Gymnázium, Praha 6, Arabská 14
předmět Programování, vyučující Mgr. Jan Lána

Reakční a Paměťový test

ročníkový projekt

Téma: test reakční doby a paměti člověka

Jan Sváček, 1.E duben 2022

Čestné prohlášení:

Prohlašuji, že jsem jediným autorem tohoto projektu, všechny citace jsou řádně označené a všechna použitá literatura a další zdroje jsou v práci uvedené. Tímto dle zákona 121/2000 Sb. (tzv. Autorský zákon) ve znění pozdějších předpisů uděluji bezúplatně škole Gymnázium, Praha 6, Arabská 14 oprávnění k výkonu práva na rozmnožování díla (§ 13) a práva na sdělování díla veřejnosti (§ 18) na dobu časově neomezenou a bez omezení územního rozsahu.

V Praze dne …….. .....

Anotace(Čeština):

V rámci tohoto ročníkového projektu je předkládán program v programovacím jazyku Java, který uživateli umožňuje otestovat svoji reakční dobu a paměť. Program také využívá grafické rozhraní JavaFX.

Abstract(English):

The subject of this coursework is software developed in the Java programming language, which allows the user to test their reaction time and memory. The program also use the JavaFX user interface framework.

Zadání ročníkového projektu

Na začátku programu si user bude moci vybrat z možností memory test a reaction test. Reaction test mu bude počítat jeho reakční dobu a nakonec mu vypočítá průměrnou reakční dobu a memory test se bude skládat z pamatování si číselných řad, když user číslo uhádne bude moci hádat dál a číslo se vždy zvětší o jedno místo.

Platforma:

- Java
- JavaFX

Obsah

Úvod

Tento ročníkový projekt umožňuje uživateli otestovat svoji reakční dobu a paměť číselných řad.

Inspirovala mě stránka Human Benchmark, kde je mnoho miniher s možností otestovat svoje dovednosti. Vybral jsem si test reakční doby a test paměti, protože v porovnání s ostatními přijdou nejzajímavější.

Pro napsání programu jsem použil programovací jazyk Java, s grafickým rozhraním JavaFX a vývojové prostředí Eclipse.

1 Vývojové prostředí

1.1 Java

“Je to objektově orientovaný programovací jazyk. Vyvinula jej firma Sun Microsystems a představila jej 23. května 1995. Současným vlastníkem oficiální implementace je firma Oracle. Stáhnou Je využíván například společnostmi Google a Android při tvorbě open source mobilního operačního systému.”[Java]

Duke - Java Maskot

1.2 JavaFX

“Je softwarová platforma postavená na bázi platformy Java a slouží pro vývoj RIA aplikací(Rich internet applications).”[JavaFX]

1.3 JavaFX Scene Builder

“Pomocí tohoto nástroje je možné navrhovat uživatelské rozhraní stylem „táhni a pusť“. Výstup je ukládán do souboru typu FXML (nástavba XML).”[JavaFX Scene Builder]

1.4 Eclipse IDE

“Je to vývojové prostředí určené pro programování v jazyce Java. Oproti ostatním vývojovým prostředím v Javě Eclipse využívá pluginů pro jeho rozšíření. To je důvodem, proč je tato platforma velice flexibilní a momentálně také nejvíce používaná.”[Eclipse IDE]

obrázek 2 - logo platformy eclipse

2 Lidské schopnosti

2.1 Reakční doba

“Je to čas, který je třeba k odezvě organismu. Přestože si je člověk schopen uvědomit i signály trvající zhruba 0,02 s, jeho reakce je pomalejší. Dobu reakce měří reaktometrie. Člověk si může zjistit svoji reakční dobu pomocí mého projektu. Také ji může trénovat a eventuálně si zlepšit svoje reflexy.”[Reakční doba]

2.2 Paměť

“Paměť je schopnost centrální nervové soustavy uchovávat a používat informace o předchozích zkušenostech. Paměť se dále dělí na senzorickou, krátkodobou a dlouhodobou. Můj program vám umožní otestovat senzorickou a krátkodobou paměť.”[Paměť]

