v textové oblasti dojde k načtení dalšího textu ze souboru funkcí *readFile()* a zvýrazňování fixací se opakuje, dokud funkce *readFile()* nedosáhne při načítání textu konce souboru. V tomto případě dojde k nastavení příznaku *endExercise*, který zaručuje ukončení cvičení při zobrazení poslední fixace.

## Trénink reakce očí

Toto cvičení je implementováno třídou *TachoPanGUI* v balíčku *tacho*. Po spuštění tohoto cvičení se nám zobrazí grafické rozhraní jako na obr. 4.11. V horní části okna se nachází panel třídy *TachoPanel*, ve kterém se zobrazují jednotlivá slova.

K zobrazení slova slouží zelené tlačítko *Nové slovo*, umístěné v pravém dolním rohu. K vykreslení slova dochází pomocí funkce *paintComponent()*. V této funkci dochází k volání funkce *nextWord()*, která generuje náhodný index do seznamu slov vytvořeného pro toto cvičení. Následně se generují souřadnice pro zobrazení vybraného slova funkcí *nextCoord()*, a poté dojde k zobrazení slova v *TachoPanelu*. Slovo je zobrazeno po dobu určenou *comboBoxem* pro rychlost zobrazení. Zobrazení po určitou dobu zajišťuje běžící vlákno, které po dané době vymaže obsah *TachoPanelu*.

Následně je uživateli zobrazena nabídka slov, ze kterých má vybrat to zobrazené. Po vybrání některého slova z nabídky je vyhodnoceno, zda uživatel zvolil správné slovo, a dle toho je upravena zobrazená statistika. Pokud uživatel nevybral slovo ze zobrazené nabídky a rovnou nechal vygenerovat další slovo, je automaticky přičítána špatná odpověď.

## Trénink periferního vidění

Trénink periferního vidění je implementován pomocí tříd obsažených v balíčku *order*. Při výběru tohoto cvičení se nejdříve zobrazí tabulka s dosaženými výsledky pro vybranou hrací plochu. Tato tabulka je implementována ve třídě *OrderStatPan*, která zároveň obsahuje také komponentu třídy *SkoreChooser*, která slouží k vybrání skóre pro určitou hrací plochu. Po vybrání velikosti hrací plochy, pro kterou chceme zobrazit skóre, dojde k volání funkce *actualizeScore()*, která aktualizuje hodnoty dosažených časů zobrazených v tabulce pro danou hrací plochu.

Ovládací panel cvičení slouží k výběru hrací plochy a spuštění tréninku. Toto ovládání je tvořeno třídou *OrderControl*. Pro výběr hrací plochy, pro kterou se bude cvičení provádět, slouží komponenta třídy *SizeChooser*.

Po spuštění tréninku se v hlavním panelu zobrazí tabulka čísel, která je prezentována třídou *OrderPan*. Velikost této tabulky je dána volbou uživatele dle vybrané velikosti hracího pole. Na základě toho je vytvořeno pole o posloupnosti 1-N čísel. A tato čísla jsou postupně pomocí funkce *Math.Random()* náhodně rozmístěna v tabulce. Poté má za úkol uživatel co nejrychleji pomocí kliknutí vybrat postupně následující čísla od hodnoty 1. Po vybrání posledního čísla dojde k zobrazení statistiky pro zvolené hrací pole a také je pod tímto skóre vypsána uživatelem dosažená hodnota. Pokud jeho čas byl dostatečně dobrý pro zápis do tabulky, je tato hodnota v tabulce zvýrazněna.

## Test rychlosti čtení a porozumění textu

Test rychlosti čtení je tvořen třídou *SpeedTestPan*. Jedná se o panel, který obsahuje komponentu třídy *JTextPane*, ve kterém se zobrazuje načtený text ze souboru.

Při spuštění tohoto cvičení dojde k zavolání funkce *reText()*, která volá funkci *initCountChars()* pro výpočet počtu znaků, které mohou být zobrazeny v *JTextPane*. Následně se volá funkce *readFile()*, která načte daný počet znaků ze souboru. Tato funkce zároveň kontroluje, aby nedocházelo při načítání k rozdělení jednoho slova mezi dvě stránky. Pokud dojde k situaci, že by poslední slovo na stránce nebylo zobrazeno celé, je toto slovo zobrazeno na další stránce. Třída *SpeedTestPan* si uchovává index na poslední načtený znak ze souboru v proměnné *actualOffset*. Název souboru, ze kterého se text čte, je obsažen v proměnné *speedFiles* ve třídě *XMLparser*. Tato proměnná obsahuje všechny názvy souborů používané pro testy rychlosti. Hodnota této proměnné je inicializována při spuštění programu z dat obsažených v XML souboru s nastavením v elementu `<speedFile>`. Zároveň si program uchovává index do seznamu na poslední čtený soubor, aby nedocházelo k opakování testů při dalším jeho spuštění.

