

Gymnázium, Praha 6, Arabská 14

Programování - ročníkový projekt

Vizualizace vícedimenzionálních těles



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, 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 Praze dne 26.dubna 2022vlastnoruční podpis.....

Obsah:

1. Anotace.....	1
2. Úvod	1
3. Zadání	1
4. Původní a finální ukládání	1
5. Zobrazení řezů	2
6. Dynamický počet os	3
7. Vkládání tělesa	4
8. Použité technologie	5
9. Závěr	5
10. Zdroje.....	6

1. Anotace

Tato okenní aplikace měla zobrazovat dvou a více dimenzionální tělesa pro usnadnění představy jejich tvaru formou rovinných průřezů v různých osách, což se zcela nepodařilo. Jde o způsob, jak předat informaci o tvaru tělesa pomocí několika průřezů na každé ose najednou, na úkor jeho plné podoby pouze na některých vybraných osách. Jsou zde popsány dvě datové struktury pro práci s předem neznámým počtem rozměrů.

2. Úvod

Tato okenní aplikace měla zobrazovat dvou a více dimenzionální tělesa pro usnadnění představy jejich tvaru formou rovinných průřezů v různých osách. Což je způsob, jak předat informaci o tvaru tělesa pomocí několika průřezů na každé ose najednou, na úkor jeho plné podoby pouze na některých vybraných osách. Bylo nutné řešit předem neznámý počet rozměrů, a tak přijít s flexibilní datovou strukturou. Vše bylo psáno s myšlenkou funkčnosti pro libovolný počet dimenzí a limit je určen pouze maximální hodnotou výběru v UI.

3. Zadání

Téma: Zobrazení objektů ve vyšších dimenzích

Popis: Program by měl uživateli umožnit procházet prostor různého počtu rozměrů / o různém počtu na sebe kolmých os, kdy si zvolí, ve kterých směrech uvidí spojitou výseč prostoru (roviny) a v jakých pouze body na odpovídající ose. Bude v něm také moci vložit a pracovat (posunout, skrýt, změnit barvu, ale to právě nevím, čeho jsem schopen, takže jsem se nechtěl zavazovat k ničemu více specifickému) s geometrickými útvary.

4. Původní a finální ukládání

Protože předem neznáme počet os a počet bodů na nich, nebylo možné vytvořit scénu najednou a namísto toho jsem vytvořil podobu řezů a řádků na vložení souřadnic a natočení kolem os jako samostatné fxml soubory, které se za běhu mohou načíst a zobrazit podle potřeby.

U standartního pole objektů je v kódu potřeba specifikovat, kolik má rozměrů, a tak mne napadlo vytvořit datovou strukturu polí postupným vkládáním posledního pole polí na indexy nového, ve které by se vyhledávalo pomocí pole indexů, určujících v jakém poli bude pokračovat vyhledávání. Obojí pole a řezy jsem nechal jako atributy objektu *DualType* pro úsporu podmínek.

```

private void arrIt() {
    DualType o = null;
    for (int i = 3; i <= dim; i++) {
        DualType p = new DualType();
        p.ay = new DualType[points];
        for (int j = 0; j < points; j++) {
            p.ay[j] = o;
        }
        o = p;
    }
    arr = o;
}

```

Obr. 1.1 Sestavení vícerozměrného pole

```

public Object getSlice(int[] indexTable) {
    DualType o = arr;
    for (int i = 0; i < indexTable.length; i++) {
        o = o.ay[indexTable[i]];
    }
    return o.cs;
}

```

Obr. 1.2 Hledání ve vícerozměrném poli

Kvůli zobrazování a vizuálnímu oddělování řezů však řezy už stejně musely být v několika vrstvách kontejnerů, které měly stejnou strukturu, a tak jsem přešel k vyhledávání přímo říkáním si o dítě kontejneru na určité pozici.

5. Zobrazení řezů

O načtení a rozložení jednotlivých řezů se stará rekurzivní metoda *boxIt*, jež do předaného kontejneru vloží určené množství, rovno zvolenému počtu bodů na ose, a to buďto již řezů, nebo dalších kontejnerů a to tak, aby do HBoxu šly VBoxy a naopak, tvoříc čtvercovou síť řezů oddělených pro přehlednost.

