Obsahovadlo

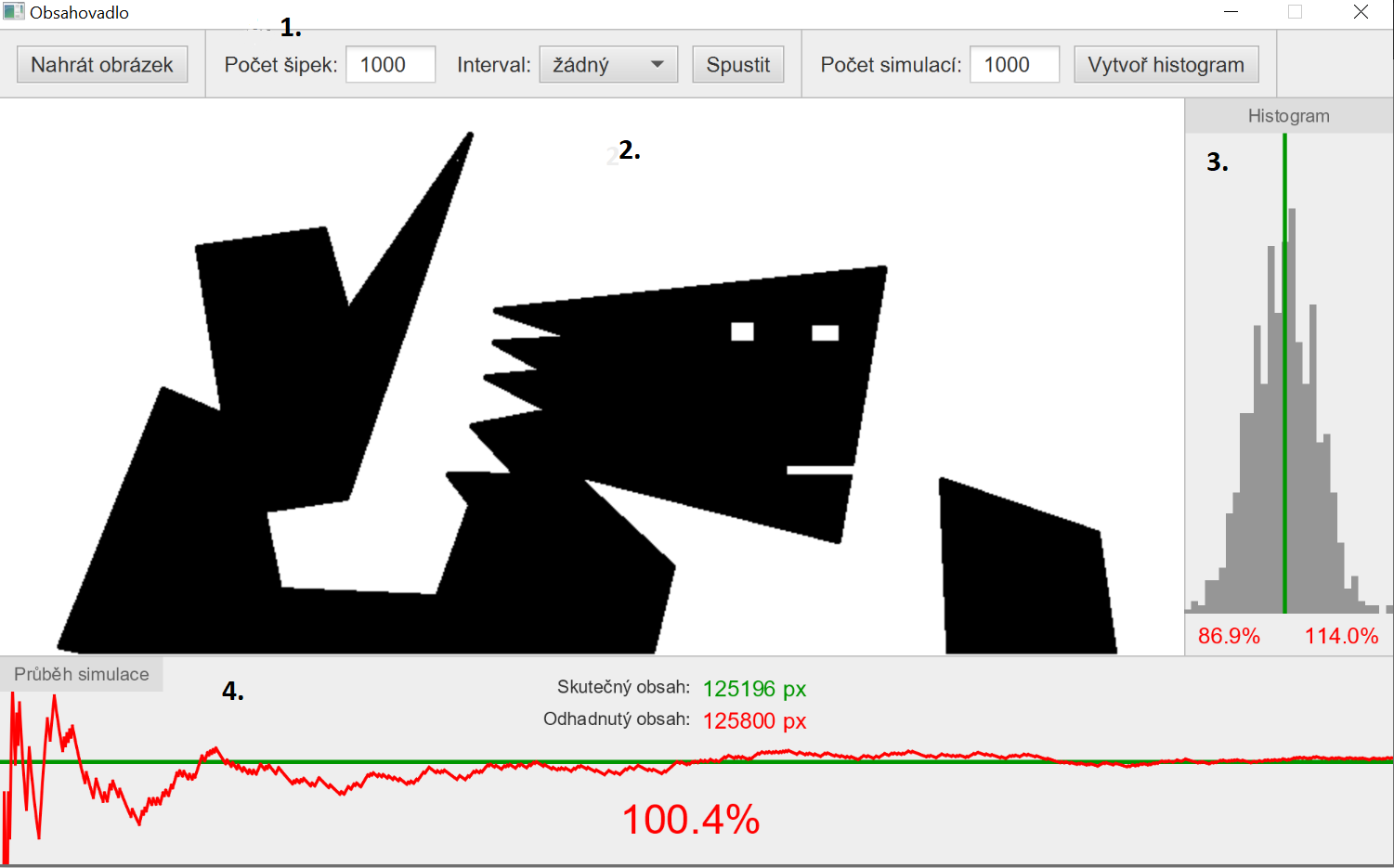
Autor: Ondra Hamala

# Úvod

Obsahovadlo je program, který využívá rovnoměrně rozložených náhodných čísel pro aproximaci obsahu různých planimetrických útvarů. Jedná se o zjednodušení běžně používané numerické metody Monte Carlo, což je numerická metoda, která slouží k výpočtu integrálů, zejména těch několikarozměrných a natolik komplexních, že jejich hodnota nelze analyticky určit.   
 Obsahovadlo je tedy program, který vám numericky, přibližně, spočítá obsah vámi nahraného rovinného útvaru. Vámi nahraný útvar musí být v rastrovém formátu, protože Obsahovadlo spočítá podle pixelů přesný obsah vámi nahraného útvaru a porovná ho s obsahem odhadnutým. Tuto komparaci dvou obsahů můžete sledovat v reálném čase po spuštění výpočtu v grafu, více se dozvíte v uživatelské dokumentaci.