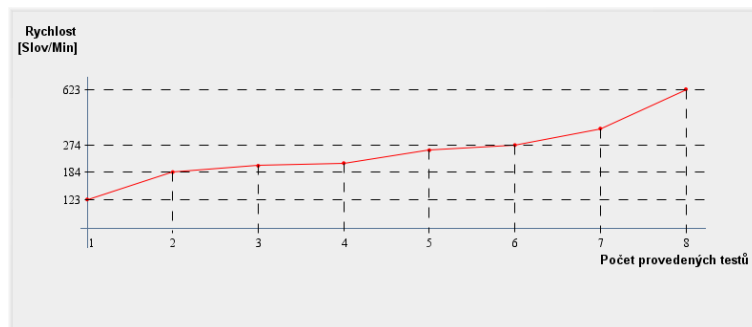
Cvičení se ovládá pomocí ovládacího panelu v horní nabídce. Tento panel je realizován třídou *SpeedControlPan* v případě spuštění cvičení s testem porozumění, anebo třídou *SpeedTestControlPan*, je-li spuštěn pouze test rychlosti. V tomto případě je místo tlačítka *Pokračovat* zobrazeno rovnou tlačítko *Vyhodnotit*. Po jeho stisku se uživateli zobrazí dialogové okno implementované třídou *EvalTest*, kterému jsou předány hodnoty počtu přečtených slov a doba, za kterou byl tento text přečten. V tomto okně je uživateli zobrazena jeho dosažená rychlost čtení, která se vypočítává podílem počtu přečtených slov a dobou, za jakou tento text přečetl.

Při spuštění testu rychlosti čtení zároveň s testem porozumění textu je v ovládaní zobrazeno tlačítko *Pokračovat*. Při stisku tohoto tlačítka dojde k zobrazení otázek zaměřených na daný text. Tento formulář je implementován ve třídě *QuizPan*. Po výběru odpovědí a stisknutí tlačítka *Vyhodnotit* se nám zobrazí dialogové okno implementované třídou *EvalQuiz*, které obsahuje kromě dosažené rychlosti čtení také informace o počtu správných a špatných odpovědí a procentuálním vyhodnocení úspěšnosti těchto odpovědí.

## 5.2 Statistiky

V této aplikaci se ukládají dosažené hodnoty v testu rychlosti čtení a ve cvičení pro trénink periferního vidění. Zaznamenané hodnoty rychlosti čtení jsou v programu zobrazeny v grafu. Na ose *X* grafu je vyneseno počet provedených testů a na ose *Y* se nachází dosažené hodnoty rychlosti čtení.

Graf je realizován třídou *SpeedStatPan*, ve které jsou pomocí funkce *paintComponent()*



Obrázek 5.1: Graf vývoje rychlosti čtení uživatele

vykresleny a propojeny jednotlivé hodnoty grafu. Z důvodu možného velkého rozptylu hodnot rychlosti čtení, které jsou vyneseny na osu  $Y$ , je tato osa logaritmická.

Hodnoty zaznamenané ve cvičení pro trénink periferního vidění jsou zobrazeny v tabulce. Tato tabulka je detailněji popsána na začátku kapitoly 5.1. Ukládání a získávání hodnot pro tyto statistiky je popsáno v následující kapitole 5.3.

### 5.3 Uchování dat

K uchování dat slouží XML soubor obsahující data strukturovaná pomocí aplikačního rozhraní DOM. Při prvním spuštění aplikace dojde k načtení dat ze souboru *settings.xml*, který se nachází v JAR souboru programu. Po prvním spuštění, pokud je to možné, dojde k vytvoření souboru *read\_settings.xml* v domovském adresáři uživatele, do kterého se ukládají hodnoty statistik a informace o provedených testech. Pokud aplikace při spuštění nalezne soubor *read\_settings.xml*, použije automaticky data obsažená v něm, místo v souboru *settings.xml*.