2.2.1 Senzorická paměť

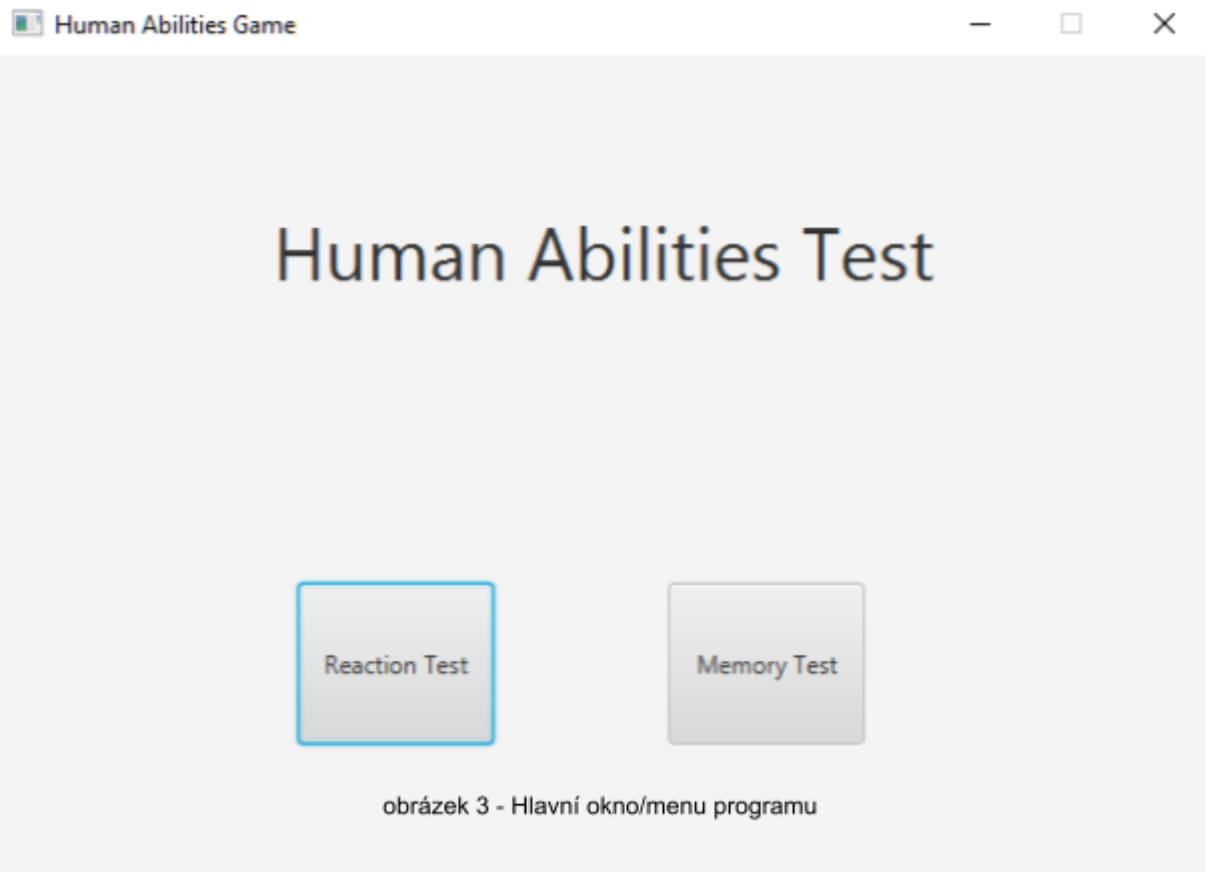
“Zjednodušeně ji lze nazvat ultrakrátká paměť. Je to část paměti, která uchovává informace přicházející ze smyslů Ty jsou podrženy po dobu nezbytně nutnou ke zpracování a rozhodnutí, zda jsou informace důležité, tedy vhodné k dalšímu zpracování či nikoliv. Pokud ano, postupují dále do krátkodobé či dlouhodobé paměti.”[Senzorická paměť]

2.2.2 Krátkodobá paměť

“Krátkodobá paměť je vědomá aktivní část paměti, ve které se odehrává většina psychických procesů (např. řešení aktuálních problémů). Zpracovávají se v ní informace dodané již zmiňovanou senzorickou pamětí a informace vyvolané z paměti dlouhodobé. Krátkodobá paměť je omezena na 5–9 prvků, které při zamezení opakování uchová na 15–20 sekund. Kapacitu lze zvýšit spojováním prvků do logických celků např. mnemotechnické pomůcky. Pro zachování informace v krátkodobé paměti je třeba si informaci opakovat tzv. fonologická smyčka(dočasně ukládá zvukové a řečové informace), jinak je paměťová stopa nenávratně ztracena.”[Krátkodobá paměť]

3 Architektura & Ovládání programu

3.1 Grafické rozhraní



obrázek 3 - Hlavní okno/menu programu

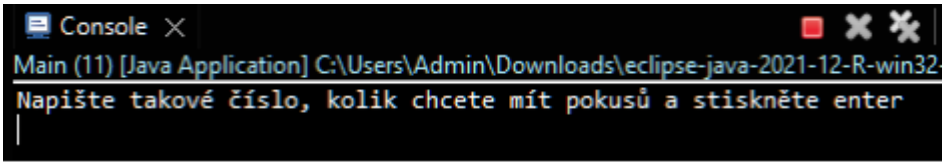
Grafické rozhraní bylo vytvořeno pomocí JavaFX Scene Builder, jehož výstup je ukládán do dílčích FXML dokumentů. Hlavní okno nabízí uživateli, jestli si chce zahrát memory test nebo reaction test a to tak, že na jedno z tlačítek(viz. obrázek 3) klikne.

