

Gymnázium, Praha 6, Arabská 14

Programování

ROČNÍKOVÁ PRÁCE



2023

Radim Tříletý

**Gymnázium, Praha 6,
Arabská 14**

Arabská 14, Praha 6, 160 00

ROČNÍKOVÁ PRÁCE

Předmět: Programování

Téma: Pacman

Autor: Radim Tříletý

Třída: 2. E

Školní rok: 2022/23

Vedoucí práce: Mgr. Jan Lána

Třídní učitel: Mgr. Blanka Hniličková

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 28. dubna 2023

Anotace:

V tomto ročníkovém projektu je předkládán program v programovacím jazyce Java. Cílem mé práce bylo naprogramovat hru Pac-man, hra spočívá, v tom, že hrajete za žlutou kuličku a snažíte se sbírat potřebný počet teček, aby jste vyhráli. Není to však tak jednoduché, jelikož se Vás celou dobu snaží chytit duchové. Program je napsán pomocí grafického rozhraní AWT (Abstract Window Toolkit), které je součástí programovacího jazyka Java.

Annotation:

In this year's project, a program in the Java programming language is presented. The aim of my work was to program the game Pac-man, the game consists in playing as a yellow ball and trying to collect the necessary number of dots to win. It's not that easy though, as there are ghosts trying to catch you all the time. The program is written using the graphical interface AWT (Abstract Window Toolkit), which is part of the Java programming language.

Zadání ročníkového projektu

Popis:

Jedná se o hru ve, které hráč utíká v bludišti před duchy. Hráč dostává body za se sbírané tečky. Hra končí, když duchové dopadnou hráče nebo, když se sbírá určitý počet teček.

Platforma:

Java

- AWT
- Java Swing
- Java Util

Obsah

1. Úvod	1
2. Knihovny	2
2.1. Knihovna AWT	2
2.2. Knihovna Java Swing	2
2.3. Knihovna Java Util	2
3. Třída Start	3
4. Třída Omg a její metody	4
4.1. metoda Omg()	4
4.2. metoda Plocha()	5
4.3. metoda hra()	5
4.4. metoda pohyb()	6
4.5. metoda kulicky()	6
4.6. metoda generujDuchy()	7
4.7. metoda Malovani()	8
5. Uživatelské problémy	9
6. Závěr	10
7. Seznam obrázků	11
8. Seznam zdrojů	12

1. Úvod

Tématem mé ročníkové práce je programování hry Pacman. Téma jsem si zvolil z důvodu, že jsem byl zvědaví, jaké je to programovat hru, která je složitější a také mám hru Pacman velice rád.

Program, který jsem naprogramoval, je hrou programovanou v Javě, ve které hráč ovládá Pacmana. Hra má několik funkcí, včetně tlačítka pro reset hry, sbírání teček v bludišti a mnoho dalšího. Program využívá Java Swing, pro grafické rozhraní, AWT pro události způsobené hráčem a také Java Until pro práci s časem (počítání kolik sekund hráč hraje hru). Program obsahuje třídu Omg a Start.

Třída Omg využívá JPanel, implementuje ActionListener a mnoho více. Má také mnoho metod, které se využívají během hry.

Třída Start využívá JFrame. Třída Start převážně slouží k zapnutí hry a vytvoření okna pro třídu Omg.

Celkově je moje práce jednoduchý program, který využívá základní funkce Javy a snaží se vytvořit hru zvanou Pacman.

2. Knihovny

2.1. Knihovna AWT

Knihovna AWT (Abstract Window Toolkit) je starší knihovna grafického uživatelského rozhraní (GUI) pro Java. Jedná se o původní nástroj pro tvorbu grafického uživatelského rozhraní, vyvíjený firmou Sun Microsystems jako součást Javy. První verze vyšla v roce 1995. Poskytuje potřebné věci pro vytváření a správu GUI prvků, jako například tlačítka a textová pole. AWT je také součástí JDK (Java Development Kit). [1]