K načítání a ukládání souborů slouží vytvořená třída pojmenovaná *XMLParser*. Proměnná této třídy vzniká při spuštění programu a pomocí funkce *initValuesFromXML()* inicializuje hodnoty ze souboru *settings.xml*, anebo ze souboru *read\_settings.xml*, je-li k dispozici. Druhou důležitou funkcí, která je obsažena v této třídě, je funkce *writeXML()*, která slouží k ukládání aktuálních dat z programu do XML souboru *read\_settings.xml*.

#### Popis elementů v XML souboru

##### Score

Element *<score>* slouží k ukládání informací o dosažených časech ze cvičení na periferní vidění. Obsahuje atributy  $x$  a  $y$ , které udávají, pro jaké hrací pole daný čas patří.

##### Speed

Element *<speed>* obsahuje hodnoty rychlosti čtení uživatele. Tyto hodnoty jsou zaznamenávány po dokončení testu rychlosti čtení.

##### SpeedFile

Elementy *<speedFile>* obsahují názvy souborů, které obsahují text pro testy rychlosti čtení.

## TachoFile

Element `<tachoFile>` obsahuje názvy souborů, které obsahují texty pro cvičení zaměřené na trénink očních fixací.

## 5.4 Kompilace a spuštění programu

Pro kompilaci programu je nutné mít nainstalováno vývojové prostředí Sun Java Development Kit (JDK)<sup>1</sup>, které je vyvíjeno firmou Sun Microsystems. Po instalaci tohoto prostředí budeme mít k dispozici knihovny používané jazykem Java, a rovněž nám budou nainstalovány programy potřebné pro kompilaci a spuštění programů v Javě. Součástí JDK je i Java Runtime Environment (JRE). JRE je prostředí, které slouží ke spuštění zkompileovaných aplikací. Bez JRE by nebylo možné přeložené programy spustit.

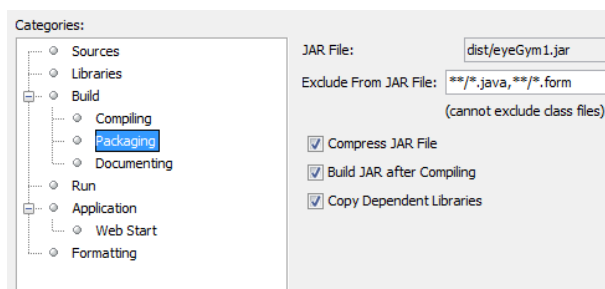
### Spuštění pomocí Java Web Start

Máme-li přístup na internet, je nejrychlejší způsob pro spuštění aplikace využít technologii JWS. Zároveň musíme mít nainstalovánu podporu javy JRE, která slouží ke spuštění aplikací v javě. Z internetové adresy <http://www.stud.fit.vutbr.cz/~xpadys03/java/launch.jnlp> si stáhneme soubor `launch.jnlp` a uložíme ho na disk. Poté v příkazové řádce pomocí příkazu `javaws launch.jnlp` dojde ke stažení a spuštění aplikace.

### Kompilace programu pomocí NetBeans IDE

Chceme-li, aby při překlada programu pomocí NetBeans IDE byl vytvořen JAR soubor, který poté můžeme použít ke spuštění programu, musíme tuto vlastnost nastavit v překladači. Toto nastavíme, když v menu *Run* zvolíme *Set Project Configuration*, a v tomto submenu zvolíme *Customize*. Zobrazí se nám dialogové okno, ve kterém můžeme nastavit různé vlastnosti překladače. Nás zajímá oblast v nabídce *Build*→*Packaging*, kde zaškrtneme *Build JAR after Compiling*.

Po tomto nastavení, už můžeme program přeložit a spustit v NetBeans IDE pomocí klávesy F6, anebo kliknutí v menu *Run* na položku *Run Project*.



Obrázek 5.2: Nastavení překladače v NetBeans IDE

<sup>1</sup>Balíček JDK můžeme stáhnout z internetové stránky <http://www.oracle.com/technetwork/java/javase/downloads/index.html>

## Spuštění pomocí JAR souboru

Zkompilujeme-li pomocí NetBeans IDE zdrojový kód programu, vytvoří se ve složce se zdrojovým kódem adresář *Dist*, který obsahuje vytvořený JAR souboru této aplikace. Máme-li k dispozici tento soubor, můžeme poté program spustit kliknutím na něj.

