

Gymnázium, Praha 6, Arabská 14

Programování



Ročníková práce

Hana Machalíková

Paměťová hra

1.E

květen 2021

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, Arabská14 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 dne

Hana Machalíková

Název práce: Paměťová hra

Autor: Hana Machalíková

Anotace: Program má za úkol trénovat hráčovu paměť. Vždy na poli tlačítek 3x3 zobrazí jistou sekvenci tlačítek, která je náhodně vygenerována. Poté se hráč snaží tuto sekvenci zopakovat. Pokud se mu to podaří, vygeneruje program nové souřadnice dalšího tlačítka. Tyto souřadnice přidá za sekvenci, kterou zobrazuje uživateli. Vždy mu tedy zobrazí všechna tlačítka předchozí plus tlačítko nové. A opět je řada na hráči, který opakuje stejnou sekvenci jako předtím, avšak s dalším tlačítkem navíc. Pokud to zvládne bezchybně, posune se do dalšího levelu, kde má opět o tlačítko víc. Pokud stiskne nesprávné tlačítko, hra se ukončí a hráč musí začít znova.

Zadání: Hráč bude před sebou pole s tlačítky. Počet bude závislý na obtížnosti, kterou si hráč vybere. Postupně se bude na tlačítkách něco ukazovat. Např. obrázek. Vždy se začne u jednoho. A postupně se počet bude zvyšovat. Pokaždé, když hráč klikne na tlačítka ve správném pořadí, postoupí do dalšího levelu. Tam se přidá do série další tlačítko. A takto to bude pokračovat, dokud bude hráč sérii opakovat správně. Pokud udělá chybu, bude muset začít od začátku, tedy od 1.

Hra bude pro jednoho hráče. Ale může s sebou soupeřit více hráčů, ale další může hrát, až jeden skončí = špatně zopakuje sérii.

1. Obsah

1. Obsah	4
2. Úvod	5
3. Jak program funguje	5
3.1. Implementace grafického uživatelského rozhraní	5
3.2. Problém s probliknutím tlačítka	6
3.3. Nejdůležitější metody	6
4. Závěr	8
5. Seznamy	9
5.1. Zdroje	9
5.2. Seznam obrázků	9

2. Úvod

Tento dokument se zabývá popisem mé ročníkové práce na téma paměťová hra. Mělo by jít o snadné trénování paměti, kdy si dotyčný hráč musí zapamatovat, jak jdou po sobě tlačítka v tabulce o velikosti 3x3. Toto pořadí mu bude zobrazeno počítačem a hráčovým úkolem bude stejně zopakovat sekvenci, jež mu byla předvedena. Při správném zopakování sekvence se mu v dalším kole (levelu) zobrazí stejná sekvence s tlačítkem navíc na konci sekvence. A takto to pokračuje, dokud hráč neudělá chybu – nestiskne nesprávné tlačítko.

Tato hra je pro jednoho hráče. Je možné soutěžit mezi více lidmi, avšak každý si musí odehrát zvlášť a následně si zapamatovat nejvyšší skóre, kterého dosáhl.

3. Jak program funguje

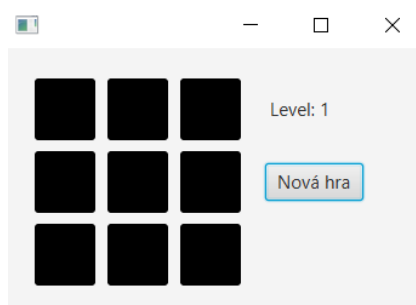
3.1. Implementace grafického uživatelského rozhraní

Celý program je naprogramován v grafické knihovně JavaFX, což je knihovna, dostupná v běžných balíčcích Javy od verze 8.0, pro tvorbu okenních aplikací. Jednoduše zde jde naprogramovat uživatelské rozhraní. Použitým jazykem je tedy Java a vývojovým prostředím jsou NetBeans IDE.