# Uživatelská dokumentace

V této části se pokusím popsat všechny funkce, které můj program uživateli nabízí. Napadlo mě, že to udělám systémem takovým, že rozdělím aplikační okno na několik částí, podobně jako v kódu ho rozděluje LayoutView a každou z nich pak popíšu.



## Hlavní menu

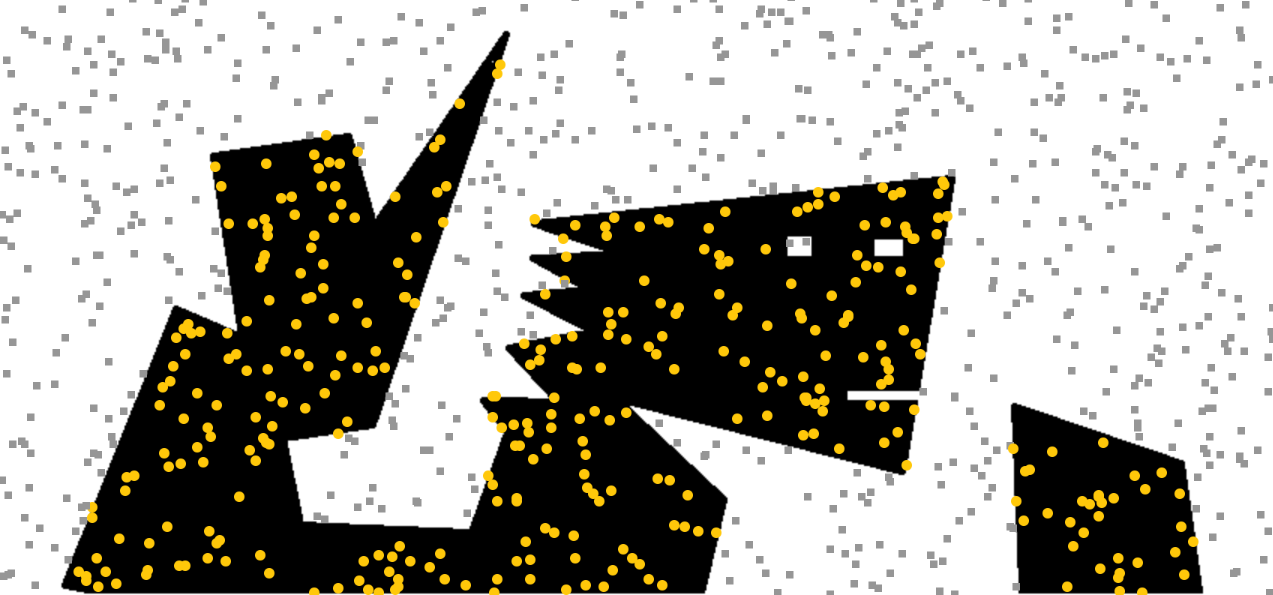


V hlavním menu se přirozeně nachází hlavní nastavení celého programu pro uživatele.

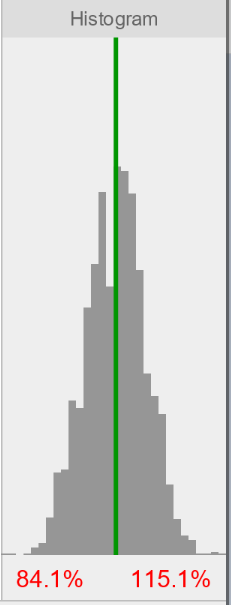
* Z leva doprava je první tlačítko „Nahrát obrázek“ (1.), to umožňuje nahrát uživateli obrázek. Pokud se nepodaří obrázek nahrát, program na to uživatele upozorní. Maximální velikost obrázku je 800x450 px, pokud bude obrázek moc velký, program na to také upozorní. Menší obrázek program přijme.
* Další je v pořadí box „Počet šipek“ (2.), hodnota v tomto boxu určuje, kolik náhodných bodů se bude generovat pro danou simulaci případně pro každou simulaci v histogram módu.
* Pak si uživatel může zvolit možnost v jakém „Intervalu“ (3.) chce, aby se jednotlivé náhodné body generovali. Tato možnost však funguje u zvolení jediné simulace. V módu histogramu je z důvodu časové náročnosti automaticky možnost ihned.
* Tlačítko „Spustit“ (4.) pak spouští samotnou simulaci o které se dozvíte více v druhém bodě.
* V menu následuje sekce pro nastavení histogramu, v něm je možné zvolit „Počet simulací“ (5.) z jejichž výsledků bude následně histogram sestaven.
* Tvorba histogramu započne po stisknutí tlačítka „Vytvoř histogram“ (6.).