3.1.1 Reaction test

Po kliknutí na tlačítko reaction test v hlavním okně/menu se spustí toto okno(viz. obrázek 4).

obrázek 4 - okno reaction testu

Pokud je uživatel připravený začít může kliknout na tlačítko start, nebo se vrátit zpátky do hlavního okna/menu stisknutím tlačítka menu. Pokud klikne na tlačítko start, tak si může v konzoli zvolit kolik pokusů chce zkusit a potvrdit to stisknutím klávesy enter.

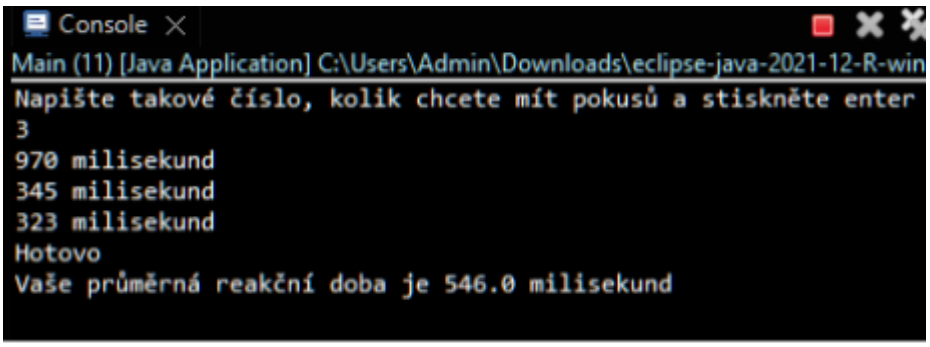


Obrázek 5 - počet pokusů v konzoli

Poté se po náhodně vygenerovaném čase objeví nové okno(viz. obrázek 5), na které uživatel musí zareagovat tím, že klikne na tlačítko click(viz. obrázek 5). Po kliknutí na tlačítko click(viz. obrázek 6) se vypíše do konzole reakční doba uživatele(viz. obrázek 7) a následně se objeví okno s tlačítkem continue(viz. obrázek 8), po jehož stisknutí se znova vygeneruje náhodný čas a poté se znovu zobrazí tlačítko click(viz. obrázek 6) Proces se opakuje tolikrát, kolik uživatel zadal počet pokusů na začátku do

konzole(viz. obrázek 5). Jestliže uživatel splní určený počet pokusů vypíše se do konzole jeho průměrná reakční doba(viz. obrázek 7) a reaction test skončí.

obrázek 6 - tlačítko click

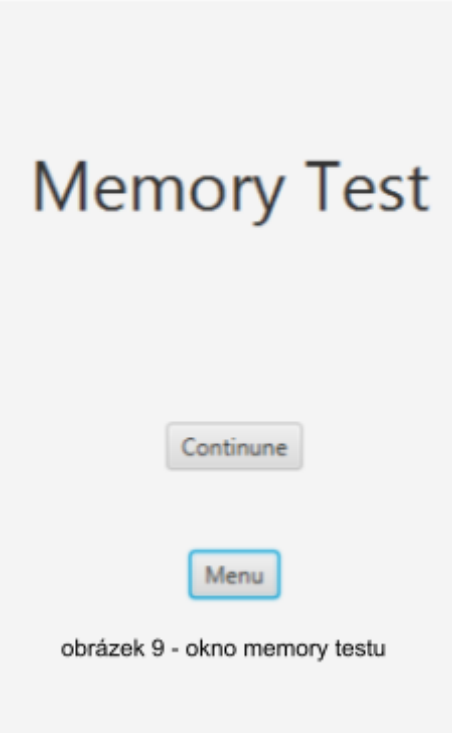


Obrázek 7 - Výpis reakční a průměrné reakční doby do konzole

obrázek 8 - okno s tlačítkem contiune

3.1.2 Memory test

Po kliknutí na tlačítko memory test v hlavním okně/menu(viz. obrázek 3) se spustí nové okno, kde se odehrává samotný hlavní okno memory testu(viz. obrázek 9).