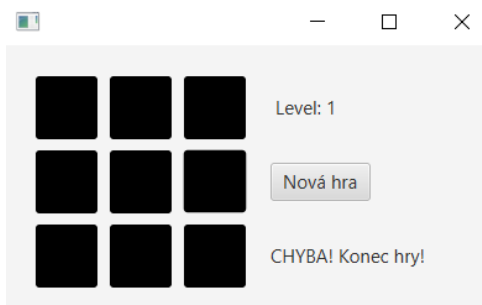
Jako komponenty používám GridPane, Button a Labely (*Obrázek 1*). GridPane mi jednoduše vytvoří tabulku, do které pak následně jednoduše vložím tlačítka o velikosti 50x50. Hned na začátku programu všem těmto tlačítkům nastavím černou barvu, aby byla tlačítka lépe vidět. Vedle tlačítek se zobrazí Label reprezentující level, ve kterém se hráč nachází. Toto číslo je shodné s počtem tlačítek, které sekvence obsahuje v daném levelu. Pod tímto labelem je ještě další Button s názvem „Nová hra“. Po stisknutí tohoto tlačítka se spustí nová hra. Tedy level je opět 1. Je nutné toto tlačítko stisknout již před první hrou, tedy hned po zobrazení okna. Předcházím tím tomu, že by se zobrazilo první tlačítko sekvence, ale hráč by ještě nebyl připravený. Tím pádem by toto tlačítko stejně musel zmáčknout.

Pokud hráč během hry špatně zopakuje sekvenci, na dalším Labelu s názvem „Stav“, umístěném pod tlačítkem „Nová hra“, se zobrazí text „CHYBA! Konec hry!“ (*Obrázek 2*). Pak už záleží pouze na hráči, zda to zkusí ještě jednou, nebo jestli aplikaci ukončí.

Pole tlačítek je veliké 3x3. Původně jsem zkoušela udělat 4x4, ale zjistila jsem, že je to moc. Při tomto vyšším počtu se sekvence pamatuje opravdu velice obtížně.



Obrázek 1 Okno aplikace



Obrázek 2 Okno aplikace s vypsanou chybou

3.2. Problém s probliknutím tlačítka

Dlouho jsem se potýkala s problémem, že jsem nevěděla, jak nechat tlačítko probliknout. Zkoušela jsem tlačítko nějak změnit, počkat třeba sekundu a nechat tlačítko změnit nazpět. Bohužel program vždy akorát počkal a nastavil poslední styl tlačítka, tedy styl původní.

Zkoušela jsem měnit barvu tlačítka, text napsaný na tlačítku, nechat ho zmizet, či z něj udělat ToggleButton, popřípadě jakkoli jinak měnit styl. Nic nepomohlo.

Poté jsem to na radu pana profesora Lány vyřešila pomocí Timelinu. Nejdříve tlačítko změním barvu na zelenou a následně Timeline po půl vteřině změním barvu opět na černou. Přesně to jsem potřebovala. Ještě jsem pak přidala krátkou pauzu po probliknutí (100 ms). Jinak se stávalo, že když se nějaké políčko mělo zobrazit dvakrát za sebou, ve skutečnosti svítilo dvojnásobnou dobu, aniž by mezitím zhaslo. A nejhorší bylo, když se mělo zobrazit třikrát. To už uživatel opravdu složitě rozpoznával, kolikrát má dané tlačítko stisknout.