## Plocha pro simulaci

Tato část je určena pro obrázek nahraný uživatelem a pro samotnou simulaci. V případě, že zvolím samostatnou simulaci se mi začnou generovat náhodné body na této ploše. Odhadnutý obsah pak určím tak, že znám přesný obsah této plochy a relativně zjistím, kolik náhodných bodů mi spadlo do zabarvené části, potom akorát vynásobím tímto koeficientem celkový obsah této plochy a mám přibližný odhadnutý obsah. Celý tento proces generování náhodných bodů (body s náhodnými souřadnicemi) je uživateli graficky vykreslen.

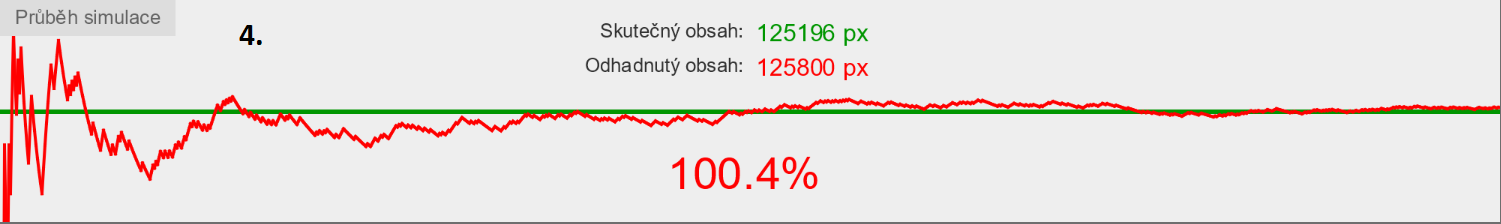


## Histogram graf

Pokud si uživatel zvolí v hlavním menu možnost histogramu, objeví se v této části aplikačního okna histogram. Histogram zobrazuje četnost výsledků v daném intervalu z X simulací. Počet simulací určuje uživatel. Pod histogramem jsou relativně vyjádřeny horní a spodní odchylky. Interval mezi těmito odchylkami je rozdělen na 30 podintervalů. Do každého tohoto podintervalu pak spadají příslušné výsledky ze simulací. Zelená vertikální čára v histogramu ukazuje, kde se na ose X nachází hodnota reálného obsahu. Nejvyšší sloupec histogramu by měl být pod touto zelenou čárou, jelikož nejvíce výsledků by mělo spadnout do podintervalu, který obsahuje hodnotu reálného obsahu. Při volbě histogramu se graficky nezobrazují náhodné body (protože tento proces se odehrává až několiktisíckrát), ale místo toho se v ploše pro obrázek zobrazí počítadlo, které říká, kolik simulací už počítač pro tvorbu histogramu vyhodnotil.

## Timeline graf

Tento graf funguje pouze při volbě samostatné simulace a ukazuje, jak se mění odhadnutý obsah v průběhu simulace. Zelená horizontální čára ukazuje reálný obsah a červená čára, která by se ideálně měla k zelené postupně přibližovat, ukazuje obsah odhadnutý. Dále v prostoru tohoto grafu máme uvedený přesný obsah, výsledný odhadnutý obsah a na kolik procent odpovídá odhadnutý obsah tomu skutečnému.



# Programátorská dokumentace

Program Obsahovadlo je napsaný v jazyce Java a jeho kód je rozdělen do několika balíků. Každý z těchto balíků obsahuje alespoň jednu třídu.

## První balík – hlavní balík

První balík obsahuje jen jednu třídu, třidu main. Ta má na starosti inicializaci celého programu.