obrázek 9 - okno memory testu

```
Hádané číslo můžete vidět níže
769  Napište číslo:
7
6
9
Hádané číslo můžete vidět níže
3215  Napište číslo:
3
2
1
5
Hádané číslo můžete vidět níže
38826  Napište číslo:
3
8
8
2
6
Hádané číslo můžete vidět níže
529521  Napište číslo:
5
2
8
5
1
2
Udělali jste chybu, můžete to zkusit znovu.
```

obrázek 10 - konzole memory testu

Nyní si uživatel může zvolit jestli chce
pokračovat v memory testu nebo jestli se chce vrátit do menu. Pro pokračování v memory testu musí uživatel stisknout tlačítko continue. Pokus se chce vrátit zpátky do hlavního okna/menu musí kliknout na tlačítko menu.

Po kliknutí na tlačítko continue se spustí memory test v konzoli(viz. obrázek 10) Uživatel si dané číslo musí zapamatovat a následně číslo zapsat, tak jak si jej pamatuje. Pokud uživatel číslo uhádne vygeneruje se další číslo, ale tentokrát o desetinný stupeň větší, než číslo předchozí. Cyklus se opakuje, dokud uživatel neudělá chybu. Jestliže se zadané číslo

neshoduje s inputem uživatele, program vypíše “udělali jste chybu můžete to zkusit znovu ” a poté skončí.

3.2 Funkcionalita programu

```
public void memoryTest(ActionEvent e) throws IOException, InterruptedException {
    boolean game = true;
    int size = 3;
    int score = 0;
    for (; game == true;) {
        Scanner sc = new Scanner(System.in);                    // Tvorba Scanu
        int[] scanNum = new int[size];                          // size = sc.nextInt();
        int[] arr = new int[size];                              // Tvorba Náhodného Pole
        int randomNum = 0;
        for (int i = 0; i <= arr.length - 1; i++) {
            Random num = new Random();
            randomNum = num.nextInt(9);
            arr[i] = randomNum;

        }
        System.out.println("Hádané číslo můžete vidět níže");   // Výpis Náhodného Pole
        for (int x = 0; x <= arr.length - 1; x++) {
            System.out.print(arr[x]);                           //vypis random cisla
        }
        System.out.println("  Napište číslo:");
        for (int i = 0; i < size; i++) {                        // Přečtení Elementu
            scanNum[i] = sc.nextInt();
        }
        boolean controlNum = true;
        for (int f = 0; f < size; f++) {                        // Kontrola Náhodného Pole s Inputem od Usera
            if (arr[f] == scanNum[f]) {
                controlNum = true;
            } else {
                controlNum = false;
                break;
            }
        }
        if (controlNum == true) {
            size++;
            score++;
        } else {
            game = false;
            System.out.println("Udělali jste chybu, vaše skore je" +" " + score + " " + ",můžete to zkusit znovu.");
            System.exit(0);

        }
    }
}
```

Obrázek 11 - metoda, která generuje a kontroluje zadané číslo s inputem uživatele

3.2.1 Metoda pro memory test

Metoda na obrázku 11 ze začátku vygeneruje náhodné číslo, a poté naskenuje input od uživatele. Jestliže se input uživatele(scanNum) rovná vygenerovanému číslu(arr) program se spustí znovu, akorát generované číslo bude mít o jeden desetinný stupeň více, než předtím. Jestliže se input uživatele nerovná rovná vygenerovanému číslu program skončí pomocí metody System.exit(). Metoda má jako parametr ActionEvent.

3.2.2 Metody pro reaction test

```
public void switchToReaction2(ActionEvent e) throws IOException, InterruptedException {
    System.out.println("Napište takové číslo, kolik chcete mít pokusů a stiskněte enter");
    Scanner scanner = new Scanner(System.in);
    pocetPokusu = scanner.nextInt();
    // Thread.sleep(2000);
    Random random = new Random();
    long randomnum = random.nextInt(4000) + 1500;
    Thread.sleep(randomnum);
    currentTime = System.currentTimeMillis();

    root = FXMLLoader.load(getClass().getResource("Reaction.fxml"));
    stage = (Stage) ((Node) e.getSource()).getScene().getWindow();
    scene = new Scene(root);
    stage.setScene(scene);
    stage.show();
```

obrázek 12 - metoda reaction testu 1.část

Metoda na obrázku 12 naskenuje od uživatele input typu int. Následně po dobu náhodně vygenerovaného času program čeká, poté si uloží současný čas a změní scénu.

```
public void startWatch2(ActionEvent e) throws IOException, InterruptedException {

        long stopTime = System.currentTimeMillis();
        root = FXMLLoader.load(getClass().getResource("reactionEnd.fxml"));
        stage = (Stage) ((Node) e.getSource()).getScene().getWindow();
        scene = new Scene(root);
        stage.setScene(scene);
        stage.show();
        long reactionTime = (stopTime - currentTime);
        prumerReactionTime += reactionTime;
        System.out.println(reactionTime + " " + "milisekund");
        i++;
```