2.2. Knihovna Java Swing

Knihovna Java Swing je moderní knihovna GUI. Poskytuje také větší flexibilitu a rozmanitost než AWT. Je to také knihovna uživatelských prvků, pro ovládání počítače pomocí grafického rozhraní a aplikačního rozhraní. Pomocí Swingu je možné vytvářet okna , tlačítka atd. [2]

2.3. Knihovna Java Util

Java Util je soubor tříd a funkcí, které poskytují nástroje pro práci s daty. Tyto třídy zahrnují nástroje pro manipulaci s datumy a časy, které jako jediné v této knihovně využívám. [3]

3. Třída Start

Třída Start je třída, která vytváří okno pro mojí hru. Tato třída využívá JFrame, což znamená, že má funkce na vytvoření okna a jeho ovládání. Třída Start má také několik metod, jako jsou například Start(), Spusteni() a main().

Metoda Start() slouží k zapnutí okna, nastavení všech jeho atributů (jako jsou jeho velikost, či název) a vypnutí celého programu po kliknutí na křížek. Také, díky této metodě, na vás ze začátku hry vyskočí zpráva.

Metoda Spusteni() přidává do okna herní plátno (třídu Omg).

Metoda main() je vstupním bodem v programu. Celkově spouští tuto hru a ukazuje ji na obrazovce.

```
public class Pacman extends JFrame {  
  
    public Pacman() {  
        // začáteční obarazovka a nastavení herního okna  
        Spusteni();  
        JOptionPane.showMessageDialog( parentComponent: this, message: "Hra se spustí tlačítkem \"Z\" (jako Začátek)");  
        this.setTitle("PAC-MAN");  
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        this.setSize( width: 1050, height: 800);  
    }  
  
    private void Spusteni() {  
        add(new Omg());  
    }  
  
    public static void main(String[] args) {  
        EventQueue.invokeLater(() -> {  
            var ch = new Pacman();  
            ch.setVisible(true);  
        });  
    }  
}
```

obr.1 - třída Start

4. Třída Omg a její metody

Třída Omg je mojí druhou a poslední třídou v mém programu. Třída Omg využívá JPanel, narozdíl od třídy Start, která využívá JFrame. Tato třída má také několik metod, jako jsou například Omg(), Plocha() atd. Nejvýznamnější z nich jsem si rozdělil do podkapitol, aby se v této dokumentaci dalo lépe vyznat.

4.1. metoda Omg()

Metoda Omg se vytváří při spuštění hry. Díky ní jsou inicializovány všechny potřebné věci. Taktéž vytváří Timer, který začne každých 5 sekund resetovat duchy a umisťovat je na náhodná místa pomocí metody resetDuchu(). Další co tato metoda dělá je, že vytváří text na vršku obrazovky, aby hráč věděl kolik teček musí se sbírat. Tato metoda taky vytváří tlačítko, které když hráč zmáčkne, resetuje se potřebné proměnné a tím i celou hru. Toto tlačítko je viditelné po celou dobu hry, hned vedle potřebného počtu teček.

```
public Omg() {  
  
    Timer timer = new Timer( delay: 5000, e -> resetDuchu());  
    if (!vhra) {  
        timer.stop();  
    }  
    timer.setInitialDelay(5000);  
    timer.start();  
    // nastavuje reset duchů na sekundy  
  
    generujDuchy();  
    zapcas = new Date();  
    Obrazky();  
    Timer();  
    plocha();  
    int finscore = 500; //potřebné skóre  
    initkul();  
    JLabel scoreT = new JLabel( text: "Potřebné tečky: " + finscore); // nastavení textu  
    scoreT.setForeground(Color.white); // nastavení barvy textu na bílou  
    scoreT.setFont(new Font( name: "Verdana", Font.BOLD, size: 20)); // nastavení velikosti a stylu písma  
    add(scoreT);  
  
    JButton resetBtn = new JButton( text: "Reset");  
    resetBtn.addActionListener(e -> {  
        // resetujeme potřebné proměnné
```

obr.2 - metoda Omg()

4.2. metoda Plocha()

Metoda Plocha() je v mém kódu použita na nastavení pozadí hrací plochy na černou barvu. Taktéž poskytuje možnost aby fungovaly tlačítka pomocí `addKeyListener(new Tlacitka());`;