3.3. Nejdůležitější metody

Jako první jsem vybrala metodu „UkazPolicko“. Ta se spouští zavoláním v ostatních metodách. Měla jsem s ní velké problémy popsané v podkapitole výše. Nakonec jsem naštěstí s pomocí pana profesora zvýrazňování tlačítka vyřešila. V této metodě se vytvoří Timeline a v něm 2 KeyFramy. Každý reprezentuje jednu barvu – černou (původní) a zelenou (problikávací). Má dva parametry jakožto souřadnice tlačítka, které má být zvýrazněno.

```
public static void UkazPolicko (int r, int s) { // metoda pro zvýraznění tlačítka
    Timeline timeline = new Timeline(
        new KeyFrame(Duration.millis(prodleva), ae -> b[r][s].setStyle("-fx-background-color: #000000; ")),
        new KeyFrame(Duration.millis(prodleva - 500), ae -> b[r][s].setStyle("-fx-background-color: #00ff00; "));
    timeline.play();
}
```

Obrázek 3 Metoda "UkazPolicko"

Metoda „HracStiskl“ se spustí při stisknutí jakéhokoliv tlačítka v poli 3x3. Tato metoda má dva parametry, které reprezentují souřadnice daného tlačítka, aby následně mohlo dojít k ověření správnosti. Nejdříve zkontroluje, zda je hráč vůbec na řadě. Pokud ne, tak se tímto stisknutím nezabývá. Pokud však ano, tak zkontroluje správnost tlačítka, které hráč stiskl. Do proměnné „zatimspravne“ se ukládá počet tlačítek, které již zopakoval správně. Díky tomuto číslu se dokážeme orientovat v poli souřadnic a zkontrolovat, zda se souřadnice stisknutého tlačítka shodují s čísly na místě hodnoty „zatimspravne“. Pokud „zatimspravne“ dosáhne hodnoty rovné kolu (úrovně), hráčův tah končí a program zobrazí sekvenci s tlačítkem navíc. Pokud se hráč splete a zmáčkne špatné tlačítko, Label s názvem „Stav“ vypíše „CHYBA! Konec hry!“. A dokud hráč nestiskne tlačítko „Nová hra“, tak program nic nedělá a pouze čeká, zda se hráč rozhodne to zkusit znovu, či to vzdá a celou aplikaci ukončí.

```

public static void HracStiskl (int r, int s) { // metoda, kdy se kontroluje správnost tlačítka, které hráč stisknul
    if (hrajehrac == true) { // zkontrolování, zda je hráč na řadě
        if ((souradnice[0][zatimspravne] == r) && (souradnice[1][zatimspravne] == s)) { // ověření správnosti hráčova stisknutí
            zatimspravne++;
            if (zatimspravne == kolo) { // při bezchybném zopakování celé sekvence ukáže další level
                kolo++;
                UkazVzor(kolo);
            }
        } else { // pokud hráč zopakuje vzor špatně, vypíše se "chyba"
            hrajehrac = false;
            stav.setText("CHYBA! Konec hry!");
        }
    }
}

```

Obrázek 4 Metoda "HracStiskl"

Ve třetí metodě s názvem „UkazVzor“ program uživateli ukazuje sekvenci z předchozích kol. Nejdříve se nastaví, že teď je na řadě počítač, aby mu do toho hráč nijak nezasahoval. Pokud se hráč nachází pouze v prvním levelu, tato část se přeskočí. Následuje vygenerování 2 náhodných celých čísel, která mohou nabývat hodnoty 0, 1, nebo 2. Tato čísla poslouží jako souřadnice k novému tlačítku, které se zařadí do sekvence a zobrazí se uživateli. Poté už se jen vynuluje proměnná „zatimspravne“.

```

public static void UkazVzor (int pocet) { // metoda pro zobrazení sekvence s jedním dalším prvkem
    hrajehrac = false; // čas pro počítač
    level.setText(" Level: " + pocet); // přepsání levelu
    int pozice; // pozice tlačítka v poli souřadnic
    for(pozice = 0; pozice < (pocet-1); pozice++) { // zopakování sekvence tlačítek z předchozích kol
        UkazPolicko(souradnice[0][pozice],souradnice[1][pozice]);
        prodleva += 600; // zvýšení prodlevy, aby se stále ukazovala všechna tlačítka
        pauza(100); // pauza, aby bylo viditelné bliknutí, když má bliknout 2x za sebou
    }
    double d = Math.random() * 3;
    int random = (int) Math.floor(d); //vygenerujeme souřadnice dalšího tlačítka
    souradnice[0][pozice] = random;
    d = Math.random() * 3;
    random = (int) Math.floor(d);
    souradnice[1][pozice] = random;
    UkazPolicko(souradnice[0][pozice], souradnice[1][pozice]); // ukázání nového tlačítka hráči
    zatimspravne = 0; // vynulování počtu políček, která hráč zopakoval správně
    hrajehrac = true; // počítač přenechá hru hráči
    prodleva = 500 + (pocet * 100); // nastavení prodlevy na základní velikost
}

