

Střední Průmyslová Škola Elektrotechnická Ječná

Programování

adresa školy:

Mine Sweeper

Robin Kremlík

Informační technologie

2023

Obsah

- 1 Cíl práce
- 2 Software
- 3 Program
- 4 Závěr
- 5 Zdroje

1 Cíl práce

hra: Hledání min

Cílem hry je odhalit všechna políčka obsahující minu nebo, ještě přesněji, odkrýt všechna políčka, která minu neobsahují.

2 Software

IntelliJ Idea k naprogramování programu
Chat GBT k dokumentaci a komentářů v programu
Google docks k napsání dokumentace

3 Program

Class

Game:

Celkově kód nastaví hru vytvořením herního panelu a herního okna a poté požaduje zaměření vstupu pro panel. To naznačuje, že třídy „GameWindow“ a „GamePanel“ jsou definovány jinde v kódové základně a pravděpodobně obsahují skutečnou implementaci pro vykreslování hry a zpracování uživatelského vstupu.

GamePanel:

Třída definuje několik instančních proměnných pro uložení rozměrů a vlastností herního pole, jako je velikost dlaždice, počet sloupců a řádků, šířka a výška a počet bomb.

Vytvoří instanci třídy "Cell" a ArrayList s názvem "field", které reprezentují herní pole.

Třída poskytuje getry pro různé vlastnosti, jako je velikost dlaždice, počet sloupců a řádků.

Konstruktor nastaví preferovanou velikost herního panelu, barvu pozadí a umožňuje dvojité ukládání do vyrovnávací paměti.

Třída implementuje metodu „run“ z rozhraní Runnable, která představuje herní smyčku. Vypočítá čas na snímek a čas na tik a poté vstoupí do herní smyčky.

V rámci herní smyčky měří čas uplynulý od předchozího snímku a ticku, aktualizuje stav hry, pokud delta čas tiků dosáhne 1, a překreslí panel, pokud delta čas snímku dosáhne 1. Také sleduje snímky za sekundu (FPS) a ticks per second (TPS) a tiskne hodnoty každou sekundu.

Třída poskytuje metodu "aktualizace", která se volá v rámci herní smyčky k aktualizaci stavu hry.

Metoda "paintComponent" je přepsána pro malování herních komponent na panelu. Volá metodu "Draw" instance "Cell", aby nakreslila buňku na grafický objekt.

Metoda "startGameLoop" se používá ke spuštění herní smyčky vytvořením nového vlákna a jeho spuštěním.

Celkově kód představuje herní panel, který zpracovává herní smyčku, aktualizuje stav hry a maluje herní komponenty na panelu. To naznačuje, že ve hře jsou další třídy a logika, jako je třída "Cell" a reprezentace herního pole, které nejsou součástí poskytnutého kódu.

GameWindow:

Celkově kód nastavuje a zobrazuje herní okno pomocí třídy JFrame. Konfiguruje různé vlastnosti okna a přidává do něj objekt "GamePanel". Herní smyčka se spustí voláním metody panelu „startGameLoop“.

Cell:

Třída rozšiřuje třídu "Position", která není uvedena ve fragmentu kódu.

Třída má proměnnou instance "gp" typu "GamePanel" představující odkaz na instanci GamePanel a ArrayList s názvem "single_cell" pro ukládání jednotlivých buněk.

Vytvoří instanci třídy "Logic" a předá "single_cell" ArrayList a "gp" odkaz pro zpracování logiky buňky.

Konstruktor inicializuje objekty Single_cell a přidá je do "single_cell" ArrayList. Přirazuje bomby náhodně k objektům Single_cell na základě procenta definovaného proměnnými "procent" a "value".

Konstruktor inicializuje proměnnou „logic“ s novou instancí třídy „Logic“ a předá odkaz „single_cell“ ArrayList a „gp“.

Konstruktor nastaví výchozí hodnoty pro pozici a stav a načte obrázky ze souborů pomocí třídy ImageIO.

Metoda "Draw" je zodpovědná za kreslení buněk na grafický objekt. Iteruje přes buňky, nastavuje umístění a velikost JButton přidruženého ke každé buňce, přidává JButton na GamePanel, nastavuje ikonu JButtonu na základě stavu buňky a kreslí příslušný obrázek na grafický objekt.

Kód obsahuje logiku pro různé stavy buněk, jako je „skrytý“, „zachycený“ a „zobrazený“. Také volá metody z třídy "Logic" pro zpracování operací souvisejících s buňkou, jako je nalezení vhodného obrázku pro buňku a odhalení sousedních buněk, pokud je buňka prázdná.

Celkově kód představuje třídu Cell zodpovědnou za správu jednotlivých buněk ve hře MineSweeper. Zvládá logiku pro přiřazování bomb, nastavení stavů buněk, načítání obrázků a kreslení buněk na grafický objekt. Spoléhá na třídu Logic pro další operace související s buňkou.

Single_cell:

Daný kód je třída Java s názvem „Single_cell“, která představuje jednotlivou buňku ve hře Hledání min. Zde je rychlé shrnutí toho, co kód dělá:

Třída implementuje rozhraní `MouseListener` a `ActionListener` pro zpracování událostí myši a akcí akcí.

