

Základy programování v jazyce Python pro střední školy

Lekce 20

Kreslení myši



Andrej Blaho

Ľubomír Salanci

Václav Šimandl

beta verze - 2019

V této lekci se naučíme používat zábavný způsob kreslení do grafické plochy pomocí myši.

1. Vytvoř nový program `mys.py` a přepiš do něj následující kód. Program poté spust'.

```
import tkinter

canvas = tkinter.Canvas()
canvas.pack()

def klik(mys):
    print(mys.x, mys.y)

canvas.bind('<B1-Motion>', klik)
```

Přibyl zde nový příkaz `canvas.bind`, díky němuž bude od této chvíle grafická plocha vědět, co má udělat, když nad ní stiskneme levé tlačítko myši a myši potom táhneme. V textovém okně se začnou vypisovat dvojice celých čísel. Víš, jaká jsou to čísla?

2. Zápis `mys.x` a `mys.y` v programu `mys.py` označuje **x**-ovou a **y**-ovou souřadnici místa v grafické ploše, kde jsi klikl. Namísto příkazu `print` v podprogramu `klik` použij příkaz `canvas.create_text`, pomocí něhož vykreslí znak `'*'`. Znak by se měl vykreslit na pozici, kde jsi klikl myši:

```
import tkinter

canvas = tkinter.Canvas()
canvas.pack()

def klik(mys):
    _____

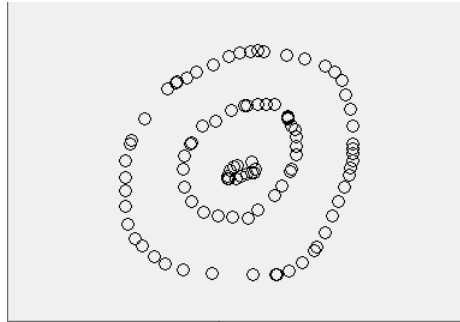
canvas.bind('<B1-Motion>', klik)
```

Program nyní při tažení myši kreslí malé hvězdičky. Pomocí parametrů `font='...'` a `fill='...'` můžeš velikost těchto znaků zvětšit na 50 a změnit jejich barvu na červenou. Jestli jsi postupoval správně, mělo by být možné vytvořit například takovýto obrázek:



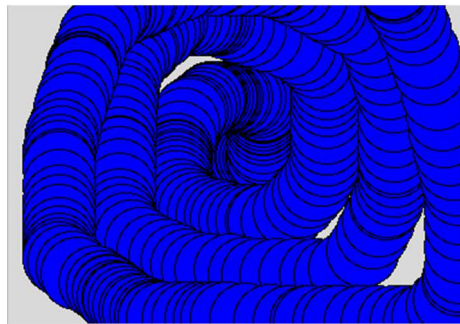
Zkus takto nakreslit i něco zajímavějšího a výsledným obrázkem se pochlub spolužákovi.

3. Pomocí příkazu `canvas.create_oval(x-5, y-5, x+5, y+5)` umíš nakreslit malý kruh se středem `[x, y]` a s poloměrem 5. V programu `mys.py` místo příkazu `canvas.create_text` vhodně použij příkaz pro kreslení malého kruhu. Nyní by se místo znaků '*' měly na místa, kudy jsi táhl myš, kreslit kruhy podobně jako na následujícím obrázku:

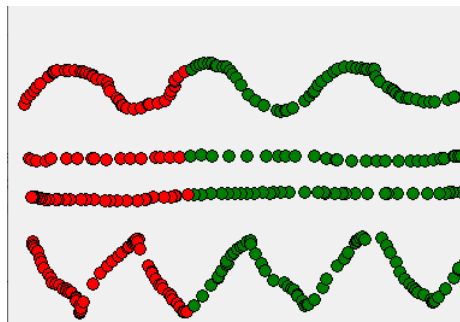


Ted' změň poloměr kreslených kruhů například na hodnotu 30. Jak se změní kreslené kruhy?

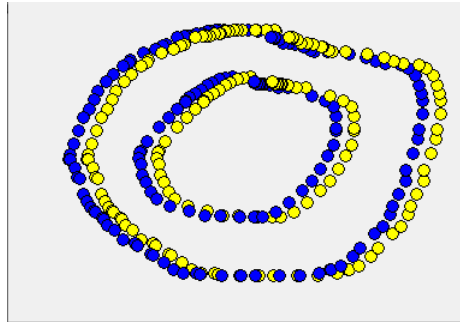
Co bys ještě musel změnit, aby bylo možné vytvořit kresbu jako na následujícím obrázku? Svou domněnku ověř.