, což znamená, že “poslouchá” klávesnici aby věděla jaké tlačítko se zmáčklo.

```
private void plocha() {  
  
    addKeyListener(new Tlacitka());  
  
    setFocusable(true);  
  
    setBackground(Color.black);  
}
```

obr.3 - metoda plocha()

4.3. metoda hra()

Metoda hra() se snaží zabezpečit úplný běh hry. Metoda hra běží opakovaně a kontroluje jestli hráč vyhrál nebo prohrál. Pokaždě se stala jedna z těchto možností stane zavolá metoda hra() výherní či prohrávací obrazovku a ukončí se. Pokud však hráč nevyhrál nebo neprohrál metoda hra() pořád běží. Tato metoda má taky převážně na starosti řízení a koordinaci herních prvků.

```
private void hra(Graphics2D g) {  
    if (score >= 500) {  
        // pokud byla hra vyhrána, zobrazíme výherní obrazovku  
        vyhobrzovka(g);  
        return;  
    }  
    if (prohra) {  
        // pokud byla hra prohrána, zobrazíme prohrávací obrazovku  
        proobrazovka(g);  
        return;  
    }  
    pohyb();  
    Malovani(g);  
}
```

obr.4 - metoda hra()

4.4. metoda pohyb()

Metoda pohyb() se stará o pohyb kuličky (Pacmana). Metoda pohyb() je volána z metody hra(), kde se aktualizuje pozice Pacmana. Tato metoda má definovanou velikost políček a poté zajišťuje, aby se Pacman pohyboval po celých políčkách a ne jenom jejich částech. Pokud je tedy Pacman na pozici, která je dělitelná velikostí herního pole a současně se mění směr jeho pohybu, změní se jeho směr pohybu. Taktéž se v metodě pohyb() aktualizují jeho souřadnice na ose X a na ose Y. Pacman se poté posouvá v závislosti na proměnné rychlost a směru pohybu.

```
private void pohyb() {  
    //pohyb pacmana a velikost políček  
    int Pole = 20;  
    if (y % Pole == 0 && x % Pole == 0) {  
        if (pohyb_y != 0 || pohyb_x != 0) {  
            pacman1 = pohyb_y;  
            pacman2 = pohyb_x;  
        }  
    }  
  
    y = y + rychlost * pacman1;  
    x = x + rychlost * pacman2;  
}
```

obr.5 - metoda pohyb()

4.5. metoda kulicky()

V této metodě jsou generovány kuličky a kontrolováno zda je hráč snědl či nikoliv. Tato metoda se skládá z několika podfunkcí (dalo by se říci).

V první části metody jsou vytvářeny kuličky a přidělovány jim jejich pozice, barva a velikost.

Poté se v druhé části kontroluje pozice hráče a pokavaď narazí na kuličku, tak se kulička na příslušné pozici vymaže. Skóre je poté zvýšeno o bod.

Ve třetí části je, že pokud hráč nasbírá určitý počet kuliček, tak se hra zastaví a vyskočí na něj upozornění, že vyhrál. V tomto upozornění je taky napsán jeho čas, od spuštění hry či zmáčknutí tlačítka reset.

V poslední části je udělaná if podmínka, která ukazuje hráči jeho aktuální počet snědených teček v dolní části obrazovky. Toto číslo se ukáže vždy, když hráč sní kuličku.

```
private void kulicky(Graphics2D g) {
    //generování kuliček
    g.setColor(Color.LIGHT_GRAY);
    for (int i = 2; i < 48; i++) {
        for (int j = 2; j < 33; j++) {
            if (kuli[i][j]) {
                int kulx = i * 20 + 9;
                int kuly = j * 20 + 9;
                boolean snezkul = Math.abs(x + 500 - kulx) < 10 && Math.abs(y + 265 - kuly) < 10;
                if (!snezkul) {
                    g.fillOval(kulx, kuly, width: 4, height: 4);
                } else {
                    kuli[i][j] = false;
                    score++;

                    if (score >= 500) {
                        rychlost = 0;
                        vhra = false;
                        JOptionPane.showMessageDialog( parentComponent: this, message: "Vyhrál jsi!! Tvůj čas: " + cas + " sekund");
                    }
                }
            }
            if (snezkul) {
                // zobrazuje aktuální poček teček
                g.setColor(Color.white); // nastavíme barvu na bílou
                g.setFont(new Font( name: "Verdana", Font.BOLD, size: 20)); // nastavíme velikost a styl písma
                g.drawString( str: "Tečky: " + score, x: 400, y: 720); // zobrazíme novou hodnotu počítadla
            }
        }
    }
}
```