## 5.5 Testování programu

Neméně důležitou částí tvorby programu bylo otestování aplikace uživateli. Jelikož pro zjištění účinnosti programu na rychlost čtení uživatelů by bylo zapotřebí, aby tito uživatelé trénovali pravidelně pomocí této aplikace po dobu minimálně jednoho měsíce, kdy by už mělo být viditelné určité zlepšení, bylo testování aplikace zaměřeno převážně na práci s tímto programem.

Aplikace byla dána k testování 20 uživatelům. Jednalo se o studenty různých vysokých škol ve věku mezi 20-25 lety. K tomuto účelu jsem vytvořil dva dotazníky, které byly těmto uživatelům dány k vyplnění. První dotazník vyplnili před prvním spuštěním aplikace a byl zaměřen na jejich přístup k literatuře a problematice rychločtení. Druhý dotazník byl zodpovězen uživateli až po vyzkoušení vytvořeného programu a obsahoval otázky zaměřené na ovládání programu, rozmístění ovládacích prvků a smysluplnost jednotlivých cvičení.

### Dotazník před spuštěním programu

Tento dotazník vyplnili uživatelé ještě před tím, než jim byla dána aplikace k testování. Dotazník obsahoval následujících 7 otázek zaměřených na jejich zkušenosti práce s počítačem a vztahu ke čtení knih.

1. Jakého jste pohlaví?
  - 1) Muž
  - 1) Žena
2. Jaký typ textu převážně čtete?
  - 1) Tištěný text
  - 2) Elektronický text
3. Raději čtete text
  - 1) Tištěný
  - 2) Elektronický
4. Kolik knih přečtete za měsíc?
  - 1) 1 knihu a méně
  - 2) 2 knihy
  - 3) 3 knihy
  - 4) 4 knihy a více
5. Zaměření vysoké školy, kterou studujete je
  - 1) Ekonomická
  - 2) Technická
  - 3) Lékařská
  - 4) Pedagogická
  - 5) Filozofická
  - 6) Umělecká
  - 7) Humanitní
6. Jaké jsou vaše zkušenosti s počítačem?
  - 1) Začátečník - spouštění aplikací, prohlížení www stránek
  - 2) Běžný uživatel - instalování jednoduchých aplikací
  - 3) Pokročilý uživatel - instalace operačního systému

#### 4) Zkušený uživatel - programování aplikací

#### 7. Chtěli byste umět rychleji číst?

- 1) Ano
- 2) Ne

Z 20 dotazovaných bylo 80 % mužů, z nichž 70 % uvedlo vysokou školu zaměřenou technickým směrem, 20 % mužů zvolilo filozofickou fakultu a zbylých 10 % mužů lékařskou fakultu. Z žen studovala vysokou školu technického zaměření pouze jedna, dvě navštěvovali lékařskou fakultu a jedna studovala filozofickou fakultu. Zaměření vysoké školy se projevilo v 6. otázce, kde 90 % mužů definovalo svou úroveň práce s počítačem jako Pokročilí nebo Zkušení uživatelé. Naopak u žen byla nejčastější odpovědí na tuto otázku Běžný uživatel.

Z odpovědí na otázky ohledně čtení knih vyplynulo, že studenti netechnických škol přečtou průměrně 3 až 4 knihy za měsíc, oproti studentům technických škol, kteří přečtou 1 až 2 knihy. Na druhou otázku dotazníku odpovědělo 80 % dotazovaných studentů, že čtou převážně elektronický text. Zároveň ale ze třetí otázky vyplynula jednoznačně shodou všech dotazovaných větší záliba ve čtení tištěných knih. V poslední otázce uvedlo 90 % studentů zájem o zlepšení své rychlosti čtení.

### Dotazník po otestování aplikace

Druhý dotazník byl uživatelům dán k vyplnění až po jejich otestování programu. Otázky v tomto dotazníku byly již zaměřeny na ovládání programu, grafický vzhled aplikace a funkčnost jednotlivých cvičení.

První dvě otázky dotazníku se týkaly spuštění aplikace. Program uživatelé spouštěli pomocí JWS (popsáno v kapitole 3.4). Z dotazníku vyplývá, že spuštění aplikace proběhlo v pořádku ve všech případech. Z 20 dotázaných uživatelů tuto aplikaci spouštělo 10 uživatelů pod operačním systémem MS Windows XP, 6 uživatelů pod MS Windows 7. Dva uživatelé spouštěli tuto aplikaci pod OS Fedora 16 a 2 uživatelé pod OS Ubuntu 11.04.