4. Podprogram `klik` v programu `mys.py` se ještě předtím, než nakreslí barevný kroužek, může pomocí příkazu větvení rozhodnout, jestli bude kreslit červený nebo zelený kroužek. Uprav podprogram `klik` tak, aby se kroužky kreslily červeně, pokud je jejich `x`-ová souřadnice menší než 150; jinak se kreslily zeleně. Poloměr všech kroužků bude 5. Příklad fungování programu můžeš vidět na následujícím obrázku:



5. Vytvoř nový program `dvojite.py` a zkopíruj do něj kód z programu `mys.py`. Uprav v programu `dvojite.py` podprogram `klik` tak, aby kreslil všechny kroužky modře s poloměrem 5. Zajisti, aby se kromě modrého kroužku nakreslil i stejně velký žlutý kroužek. Jeho střed však bude o 15 posunutý vpravo (k x-ové souřadnici přičteš 15). Při tažení myši by měl vzniknout efekt jako na následujícím obrázku:



6. Vytvoř si nový program `odstranit.py` a zkopíruj do něj kód z programu `dvojite.py`. Přepiš do programu `odstranit.py` následující žlutě označený kód:

```
import tkinter

canvas = tkinter.Canvas()
canvas.pack()

def klik(mys):
    ...
    ...

def smaz(mys):
    canvas.delete('all')

canvas.bind('<B1-Motion>', klik)
canvas.bind('<ButtonPress-3>', smaz)
```

Nyní by mělo vše fungovat stejně, ale program bude také reagovat situaci, kdy do grafické plochy klikneš pravým tlačítkem myši. Zkus něco do plochy nakreslit a potom klikni do plochy pravým tlačítkem. Můžeš to opakovat i vícekrát. Diskutuj se spolužákem, co se po kliknutí pravým tlačítkem myši stalo.

Použil jsi tu nový příkaz `canvas.delete('all')`, pomocí kterého se z grafické plochy vymaže doposud vytvořená kresba.

7. Nyní se naučíme používat nový grafický příkaz `canvas.create_line(x1, y1, x2, y2)`. Pomocí něho lze nakreslit jednoduchou čáru (úsečku) z bodu `[x1, y1]` do bodu `[x2, y2]`. Vytvoř nový program `paprsky.py` a zkopíruj do něj kód z programu `odstranit.py`. V programu `paprsky.py` uprav podprogram `klik` podle následujícího kódu (změny jsou vyznačeny žlutě). Program poté spust.

```
import tkinter

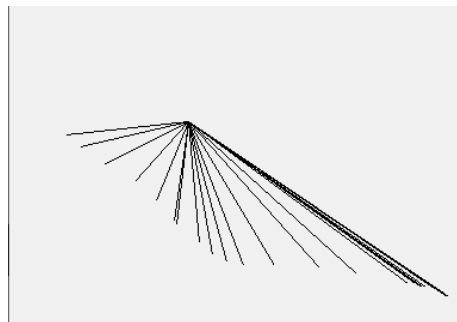
canvas = tkinter.Canvas()
canvas.pack()

def klik(mys):
    canvas.create_line(150, 100, mys.x, mys.y)

def smaz(mys):
    canvas.delete('all')

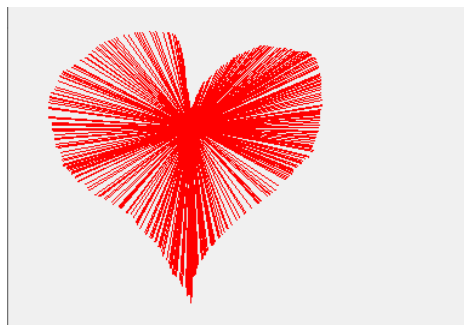
canvas.bind('<B1-Motion>', klik)
canvas.bind('<ButtonPress-3>', smaz)
```

Když program spustíš, můžeš s jeho pomocí vytvořit například takovýto obrázek:



Při tažení myši se kreslí úsečky z bodu `[150, 100]` do aktuální pozice myši. Proto mají všechny tyto úsečky společný jeden vrchol.

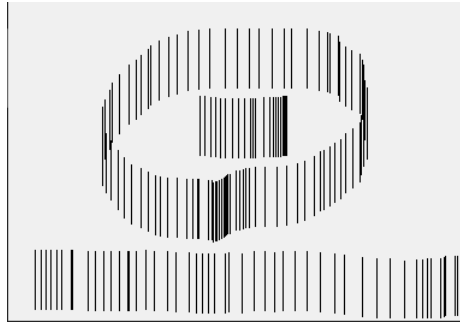
Dokázal bys nakreslit červené srdce jako je na následujícím obrázku? Barvu úsečky nastavíš stejně jako například barvu textu v příkazu `canvas.create_text` nebo barvu výplně v příkazu `canvas.create_oval`.



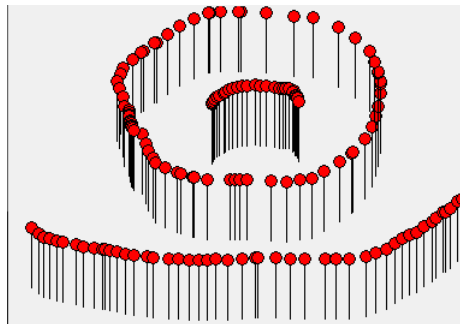
Výsledným obrázkem se pochlub spolužákovi.