Při vytváření této struktury se zároveň předávají postupně se upřesňující souřadnice k popisu řezu ve výsledku podobné cestě k vyhledávání v původním poli. Tak jsem se rozhodl proto, že původně mne ke generování cest k zpětnému průchodu strukturou napadlo počítání po jedné v soustavě se základem rovným počtu bodů, jehož implementace nevyšla a kótování hned při vytváření řezu je úspornější.

```

private void boxIt(int level, Pane parent, int[] labeling) {
    /*
    Pokud level == 2 nebo points == 1 přidáváme CordedSlice a je jich stejný počet jako points.
    Buďto jsme totiž v nejvnitřnější vrstvě nebo je není třeba vytvářet pro jediný průřez.
    Ukončujeme před dalším vytvářením Boxů.
    */
    try {
        if (level == 0 || points == 1) {
            Pane p = FXMLLoader.load(getClass().getResource("CorddedSliceXML.fxml"));
            parent.getChildren().add(p);

            Label l = (Label) p.getChildren().get(1);
            l.setText("");
            for (int i = 0; i < labeling.length; i++) {
                l.setText(l.getText().concat(Double.toString( (labeling[i] * vR)))); //times distar
                if (i != labeling.length - 1){
                    l.setText(l.getText().concat(" ; "));
                }
            }
            //l.setText(Integer.toString(labeling[0]));
            return;
        }
    } catch (IOException e) {
        System.out.println("Can not load \"CorddedSliceXML.fxml\"");
    }
    /*
    Je-li ještě třeba vytvářet vrstvy (tzn. neplatí předešlé), přidáme nový Hbox, nebo VBox
    podle toho, zda, s ním, bude počet vrstev (zevnitř) sudý, nebo lichý, respektive.
    Volá pro něj boxIt za každý prvek jemu náležící (= points).
    */
    Pane m;
    if (level % 2 == 0) {
        parent.getChildren().add(m = new VBox());
        //m.setStyle("-fx-border-width: 2;" + "-fx-border-color: #AAAA00");
        /*
        Existuje-li více VBoxů (které jsou vždy naplněny HBoxy) na stejné vrstvě,
        je třeba je orámovat, tvoří tak jednotlivé prvky i vzhledově a dají se skládat za sebe.
        */
        if (level != dim - 2) {
            m.setStyle("-fx-border-width: 2;" + "-fx-border-color: #AAAA00");
        }
    } else {
        parent.getChildren().add(m = new HBox());
    }
    level -= 1;
    for (int i = 1; i <= points; i++) {
        labeling[level] = i - 1;
        boxIt(level, m, labeling);
    }
}

```

Obr. 2 Generace a zobrazení řezů

6. Dynamický počet os

Při potvrzení nastavení se opraví počet řádek pro zadávání souřadnic a to přidáváním a ubíráním HBoxů s TextField, cyklus od jedné po větší z počtu rozměrů a počtu řádků pro zadávání, překročí-li počet rozměrů a pokračuje, ubírá nadbytečné řádky, pokračuje-li nad počet řádků, začne je přidávat.

```

private void correctCords() {
    int children = coordinates.getChildren().size();

    for (int i = 1; i <= Math.max(dim, children); i++) {
        if (i > dim) {
            coordinates.getChildren().remove(dim);
        }
        if (children < i) {
            try {
                HBox m = FXMLLoader.load(getClass().getResource("EnterCordsXML.fxml"));
                //přidání právě načteného HBoxu m
                coordinates.getChildren().add(m);
                //vezme Label, dítě od m, a opraví jeho text
                Label l = (Label) m.getChildrenUnmodifiable().get(0); //cu
                l.setText("t" + i + ": ");
            } catch (IOException e) {
                System.out.println("EnterCordsXML.fxml file not found!");
            }
        }
    }
}

