

Mapa USA

Základy čtení dat, zobrazování a interakce s daty. Na mapu USA dáme vizualizaci datové sady, kterou máme definovanou pro 50 států USA.

Přidání mapy do sketche

Najděte si v průzkumníku soubor *map.png* a pomocí Drag and Drop jej přetáhněte do okna Processingu – sketche. Dojde tak k automatickému prolinkování sketche a souboru. Další možností jak soubory přidat je přes *Menu -> Sketch -> Add file*.

Vykreslení obrázku mapy

Třída *PImage* je kontejner pro uložení odkazu na obrázek. Ve funkci *setup()* si načteme obrázek mapy a následně jej v *draw()* vykreslíme na pozici bodu (0,0), aby nám obrázek zabral celé okno.

```
PImage mapImage;
void setup( ) {
  size(640, 400);
  mapImage = loadImage("map.png");
}
void draw( ) {
  background(255);
  image(mapImage, 0, 0);
}
```

Specifikace bodů na mapě

V souboru *locations.tsv* jsou specifikovány body středů pro každý stát. Budeme potřebovat data načíst do sketche, k tomu slouží kód v souboru *Table.pde*. Třída *Table* načítá datový soubor jako grid řádků a sloupců. Pro přístup např. k float hodnotě můžete použít následující funkci:

```
table.getFloat(row, column)
```

Řádky a sloupce jsou číslovány od 0, čili hlavičky sloupců (pokud jsou) budou mít atribut *row* roven 0.

Vytvoříme si objekt *locationTable* a pro čtení hodnot souřadnic z řádků použijeme funkci *locationTable.getFloat()*. Následně na těchto pozicích vykreslíme kruh (elipsu se stejnou výškou a šířkou). Updateovaný kód (části **boldem** jsou nové):

```
PImage mapImage;
Table locationTable;
int rowCount;

void setup( ) {
  size(640, 400);
  mapImage = loadImage("map.png");
  // Make a data table from a file that contains
```

```

    // the coordinates of each state.
    locationTable = new Table("locations.tsv");
    // The row count will be used a lot, so store it globally.
    rowCount = locationTable.getRowCount();
}
void draw( ) {
    background(255);
    image(mapImage, 0, 0);
    // Drawing attributes for the ellipses. They affect
    //all subsequent drawings.
    smooth();
    fill(192, 0, 0);
    noStroke();
    // Loop through the rows of the locations file and draw the points.
    for (int row = 0; row < rowCount; row++) {
        float x = locationTable.getFloat(row, 1); // column 1
        float y = locationTable.getFloat(row, 2); // column 2
        ellipse(x, y, 9, 9);
    }
}

```

Vizualizace dat

Elipsy už máme vykreslené a rádi bychom namapovali nějaké hodnoty na vizuální proměnné, které se nabízí – to jsou velikost a barva. Zaměříme se na změnu velikosti elips v závislosti na hodnotách v datové sadě.

Do sketchu si přidáme další *Table* object *dataTable* a načteme do něj data ze souboru *random.tsv*. Je potřeba zjistit, jaké je v sadě minimum a maximum, abychom mohli rozsah dat namapovat na rozsah velikostí elips, jaké si stanovíme – a hlavně – vizualizovat jej.

Využijeme built-in hodnot *MIN_FLOAT* a *MAX_FLOAT* pro inicializaci proměnných *dataMin* a *dataMax* abychom zaručili, že tyto hodnoty budou přepsány první větší/menší hodnotou z datové sady. Kód pro zjištění minima a maxima z datové sady, až na inicializaci proměnných je umístěn ve funkci *setup()*:

```

Table dataTable;
float dataMin = MAX_FLOAT;
float dataMax = MIN_FLOAT;

// Read the data table.
dataTable = new Table("random.tsv");
// Find the minimum and maximum values.
for (int row = 0; row < rowCount; row++) {
    float value = dataTable.getFloat(row, 1);
    if (value > dataMax) {
        dataMax = value;
    }
    if (value < dataMin) {
        dataMin = value;
    }
}

```

Na závěr chceme vykreslit datový bod pro každý stát. Vytvoříme si funkci *drawData()*, která vezme x a y souřadnice a zkratku státu jako parametry. Zkratka se nachází v prvním sloupci tabulky, dá se o ní tedy uvažovat jako o „hlavičce řádků“. Stejně tak můžeme modifikovat funkci *getFloat()* aby brala zkratku státu místo čísla řádku.

V následující části je také aktualizovaná funkce *draw()*.

```
void draw( ) {
    background(255);
    image(mapImage, 0, 0);
    smooth( );
    fill(192, 0, 0);
    noStroke( );
    for (int row = 0; row < rowCount; row++) {
        // getRowName is equivalent to getString(row, 0)
        String abbrev = dataTable.getRowName(row);
        float x = locationTable.getFloat(abbrev, 1);
        float y = locationTable.getFloat(abbrev, 2);
        drawData(x, y, abbrev);
    }
}

// Map the size of the ellipse to the data value
void drawData(float x, float y, String abbrev) {
    // Get data value for state
    float value = dataTable.getFloat(abbrev, 1);
    // Re-map the value to a number between 2 and 40
    float mapped = map(value, dataMin, dataMax, 2, 40);
    // Draw an ellipse for this item
    ellipse(x, y, mapped, mapped);
}
```

Úkol

Zkuste modifikovat mapování dat tak, aby byla použita vizuální proměnná barva.