8. Vytvoř nový program `spendliky.py` a zkopíruj do něj kód z programu `paprsky.py`. Uprav kód programu `spendliky.py` tak, aby každá úsečka začínala na pozici myši (`[mys.x, mys.y]`) a končila v bodě posunutém o 50 směrem vzhůru (`y`-ová souřadnice konce úsečky bude o 50 zmenšená).

Pomocí programu můžeš nakreslit například takovýto obrázek:



9. Uprav program `spendliky.py` tak, aby byl na konci každé úsečky nakreslen červený kroužek. Jestli jsi postupoval správně, mělo by být možné pomocí programu nakreslit například takovýto obrázek:

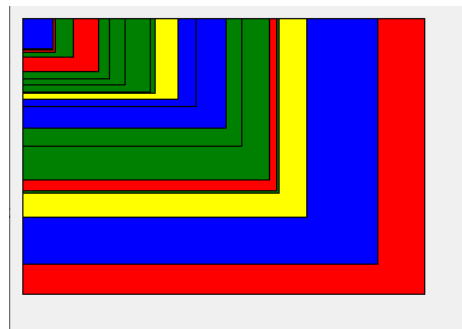


V dalším příkladu (na následující straně) budeš kreslit barevné obdélníky.

10. Vytvoř nový program `mys_obdelniky.py` a zkopíruj do něj kód z programu `spendliky.py`. V programu `mys_obdelniky.py` uprav podprogram `klik` tak, aby byl schopen nakreslit obdélník, jehož levý horní roh bude mít vždy souřadnice `[10, 10]` a pravý dolní roh bude na aktuální pozici myši `[mys.x, mys.y]`. Tento obdélník bude vybarvený náhodně zvolenou barvou. Pro náhodný výběr jedné ze čtyř barev můžeš využít následující kód:

```
barva = random.choice(['red', 'yellow', 'blue', 'green'])
```

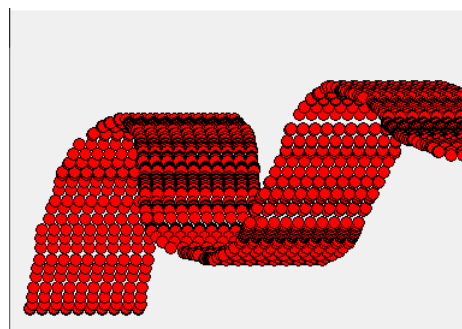
Při tažení myši by se ti mělo podařit vytvořit podobnou kresbu jako na následujícím obrázku:



11. Vrať se k programu `dvojite.py` a uprav v něm kód podprogramu `klik` tak, aby byl schopen kreslit 10 červených kroužků. Tyto kroužky budou nakreslené těsně vedle sebe:
- První kroužek bude na pozici myši
 - Každý další bude mít svůj střed posunutý o 10 vpravo oproti předchozímu kroužku (tedy `x`-ovou souřadnici zvětší o 10)

Vykreslení jednotlivých kroužků zajisti pomocí cyklu.

Při tažení myši by se ti mohlo podařit vytvořit podobnou kresbu jako na následujícím obrázku:



12* Vytvoř nový program `sprej.py` a zkopíruj do něj kód z programu `dvojite.py`. Nyní budeš upravovat program `sprej.py` tak, aby vznikl efekt spreje.

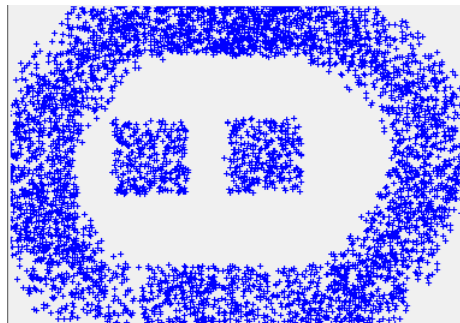
Nejprve před příkaz `canvas.bind` (mimo podprogram `klik`) zapiš kód

```
barva = 'blue'
```

Dále budeš upravovat podprogram `klik`:

- Nejprve se náhodně zvolí dvě čísla `dx` a `dy` z intervalu `<-30, 30>` (pomocí příkazu `random.randint(-30, 30)`)
- Tato dvojice čísel vyjadřuje posunutí nakreslené barevné tečky oproti pozici myši. Tečka tedy bude kreslena na pozici `[mys.x+dx, mys.y+dy]`
- Na tuto posunutou pozici nakresli tečku pomocí příkazu `canvas.create_text` jako znak `'+'`. Jako barvu kresleného znaku `'+'` použij proměnnou `barva`
- Postup z předchozích tří bodů při každém kliknutí zopakuj pomocí cyklu 50krát, čímž se nakreslí 50 malých znaků `'+'`, které nebudou příliš daleko od místa, kde jsi klikl

Program vyzkoušej. Jestli jsi postupoval správně, může vzniknout například takovýto obrázek:



Když budeš chtít změnit barvu spreje, stačí do příkazového řádku zapsat kód:

```
barva = 'yellow'
```

Od tohoto okamžiku bude sprej kreslit žlutou barvou.