obrázek 13 - metoda reaction testu část 2

```
    }
```

Metoda na obrázku 13 se spustí po kliknutí na tlačítko start(viz. obrázek 4)a poté si uloží současný čas do proměnné currentTime a změní scénu. Následně vypočítá reakční čas, tím že od současného času(stopTime) odečte minulý aktuální čas(currentTime), vypíše jej do konzole a zvýší hodnotu pomocné proměnné o 1.

```
public void tryAgain(ActionEvent e) throws IOException, InterruptedException {
    if (i != pocetPokusu) {
    Random random = new Random();
    long randomnum = random.nextInt(4000) + 1500;
    Thread.sleep(randomnum);
    currentTime = System.currentTimeMillis();

    root = FXMLLoader.load(getClass().getResource("Reaction.fxml"));
    stage = (Stage) ((Node) e.getSource()).getScene().getWindow();
    scene = new Scene(root);
    stage.setScene(scene);
    stage.show();
    }
    else {
        System.out.println("Hotovo");
        System.out.println("Vaše průměrná reakční doba je" + " " + prumerReactionTime/pocetPokusu+ " " + "milisekund")
        root = FXMLLoader.load(getClass().getResource("Main2.fxml"));
        stage = (Stage) ((Node) e.getSource()).getScene().getWindow();
        scene = new Scene(root);
        stage.setScene(scene);
        stage.show();

    }
}
```

obrázek 14 - metoda reaction testu 3.část

metoda na obrázku 14 zkontroluje, zda se pomocná proměnná i nerovná počtu pokusů. Pokud se proměnná i nerovná počtu pokusů, tak si uloží současný čas. Následně si uloží aktuální čas do proměnné currentTime a změní scénu. Pokud se proměnná i rovná počtu pokusů, tak do konzole vypíše průměrný reakční čas, který je vypočítaný na základě všech předchozích pokusů a změní scénu.

Závěr

Reaction test se podařilo splnit podle zadání. Memory test má ovšem malý nedostatek a to sice, že vygenerované náhodné číslo zůstává v konzoli po celou dobu memory testu(viz. obrázek 10), z důvodu neznalosti daného tématu se mi nepodařilo, aby se číslo po např. 3 vteřinách schovalo a tedy aby jež uživatel nemohl vidět.

Díky tomuto ročníkovému projektu jsem si se naučil lépe pracovat s JavaFX a obecně jsem se v programování zlepšil. Zopakoval jsem si základní věci, jako třeba vypisování pole a používání tříd scanner a random. Naučil jsem se věci jako, přepínání mezi více scénami, funkce např.System.currentTimeMillis();, pracovat s Java Scene Builder a tak dále.

Seznam obrázků

Seznam zdrojů a použité literatury

- Java (programovací jazyk) - Wikipedie,

  [Java] https://cs.wikipedia.org/wiki/Java_(programovac%C3%AD_jazyk). Accessed 1 May 2022.
- JavaFX - Wikipedie,

  [JavaFX] https://cs.wikipedia.org/wiki/JavaFX.

  Accessed 1 May 2022.
- JavaFX Scene Builder - Wikipedie,

  [JavaFX Scene Builder] https://cs.wikipedia.org/wiki/JavaFX_Scene_Builder.

  Accessed 1 May 2022.
- Eclipse(vývojové prostředí) - Wikipedie,

  [Eclipse IDE] https://cs.wikipedia.org/wiki/Eclipse_(v%C3%BDvojov%C3%A9_prost%C5%99ed%C3%AD).

  Accessed 1 May 2022.
- Reakční doba - Wikipedie,

  [Reakční doba] https://cs.wikipedia.org/wiki/Reak%C4%8Dn%C3%AD_doba.

  Accessed 1 May 2022.
- Paměť - Wikipedie,

  [Paměť] https://cs.wikipedia.org/wiki/Pam%C4%9B%C5%A5.

  Accessed 1 May 2022.
- Paměť - Wikipedie,

  [Senzorická paměť] https://cs.wikipedia.org/wiki/Pam%C4%9B%C5%A5.

  Accessed 1 May 2022.
- Paměť - Wikipedie,

  [Krátkodobá paměť] https://cs.wikipedia.org/wiki/Pam%C4%9B%C5%A5.

  Accessed 1 May 2022.