SPŠE Ječná

Obor: Programování aplikací

Ječná 30

Labirint v JSwingu

Rusňak Matyáš

Informační technologie

Rok vytvoření

2025/8.5

Obsah

1. Cíl

Cílem projektu bylo vytvořit Labirint s vlákny.

Kde bude možné jednoduše měnit labirint pro potřeby.

Je tam možné se pohybovat svým cahrakterem po bludišti a plnit pod tasky.

Podařilo se vytvořit NPC které pronásleduje hráče.

Dalším bodem bylo si vyzkoušet Jswing a vlákna.

2. Popis hry

2.1 Algoritmus

Prostředí je mapa která se vytváří při spuštění hry pomocí dvourozměrného pole.

Je složena z blocků ze kterých se skládají zdi a cesty v labirintů.

Velikost blocků je 40 na 40 pixelů.

Celá mapa je o velikosti 40 na 36 blocků.

2.2 Postavy

Postavy jsou 2 hráč a nepřítel.

Hráčem se pohybuješ po mapě.

Nepřítel pronásleduje hráče po jeho stopách.

2.3 Mechaniky

Chození po mapě hráčskou postavou pomocí šipek.

Pohyb je napsán pomocí key Adapter

Nepřítel pronásleduje stopu hráče a kopíruje jeho chození.

Celé pronásledování funguje pomocí Runnable

Na startovním okně je možné nastavit obtížnost přesněji rychlost pohybu nepřítele.

3. System requirement

Program byl vyvyjen v jazyce Java konkrétně v Java SE 23.0.1. Pro spuštění je duležité mít spávny JDK. K programu neni potreba žádná externí knihovna ani rámec.

Program je spustitelný v command line tak v libovoném prostředí podporující Java.

(Tim si nejsem jsitej a nerad bych kecal=Prosim otvirejte v **commad line** nebo **Intel Idea Jet Brains**)

4. Základní struktůra

Program je navržen objektově. A je rozložen do několika hlavních tříd a pak podtříd které spolu komunikují a využivají vlastnosti druhých.

Main slouží pro prvotní spuštění programu.

Třáda Walls() představuje mapu hry uchovávající aktuální stav pokroku hráče a nástroj vykreslení jednotlivých částí kodu.

Třáda Player() mění pozici hráče a určuje kam muže a nemůže jít + dodává možnost interagovat s třídama oddědenými od Doors().

Třáda MyFrame() je okno na kterém hra běží a vykresluje aktualni stav.

Třáda StartWindow() je okno díky kterému se dostanete k hlavní hře.

V třídě se taky sjednotí jeden objekt MyFrame() který je poté sdílený mezi třídami.

Toto platí i pro FollowingPlayer().

Třáda GameLoop() je hlavní třída kde se updateuje okno a hráč a stav hry v intervalech 60 FPS.

Třáda FollowingPlayer() uchovává hodnoty poslední polohy hráče a aktualizuje třídu Enemy() pro pohyb nepřítele.

5. Testovací data

Program lze testovat manuálně prostřednictvím různých scénářů pokrývající klíčové herní situace.

Mezi testy patří kontrola funkčnosti programu při zadani špatného vstupu.

Kontrola výtěztví a správné resetování hry při smrti (nepřetrvávají žádné hodnoty rušící správný pruběh hry)

6. Uživatelká příručka

Vše se nachází ve hře jako tlačítko help na horním levém místě okna hry.

Pohyb je pomocí šipek.

Interakce se dveřmi je automatická.

Nepřítel začne pronásledovat až po 5 změnách pozice a v ten moment se teleportuje na první poyici hráče

Pro zadání odpovědi vyžadujici text je nutno nejdrive otazku odsouhlasit.

Toto však neplatí pro matematické odpovědi tam to lze zadat rovnou.

7. Závěr

Vytváření hry mě bavilo a donutilo mě se naučit nové věci.

Největší problémy nastaly u grafické stránky a pochopení funkce Grapics 2D a Runnable.

Další oříšky byly u vytváření mapy aby byla co nejvice dynamická a lehce měnitelná dle potřeb.

Mezi mé problémy zapadala komunikace mezi třídami.

Tento projekt mi pomohl se více soustředit na moržosti programování.

8. Zdroje

- Chat GPT
- https://www.youtube.com/watch?v=xND0t1pr3KY&pp=ygUIYnJvIGNvZGU%3D

1

Chat GPT

```
// creat GUI window (MyFrame) must be on EDT=Event Dispatch Thread
        SwingUtilities.invokeLater(() -> {
            MyFrame frame = new MyFrame(name); // GUI window

            // Vlákna se spustí v samostatném vlákně, ale až po vytvoření GUI
            //it will be easier for gui because gui can freeze if i wlii do some activities in action listener
            new Thread(() -> {

                GameLoop loop = new GameLoop(frame);



                System.out.println(modes);

                FollowingPlayer followingPlayer = new FollowingPlayer(frame, loop, modes);
                Thread thread1 = new Thread(loop);
                Thread thread2 = new Thread(followingPlayer);



                thread1.start();
                thread2.start();

                try {
                    thread1.join();
                    thread2.join();
                } catch (InterruptedException ex) {
                    ex.printStackTrace();
                }                }).start();             });
```

2

Chat GPT

```
@Override
public void paint(Graphics g) {
   // Vytvoření off-screen bufferu (pouze JEDNOU, ne pokaždé v paint)
   if (bufferedImage == null) {
      bufferedImage = new BufferedImage(widthWindow, heightWindow, BufferedImage.TYPE_INT_ARGB);
   }

   g2 = bufferedImage.createGraphics();

   // Vyčisti pozadí
   g2.setColor(getBackground());
   g2.fillRect(0, 0, getWidth(), getHeight());

   // Vykresli herní objekty
   walls.paint(g2);
   player.draw(g2);
   enemy.draw(g2);

   endText();



   // Přenes obraz z bufferu na obrazovku
   g.drawImage(bufferedImage, 0, 0, this);

   g2.dispose(); // ukončíme práci s g2, ne s g!
}

public void endText() {
   // Game over text
   if (gameOver) {
      g2.setColor(Color.RED);
      g2.setFont(new Font("Arial", Font.BOLD, 250));
      g2.drawString("Game Over", 100, 400);

   }}
```