Následovaly otázky 3 a 4, které byly zaměřeny na přehlednost hlavního menu, které se zobrazí po spuštění aplikace a zároveň na orientování v horní nabídce, obsahující zobrazená cvičení pro vybranou oblast tréninku rychločtení. V hlavním menu si uživatelé chválili ikony, které rozřazují jednotlivá cvičení do skupin zaměřených na danou oblast zraku. Horní nabídka byla hodnocena v 80 % jako jednoduchá a přehledná. Následovaly otázky na přehlednost, nastavení a funkčnost jednotlivých cvičení.

Otázka č. 5 sloužila k získání názoru uživatelů na cvičení zaměřená na oční gymnastiku. 60 % dotázaných by si přálo více cvičení na tuto problematiku, jelikož po několikátém spuštění tohoto programu už tyto cviky znali a moc je nezajímaly.

Otázky 6 a 7 byly zaměřeny na cvičení očních fixací. Zde uvedlo 40 % uživatelů, že využívali hlavně přednastavené obtížnosti. Zbylých 60 % spouštělo cvičení převážně pomocí vlastního nastavení. Vlastnosti v panelu s nastavením v tomto cvičení pochopilo ihned 80 % dotázaných, zbylých 20 % uvedlo, že pochopili jednotlivé prvky po prvním spuštění tohoto cvičení.

Následovaly otázky 7, 8 a 9, které se ptaly na Testy rychlosti. Zobrazení grafu rychlosti uvedli uživatelé jako důležitý prvek při používání programu. Líbila se jim možná kontrola svého zlepšení. Zároveň u tohoto grafu uvedlo 20 % uživatelů, že by si přálo podrobnější statistiku ohledně jejich tréninku. U testu porozumění textu by 35 % uživatelů uvítalo větší množství otázek.



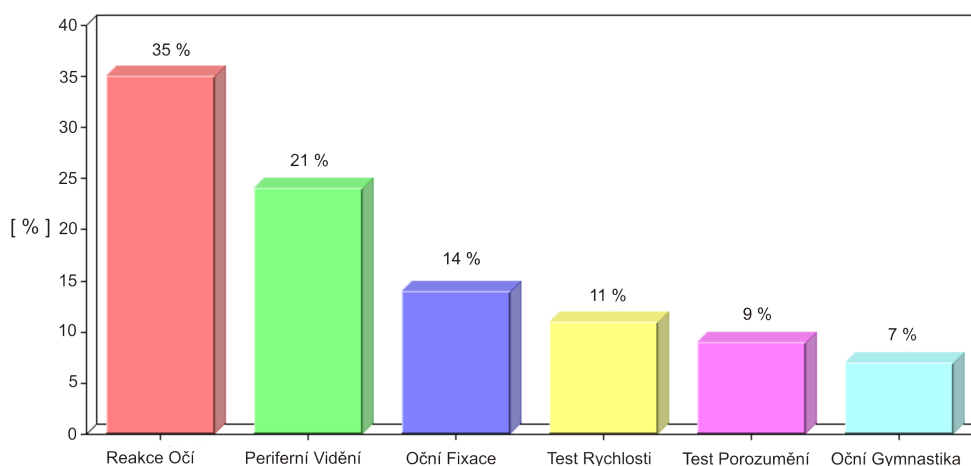
Další dvě otázky byly zaměřeny na trénink periferního vidění. Toto cvičení vyhodnotili uživatelé jako nejpřehlednější, protože neobsahuje množství prvků, které by musel uživatel nastavit.

Poslední dvě otázky zaměřené na konkrétní cvičení se týkaly tréninku reakce očí. U tohoto cvičení uvedlo 70 % uživatelů, že pochopili nastavení jednotlivých prvků a provádění cvičení během 15 sekund, 20 % uživatelů pochopilo jednotlivá nastavení do půl minuty a zbylí uživatelé se v tomto cvičení zorientovali do jedné minuty. Uživatelé uvítali výběr různých obtížností, které určovaly délku slova a velikost panelu, do kterého se jednotlivá slova zobrazovala.