```

Obrázek 5 Metoda "UkazVzor"

4. Závěr

Celkově jsem s tímto programem docela spokojená. Na to, jak hrozně moc mi během psaní nechtělo fungovat, to nedopadlo úplně nejhůř. Samozřejmě je ještě co vylepšovat. Například by se dalo ještě vypisovat, zda je na řadě hráč nebo jestli program předvádí sekvenci. Toto jsem zkoušela do programu zakomponovat, ale úplně jsem se do toho zamotala, že to nakonec vypisovalo pouze, že je na řadě hráč. Tak jsem to radši vzdala, jelikož to bylo velice matoucí. Myslím si však, že program funguje vcelku dobře i bez toho.

Když jsem narazila na problém popsany v kapitole 3.2, myslela jsem, že tím celý můj projekt skončí, že nebudu schopná nijak ho zprovoznit. Už jsem byla připravená použít náhradní řešení, kdy by se souřadnice vypisovaly do výstupu. Tím ale celá hra ztratila smysl, jelikož by se dalo naprosto jednoduše opisovat a žádný trénink paměti by zde neprobíhal. Proto jsem velice ráda, že se mi povedlo tento problém vyřešit.

Se svou prací jsem tedy spokojená, obzvlášť proto, že program dělá vše, co bylo v zadání. To pro mě je nejdůležitější. O tom, že je vždy co vylepšit, není pochyb a ani tento program není výjimkou.

5. Seznamy

5.1. Zdroje

2021. <https://riptutorial.com/javafx/example/670/animating-a-property-with-timeline>. *riptutorial.com*. [Online] 5. květen 2021. [Citace: 2. květen 2021.]

2021. https://stackoverflow.com/questions/24104313/how-do-i-make-a-delay-in-java?fbclid=IwAR1d_ARTwq-f_T_MrANk56CMO7uo9VWWcwSD9nNeiRxhklqUJAw3K39GCzA. *stackoverflow.com*. [Online] 3. květen 2021. [Citace: 19. duben 2021.]

Jenkov, Jakob. 2021.

<http://tutorials.jenkov.com/javafx/button.html?fbclid=IwAR21EzR3n2VMzAfUeT9xZDrbGQkENRnUVau6ugYHrcrWrJbdoz-UL0wZig0>. *tutorials.jenkov.com*. [Online] 3. květen 2021. [Citace: 17. duben 2021.]

<http://tutorials.jenkov.com/javafx/button.html?fbclid=IwAR21EzR3n2VMzAfUeT9xZDrbGQkENRnUVau6ugYHrcrWrJbdoz-UL0wZig0>.

Mikula, Tomas. 2021. <http://tomasmikula.github.io/blog/2014/06/04/timers-in-javafx-and-reactfx.html>. *tomasmikula.github.io*. [Online] 3. květen 2021. [Citace: 3. květen 2021.]

5.2. Seznam obrázků

Obrázek 1. Okno aplikace	Chyba! Záložka není definována.
Obrázek 2. Okno aplikace s vypsanou chybou	Chyba! Záložka není definována.
Obrázek 3 Metoda "UkazPolicko"	6
Obrázek 4 Metoda "HracStiskl"	7
Obrázek 5 Metoda "UkazVzor"	7