Konstruktor bere `JButton` jako parametr a nastavuje výchozí hodnoty pro stav buňky a další vlastnosti. Také přiřadí poskytnuté `JButton` k proměnné instance "`jButton`".

Třída má různé metody pro nastavení umístění, velikosti a ikony `JButton` přidruženého k buňce. Má také metody pro nastavení a získání stavu buňky, počtu bomb a stavu bomby.

Třída poskytuje metody pro změnu stavu buňky, jako je skrytá, zobrazená a zachycená.

Třída definuje objekt `MouseListener` pro zpracování událostí myši na `JButton`. Detekuje kliknutí levým a pravým tlačítkem a volá příslušné metody objektu stavu buňky (např. `onLeftClick` a `onRightClick`).

Celkově kód představuje třídu `Single_cell`, která zapouzdřuje chování a vlastnosti jednotlivé buňky ve hře `MineSweeper`. Spravuje `JButton` přidružené k buňce, zpracovává události myši a aktualizuje stav buňky na základě interakcí uživatele.

Logic:

Třída přebírá `ArrayList` objektů `Single_cell` a objekt `GamePanel` jako parametry ve svém konstruktoru. Inicializuje objekt `Logic` voláním metod `findMineInTheField` a `getImage`.

Třída definuje proměnné `BufferedImage` pro ukládání různých obrázků představujících čísla (od 0 do 8) a používá třídu `ImageIO` k načítání těchto obrázků ze souborů.

Metoda `findImage` bere jako parametr objekt `Single_cell` a vrací příslušný obrázek na základě počtu bomb v buňce. Využívá příkaz `switch`, aby odpovídal číslu a vrátil odpovídající obrázek.

Metoda `findMineInTheField` prochází `ArrayList` objektů `Single_cell` a kontroluje sousední buňky, aby spočítal počet bomb kolem každé buňky. Podle toho aktualizuje vlastnost `numberOfBombs` každé buňky.

Metoda `ifCellsZero` bere jako parametry `ArrayList` objektů `Single_cell`, objekt `GamePanel` a `index`. Zkontroluje sousední buňky, které jsou nulové (žádné bomby) a spustí metodu `onLeftClick` příslušného objektu stavu buňky.

Celkově třída Logic zvládá logiku pro hledání sousedních bomb, jejich počítání a určování vhodných obrázků pro buňky ve hře MineSweeper. Pomáhá při aktualizaci stavu a chování buněk na základě pravidel hry a uživatelských interakcí.

4 Závěr

Logiku programu jsem napsal celou sám a co se týče grafiky tak tu jsem udělal podle youtubové série “How to Make a 2D Game in Java” od kanálu jménem “RyiSnow”, kterou jsem dokoukal do čtvrtého dílu následně jsem jeho kód modifikoval aby se dal použít k zadání. Nejdřív mi dělal problém generace mapy na té jsem strávil nejvíce času ale pak jsem ji celou předělal a vše fungovalo dobře, pak jsem měl problém s myší aby mi detekovala kliknutí a pravého tlačítka takže jsem vzal kód ze stánky java2s

```
import java.awt.BorderLayout;
import java.awt.Container;
import java.awt.event.InputEvent;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
import java.awt.event.MouseListener;
/*from w w w . j a v a 2 s . c o m*/
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.SwingUtilities;

public class Main {
    public static void main(String args[]) {
        JFrame frame = new JFrame("Button Sample");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        JButton button = new JButton("Select Me");

        MouseListener mouseListener = new MouseAdapter() {
            public void mousePressed(MouseEvent mouseEvent) {
                int modifiers = mouseEvent.getModifiers();
                if ((modifiers & InputEvent.BUTTON1_MASK) == InputEvent.BUTTON1_MASK) {
                    // Mask may not be set properly prior to Java 2
                    // See SwingUtilities.isLeftMouseButton() for workaround
                    System.out.println("Left button pressed.");
                }
                if ((modifiers & InputEvent.BUTTON2_MASK) == InputEvent.BUTTON2_MASK) {
                    System.out.println("Middle button pressed.");
                }
                if ((modifiers & InputEvent.BUTTON3_MASK) == InputEvent.BUTTON3_MASK) {
                    System.out.println("Right button pressed.");
                }
            }

            public void mouseReleased(MouseEvent mouseEvent) {
                if (SwingUtilities.isLeftMouseButton(mouseEvent)) {
                    System.out.println("Left button released.");
                }
                if (SwingUtilities.isMiddleMouseButton(mouseEvent)) {
                    System.out.println("Middle button released.");
                }
                if (SwingUtilities.isRightMouseButton(mouseEvent)) {
                    System.out.println("Right button released.");
                }
                System.out.println();
            }
        };

        button.addMouseListener(mouseListener);

        frame.add(button, BorderLayout.SOUTH);
        frame.setSize(300, 100);
        frame.setVisible(true);
    }
}
```

5 Zdroje

youtube série:

https://youtube.com/playlist?list=PL_QPQmz5C6WUF-pOQDsbsKbaBZqXj4qSq

link na java2s:

http://www.java2s.com/Tutorials/Java/Swing/JButton/Add_mouse_action_to_a_JButton_in_Java.htm

I