Poté následovaly dvě otázky pro vyhodnocení nejzábavnějšího cvičení. U této otázky uvedli uživatelé pouze dvě cvičení. Za nejzábavnější cvičení považovalo 60 % uživatelů trénink reakce očí. Zbýlých 40 % uživatelů hlasovalo pro trénink periferního vidění. Pro ostatní cvičení nehlasoval ani jeden uživatel. Tato dvě cvičení zaujala uživatele nejvíce a měli je spuštěna po většinu času běhu aplikace.

Oproti tomu jako nejméně zábavné cvičení vyhodnotili uživatelé cvičení pro trénink očních fixací, kdy provedení jednoho cvičení vyžadovalo delší dobu koncentrace a uživatele přestalo rychle bavit.

Testovací verze aplikace obsahovala navíc také čítač spuštění jednotlivých cvičení, který ukládal do textového souboru počet spuštění každého cvičení. Po otestování aplikace jsem obdržel od všech testerů tyto soubory a získal jsem průměrné hodnoty spuštění jednotlivých cvičení. Na grafu 5.3 můžeme vidět, jak často byla jednotlivá cvičení spouštěna.



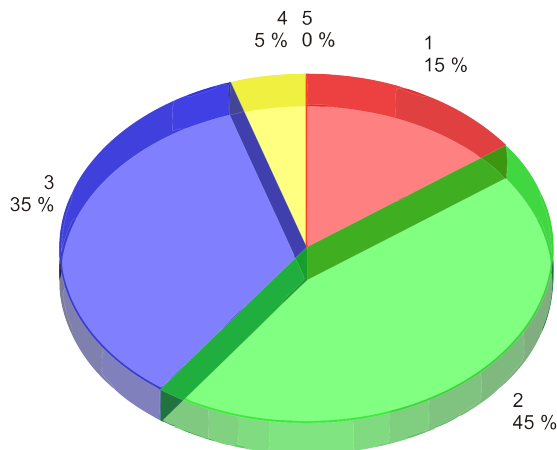
Obrázek 5.3: Podíl spuštění jednotlivých cvičení během testování

Dále mohli uživatelé napsat, co by v této aplikaci ještě chtěli. Zde odpovědělo 85% uživatelů, že by uvítalo více zábavných cvičení i na další vlastnosti zraku, než jen na periferní vidění a reakci očí. Dále by 40 % uživatelů uvítalo více grafických cvičení, kde se nepracuje pouze s textem.

Následovala otázka ohledně vzhledu aplikace, kde 70 % uživatelů ocenilo jednoduchý vzhled aplikace, který je nerušil od koncentrace při provádění jednotlivých cvičení a 20 % uživatelů odpovědělo, že je vzhled nijak nezaujal. Dále by 40 % uživatelů uvítalo propracovanější vzhled jednotlivých cvičení.

Zvýraznění důležitých tlačítek pro ovládání aplikace zelenou barvou uvítalo 85 % uživatelů, kteří uvedli, že jim usnadnilo orientaci v jednotlivých cvičeních.

Poté uživatelé ohodnotili celkový dojem aplikace a její funkčnost. Tuto aplikaci ohodnotilo 60 % jako zábavnou a snadnou pro zorientování v ovládnání a 10 % uživatelů uvedlo aplikaci jako nudnou a nezajímavou. Nakonec aplikaci měli dotázaní ohodnotit na stupnici 1-5, kde 1 je nejlepší. Výsledky hodnocení můžeme vidět na grafu 5.4.



Obrázek 5.4: Hodnocení aplikace uživateli

Během testování aplikace uživateli jsem na základě jejich připomínek upravil obtížnost některých cvičení a přidal širší možnosti nastavení cvičení. Zároveň jsem změnil některé ovládací prvky programu, které uváděli jako nepřehledné a zbytečně složité. Ve verzi aplikace s těmito novými prvky hodnotili uživatelé ovládnání v daných cvičeních jako rychlejší a jednodušší.

### Návrhy na zlepšení programu

Na základě vyhodnocení dotazníků a připomínek uživatelů bych navrhl následující vylepšení aplikace:

- navrhnout a implementovat více zábavných cvičení, ve kterých dosahuje uživatel určitého skóre.
- vytvořit cvičení, které by umožnilo porovnávat reakce více uživatelů mezi sebou pomocí počítačové sítě.
- přidat možnost sdílení dosažených výsledků mezi dalšími uživateli, globální žebříček nejlepších čtenářů používajících tuto aplikaci.
- vytvořit podrobnější statistiku pro všechna cvičení. V této statistice by byly informace o přesné době provádění jednotlivých cvičení, celkové době tréninku rychločtení a také informace o dosažených hodnotách v jednotlivých cvičení (rychlost čtení, počet správných a špatných odpovědí ze cvičení na trénink reakce očí a při jakém nastavení cvičení bylo těchto hodnot dosaženo).  
a celkovou dobu provádění všech cvičení.