## Druhý balík – controller

Třída má v sobě dvě metody. První z nich se jmenuje **startSimulation** a spouští simulaci. Pozoruhodné je, že spouští simulaci ve vlastním vlákně. Hlavní aplikační vlákno dostane pomocné vlákno, které má na starosti výpočet, aby se mohla překreslovat během výpočtu obrazovka (aby se vykreslovaly šipky).  
 Dále obsahuje metodu **startHistogram**, která má za úkol spustit všechny simulace pro tvorbu histogramu podle toho, kolik jich uživatel zadal. Také používá pomocné vlákno, na obrazovce se překresluje během výpočtu pokrok v tvorbě histogramu. Bez vedlejšího vlákna by toto nešlo.  
 Dále má za úkol samotný histogram vytvořit. Ten vytvoří tak, že z výsledků všech proběhlých simulací vezme spodní a horní hodnotu. Tento interval mezi spodní a horní hodnotou pak rozdělí na 30 podintervalů, toto číslo je pevně nastaveno v konstantě HISTOGRAM\_RESOLUTION. Výsledné hodnoty ze všech simulací se pak rozdělí do těchto 30 intervalů. Výsledek se zabalí to objektu Result, ze třídy Result a pošle se do třídy HistogramView k vykreslení. Ta vykreslí histogram podle četnosti výsledných hodnot v daném intervalu. Pod vykresleným grafem jsou také zobrazeny relativní odchylky od skutečné hodnoty obsahu. Graf by měl mít v ideálním případě tvar Gaussovy křivky. Začíná se jí podobat tím víc, čím víc simulací a „šipek“ (náhodných čísel) pro každou simulaci zvolíme. Zatím nemám dostatečný matematický aparát, abych dokázal popsat, proč to tak funguje, ale plánuji si to doplnit.

## Třetí balík – model

### Třída Random

V této třídě se nachází generátor pseudonáhodných čísel, který jsem napsal pro generování náhodných čísel do mého programu. Kód jsem našel v anglické učebnici *Numerical Solution of SDE Through Computer Experiments (Kloden Peter, Platen Exkhard, Schurz Henri, str.8, 3. vydání).* Hodnoty v mém programu jsou stejné, jako v kódu v učebnici, až na seed. Seed pro můj generátor beru podle unix timestampu (jedná se o počet milisekund od 1. ledna 1970).

### Třída Result

Slouží k předání výsledků o histogramu z controlleru do view. Do tohoto objektu se výsledky v podstatě zabalí.

### Třída Settings

Tato třída obsahuje konstanty s hodnotami nastavení rozměrů GUI (uživatelské grafické rozhraní) a nastavení parametrů histogramu.

### Třída Simulation

Třída Simulation obsahuje samotnou simulaci se všemi jejími parametry. Rozlišuje, zda se jedná o simulaci jedinou nebo o simulaci histogramovou. Generuje postupně náhodné body, já je nazývám „šipkami“, protože vykreslování bodů při zvolení jediné simulace připomíná graficky házení šipek.

Čtrvtý balík – View

Tento balík obsahuje 5 tříd, z nichž má každá za úkol zobrazení jiné části aplikačního okna a pak jednu třídu LayoutView, která celé aplikační okno rozdělí na 5 částí jako BorderPane. Z nichž jednu část (left) nepoužívám. V jednotlivých třídách pak podle části BorderPanu vykresluji obsah pomocí canvasu a scénického grafu.

### Třída LayoutView

Na této třídě je zajímavé to, že obsahuje takzvaný singleton. Jedná se o to, že chci mít objekt LayoutView dostupný odkudkoli, takže si na začátku vyrobím objekt a uložím ho do statické proměnné. Potom následuje vytvoření veřejného getteru, abych měl přístup k potřebným hodnotách i z dalších tříd. Nicméně konstruktor se udělá privátní, aby nikdo nemohl vyrobit druhý LayoutView.

# Závěr

Přes všechny své funkce je Obsahovadlo zatím ve své základní verzi, ve které slouží k výpočtu přesného a odhadnutého obsahu obrazce/obrazců v rovině a jejich následné porovnání. Existují programy, které plně využívají generování náhodných čísel pro výpočet plochy pod křivkou (integrálů), objevil jsem například program/programovací jazyk Mathematica od společnosti Wolfram, kde metoda Monte Carlo funguje přehledně i s grafickou vizualizací. V Obsahovadlu tedy vidím spoustu potenciálu na vylepšení, který plánuji s lepším matematickým aparátem v navazujícím studiu na vysoké škole využít.  
 Například by se mi líbilo implementovat základní funkce malování do samotného programu, aby se obrázky nemuseli jen nahrávat. Chtěl bych dodělat matematicky složitější funkce a třeba vytvořit verzi pro výpočet plochy pod křivkou (integrálů). Také je stále prostor pro zlepšování grafického prostředí pro uživatele. Takže jak vidíte, prostoru, kam jít s programem dál, je opravdu mnoho a těším se, až se k němu po maturitě opět vrátím.