obr.6 - metoda kulicky()

4.6. metoda generujDuchy()

Tato metoda generuje duchy na náhodných místech v hlavním obdelníku . Náhodné pozice jsou generovány pomocí funkce `Math.random()` , která vrací hodnotu mezi 0 a 1. Tato hodnota je poté násobena a duch dostane hodnotu pro osu X mezi 2 a 47, a pro osu Y mezi 2 a 32. Cyklus se poté opakuje, dokud není dostatečný počet duchů. Duchy poté vykreslí metoda `Malovani()`.

```
private void generujDuchy() {
    //generuje duchy v pozicích hlavního obdelníku
    int i = 0;
    while (i < duchove.length) {
        int x = (int) (Math.random() * 46 + 2);
        int y = (int) (Math.random() * 31 + 2);
        duchove[i][0] = x;
        duchove[i][1] = y;
        i++;
    }
}
```

obr.7 - metoda generujDuchy()

4.7. metoda Malovani()

V této metodě se provádí vykreslení hry (kolizí, stěn, Pacmana, duchů).

Kód začíná vykreslením kuliček z metody kulicky() a vykreslením Pacmana. Taktéž tato část vkládá Pacmana na souřadnice $x + 500$ a $y + 265$.

Dále dává Pacmanovy kolize v podobě neviditelného obdelníka. Poté následuje cyklus, který vykreslí obrázky duchů, a taktéž jim přidělí neviditelné obdelníky obdelníky.

Poté se kontroluje jestli se tyto obdelníky (Pacmanův a duchův) nesrazily, a pokud ano, tak hra končí prohrou.

Další část je, že je-li prohra nebo výhra, tak funkce končí a nic jiného se nedělá.

Potom následuje vykreslování stěn pole, ve kterém hráč hraje. Všem se dají souřadnice a jakou výšku a šířku mají mít.

Nakonec se na souřadnice vykreslených obdelníků dají ty neviditelné obdelníky, které kontrolují zda do nich Pacman narazil a pokud ano, tak mu dají rychlost na 0, čímž ho zastaví.

```
private void Malovani(Graphics2D g) {
    kulicky(g); //vykreslení kuliček
    g.drawImage(kulicka, x: x + 500, y: y + 265, observer: this); // vykreslení pac-mana
    Rectangle kulickaRect = new Rectangle(x: x + 500, y: y + 265, kulicka.getWidth( observer: this), kulicka.getHeight( observer: this));
    for (int[] ints : duchove) {
        int x = ints[0] * 20;
        int y = ints[1] * 20;
        g.drawImage(duch, x, y, observer: this); // vykreslení duchů
        Rectangle DuchR = new Rectangle(x, y, duch.getWidth( observer: this), duch.getHeight( observer: this)); //kolize pro duchy
        if (kulickaRect.intersects(DuchR)) {
            //prohra když se duch dotkne
            prohra = true;
        }
    }
    if (vyhra) {
        return;
    }
    if (prohra) {
        return;
    }

    //steny
    g.setColor(Color.blue);
    // základní obdelník
    g.fillRect(x: 20, y: 20, width: 20, height: 660);
    g.fillRect(x: 980, y: 20, width: 20, height: 680);
    g.fillRect(x: 20, y: 20, width: 960, height: 20);
    g.fillRect(x: 20, y: 680, width: 960, height: 20);
}
```

obr.8 - část metody Malovani()

5. Uživatelské problémy

Tato hra není stvořena pro uživatele, který nezná hru Pacman. Není zde žádný návod jak se hra hraje.