```

Obr. 3 Oprava počtu os

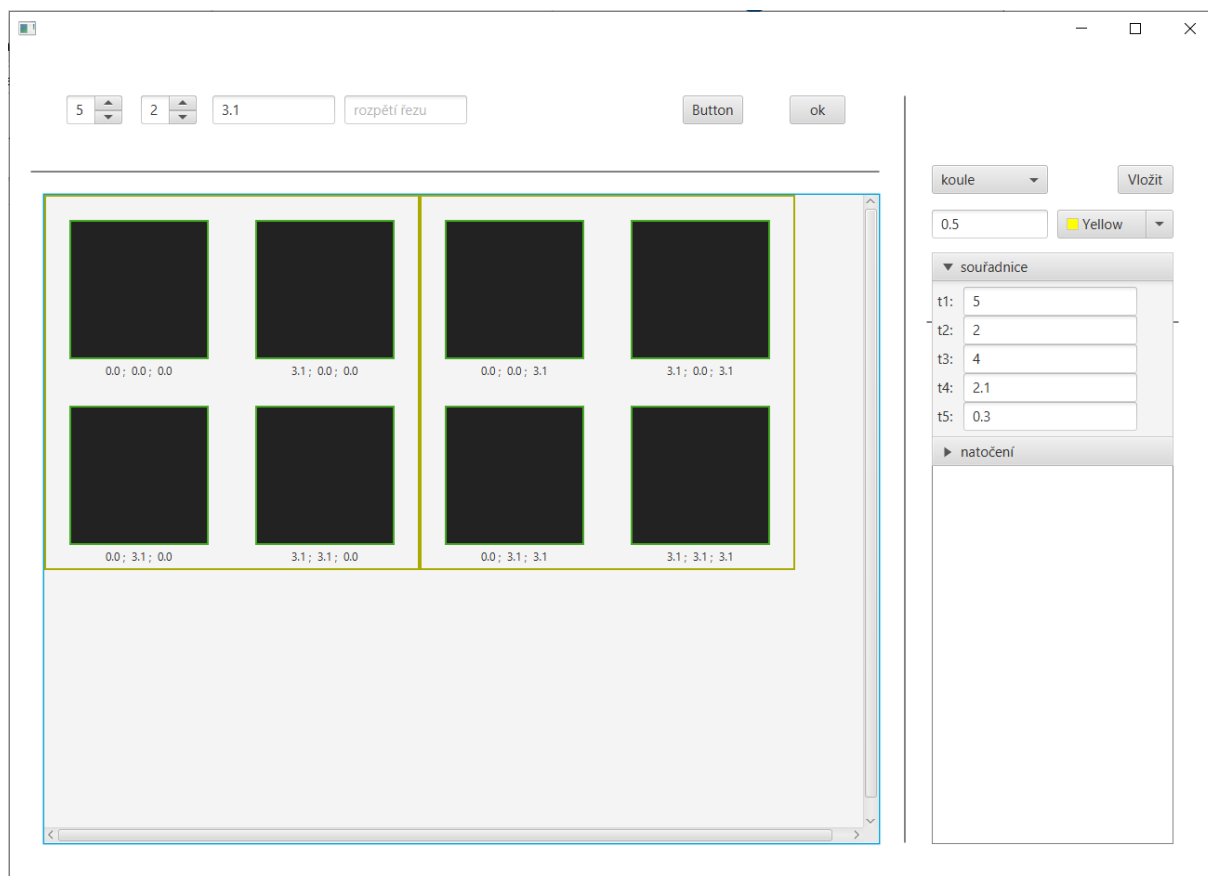
7. Vkládání tělesa

Při potvrzení vložení se měla přečíst data o tělesu jako jsou souřadnice, natočení, velikost, barva a vůbec jaký je to tvar, na kterém by záviselo, jaká část se dál spustí. Pro kouli by se při průchodu řezů spočítala vzdálenost řezu od středu koule a z toho poloměr koule k vykreslení. Pro krychli by se spočítaly souřadnice určující její hrany a pak při průchodu řezy by se zkontrolovalo jaké řez protínají, popřípadě náleží. Při implementaci zpětného čtení kót, souřadnic, řezů jsem ale nemohl přijít na příčinu chyby, vždy buďto prázdný String k parsování na double, nebo čtení indexu mimo pole.

instalace, použité technologie

Instalace a návod k používání

Uživatel musí mít prve nainstalovanou Java Runtime Environment 1.8.0 nebo novější. Poté na <https://github.com/gyarab/2021-2e-Liska-OuterSight> si stáhne OuterSight.jar a tento soubor stačí otevřít.



Obr. 3 Uživatelské rozhraní

Na horní liště zleva viz. obr. 3 najdeme výběr počtu dimenzí, počet řezů na osách, k zadání, vzdálenost řezů od sebe v jedné ose a rozpětí řezu (vzdálenost od horního k dolnímu rohu řezu). *Button* používám k testování funkcí, *Ok* potvrzuje zvolené nastavení na této liště.

Na pravém sloupci shora dolů, prvně je výběr tělesa (koule nebo krychle), *Vložit* má zahájit vykreslování s údaji zvolenými uživatelem. Dále je zadání velikosti tělesa, jeho barva, *souřadnice* středu tělesa na jednotlivých osách (označených t1 až tx) a *natočení* tělesa kolem jednotlivých os.

8. Použité technologie

Kód jsem psal v NetBeans IDE 1.8.2, FXML soubory generoval pomocí SceneBuilder-8.5.0. Knihovny: javafx.scene, javafx.fxml

9. Závěr

Aplikace by jistě byla přínosná jako nástroj pro lepší představu o tvaru tělesa ve více dimenzionálním prostoru, pokud by fungovala. V projektu se mi podařilo vytvořit efektivní nástroj pro rozmístění mnoha objektů (Node) na scéně podle souřadnic. Původní vícerozměrné pole bude určitě užitečné i v jiných projektech.

10.Zdroje

1. <https://stackoverflow.com/questions/38248442/insert-the-same-object-several-times-in-fxml-file-javafx> dne 12.06.2022
2. <https://stackoverflow.com/questions/54888992/getting-children-of-a-node-after-fxml-load> dne 3.07.2022
3. <https://www.youtube.com/watch?v=hSTEVJe4HSE> dne 10.06.2022
4. http://www.java2s.com/Tutorials/Java/JavaFX_How_to/HBox/Add_blue_border_to_HBox.htm dne 02.08.2022
5. <https://stackoverflow.com/questions/33683302/loading-fxml-file-in-main> dne 02.08.2022
6. <https://stackoverflow.com/questions/15453959/javafx-2-pane-dynamic-resizing> dne 15.08.2022
7. <https://stackoverflow.com/questions/17145392/adding-items-to-choicebox> dne 14.06.2022