# Kapitola 6

## Závěr

V rámci mé bakalářské práce jsem vytvořil program pro nácvik techniky rychlého čtení.

Cvičení použita pro nácvik této techniky jsem čerpal z knih [3], [4] a [6] a také jsem se inspiroval z existujících programů na tuto problematiku popsanych v kapitole 2.5. Cvičení, která jsem se rozhodl implementovat v mé aplikaci, jsou popsána v kapitole 4.4.

Před tvorbou samotné aplikace jsem si musel zvolit programovací jazyk a vývojové prostředí, ve kterém bych program implementoval. Vybíral jsem si mezi vývojovým prostředím Qt Creator pro jazyk C++ a vývojovými prostředími NetBeans IDE a Eclipse pro jazyk Java. Nakonec jsem zvolil jazyk java a vývojové prostředí NetBeans IDE. Výběr implementačního jazyka a vývojového prostředí je detailněji popsán v kapitole 3.

Po výběru implementační platformy jsem vyzkoušel a popsal 3 vybrané existující programy pro trénink rychlého čtení, které jsem podrobněji popsal v kapitole 2.5.

Následně jsem implementoval jednotlivá vybraná cvičení pro nácvik rychlého čtení. Popis implementace těchto cvičení se nachází v kapitole 5.1. Tato jednotlivá cvičení jsem následně implementoval do jednotné aplikace, jejíž návrh uživatelského rozhraní jsem popsal v kapitole 4.3.

Na závěr jsem takto vytvořený program dal otestovat dvaceti uživatelům. Během testování aplikace jsem na základě jejich připomínek upravil obtížnost některých cvičení a změnil některé ovládací prvky, které byly zbytečně složité. Průběh testování aplikace je popsán v kapitole 5.5. Po otestování aplikace uživatelé vyplnili dotazník s otázkami zaměřenými na ovládání a vzhled programu a smysluplnost jednotlivých cvičení. Na základě informací obdržených po vyhodnocení dotazníků jsem zjistil, že by 85 % uživatelů uvítalo více zábavných cvičení. Dále by bylo vhodné implementovat podrobnější statistiky, ze kterých by mohl uživatel analyzovat jak dlouho a kdy se daným cvičením věnoval. Také by uživatelé ocenili možnost soupeření s ostatními uživateli používající tuto aplikaci přes internet.

# Literatura

- [1] Walter B. Pitkin: *Art of Rapid Reading*. Grosset and Dunlap, 1928.
- [2] Wolfgang Zielke: *Jak číst rychleji a lépe*. Svoboda, 1988, ISBN 25-066-88.
- [3] Howard Stephen Berg: *Super Reading Secrets*. Grand Central Publishing, 1992, ISBN 04-463-6299-3.
- [4] Howard Stephen Berg: *Speed Reading the Easy Way*. Barron's Educational Series, 1998, ISBN 08-120-9852-8.
- [5] Mauro Marinilli: *Java Deployment with JNLP and Webstart*. Sams, 2001, ISBN 0-672-32182-3.
- [6] Tony Buzan: *The speed reading book*. BBC Active, 2003, ISBN 05-634-8702-X.

## Odkazy na existující programová řešení

- [7] Rocket Reader. [online]. [cit. 2012-04-24]. Dostupné z:  
<http://www.rocketreader.com/download/RocketReaderDownload.html>.
- [8] The Readers's Edge. [online]. [cit. 2012-04-24]. Dostupné z:  
<http://www.readfaster.com/tre/>.
- [9] Ace Reader Pro. [online]. [cit. 2012-04-24]. Dostupné z:  
<http://www.acereader.com/personal/free-trial.html>.

# Příloha A

## Obsah CD

- Zdrojové kódy (složka program)
- Aplikace pro nácvik techniky rychlého čtení (složka dist)
- Zdrojové kódy teoretické části v Latexu (složka doc)
- bakalarska\_prace.pdf
- plakat7016x4961.jpg
- plakat7016x4961.pdf
- Netbeans 7.1.2 (instalační soubor)
- Java SE Development Kit 7 Update 4 (instalační soubor)