Také je tady větší problém v podobě bugů. Hráč se může jednoduše dostat přes stěny, i když tam mám kolize, opakováním mačkáním jedné šipky pořád dokola.

Poslední problém, který moje hra má, je možnost objevení se ducha přímo v Pacmanovi, což vyústí v automatickou prohru.

6. Závěr

V ročníkovém projektu se mi podařilo naprogramovat hru v jazyce Java. Tato hra splňuje základní a nejdůležitější funkce pro běh programu, jako je například pohyb, počítání kuliček, proherní a výherní obrazovka.

Největší problém jsem měl s detekcí kolizí. S tím jsem si nevěděl rady a jsem rád, že jsem to nějak dokázal udělat. Při psaní mého kódu jsem narazil na spoustu problémů, ale nejtěžší problém, který mi nešel vyřešit, byla ta detekce kolizí. Ostatní problémy jsem nějak také dokázal vyřešit.

Největší problém, který jsem bohužel nedokázal vyřešit, byl pohyb duchů, jako je v obyčejné hře Pacman. Nahradil jsem ho tedy náhodným objevováním duchů na náhodných pozicích.

Celkově bych považoval projekt za úspěšný. Řekl bych, že jsem získal nové zkušenosti v programování, tak i nové zkušenosti s Javou. Věřím, že může dojít ke zlepšení programu, ale i tak jsem s mojí prací velmi spokojený.

7. Seznam obrázků

Obr.1 - třída Main	3
Obr.2 - metoda Omg()	4
Obr.3 - metoda plocha()	5
Obr.4 - metoda hra()	5
Obr.5 - metoda pohyb()	6
Obr.6 - metoda kulicky()	7
Obr.7 - metoda generujDuchy()	7
Obr.8 - část metody Malovani()	8

8. Seznam zdrojů

- [1] Přispěvatelé projektu Wikimedia. (2017). Abstract Window Toolkit. *cs.wikipedia.org*. URL: https://cs.wikipedia.org/wiki/Abstract_Window_Toolkit
- [2] Přispěvatelé projektu Wikimedia. (2023c). Swing (Java). *cs.wikipedia.org*. URL: [https://cs.wikipedia.org/wiki/Swing_\(Java\)](https://cs.wikipedia.org/wiki/Swing_(Java))
- [3] Wikipedia contributors. (2023). Java collections framework. *Wikipedia*. URL: https://en.wikipedia.org/wiki/Java_collections_framework
- [4] Troadec, G. (2022, December 23). How You Can Code Pac-Man In 2 Weeks Using Java | Javarevisited. *Medium*. URL: <https://medium.com/javarevisited/what-i-achieved-after-13-weeks-of-learning-java-e6a081c401c9>
- [5] gaspar coding. (2020, September 10). *Pacman in Java, Programming Tutorial 2/2* [Video]. YouTube. URL: <https://www.youtube.com/watch?v=r7i25lbmBd4>
- [6] **Obrázek duchů**
<https://www.google.com/imgres?imgurl=https%3A%2F%2Fwww.pngwing.com%2Fpngs%2F269%2F204%2Fpng-transparent-red-pacman-ghost-pac-man-party-worlds-biggest-pac-man-ghosts-pac-man-ghost-video-game-smiley-ghosts-thumbnail.png&tbnid=V2PiH5pI92PZVM&vet=12ahUKEwiVmrHOjcr-AhWshPoHHeISDaAQMygHegQIARAx.i&imgrefurl=https%3A%2F%2Fwww.pngwing.com%2Fen%2Ffree-png-zwzjt&docid=dB5pF4HlUu3PAM&w=360&h=360&q=pycman%20ghost%20prazdne%20pozadi&ved=2ahUKEwiVmrHOjcr-AhWshPoHHeISDaAQMygHegQIARAx>
- [7] Jenkov, J. (n.d.). *Java Tutorial*. URL: <https://jenkov.com/tutorials/java/index.html>
- [8] <https://www.ics.uci.edu/~jacobson/ics80J/Lab4/Pacman.java>