

# projekt do předmětu PGR – Počítačová grafika

## 2D kreslení „jako v Google SketchUp“

řešitelé: **Vojtěch Šimeta**, xsimet00  
**Klára Vaňková**, xskota05

### Zadání

- Uživatelská aplikace pro vektorové 2D kreslení.
- Vytváření 2D prezentace modelů pomocí čar.
- Jednoduché uživatelské rozhraní.
- Chytré kreslení – přichytávání koncových bodů.
- Módy pro zarovnání kreslené čáry k rovnoběžkám a kolmicím.
- Nastavení délky kreslené úsečky.
- Výběr nakreslených objektů a jejich následná modifikace:
  - smazání,
  - transformace,
  - posun.
- Přiblížení / oddálení.
- Posun kreslicí plochy.
- Možnost vrácení změny (undo) a následné znovu provedení (redo).

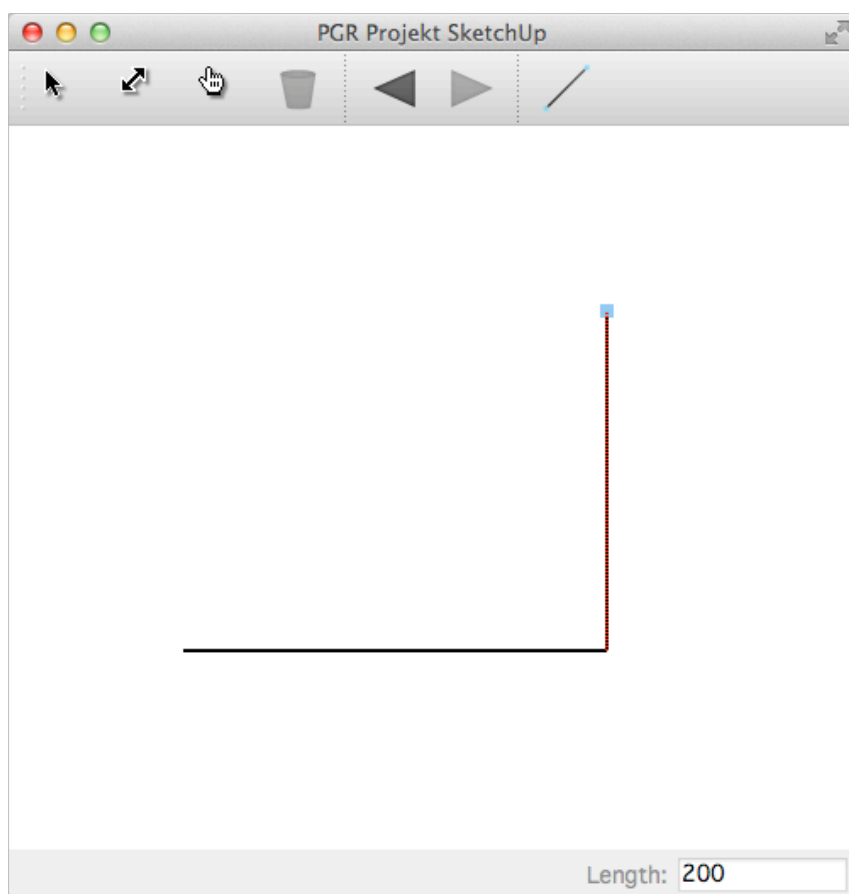
### Použité technologie

- Naprogramováno v C++.
- Použití knihovny OpenGL k vykreslování čar.
- Framework – Qt.
- Vývoj a testování na systému OS X 10.9.
- Dále testováno na **xxx!!!!!!!!!!!!!!!!!!!!!!!!!!!!**

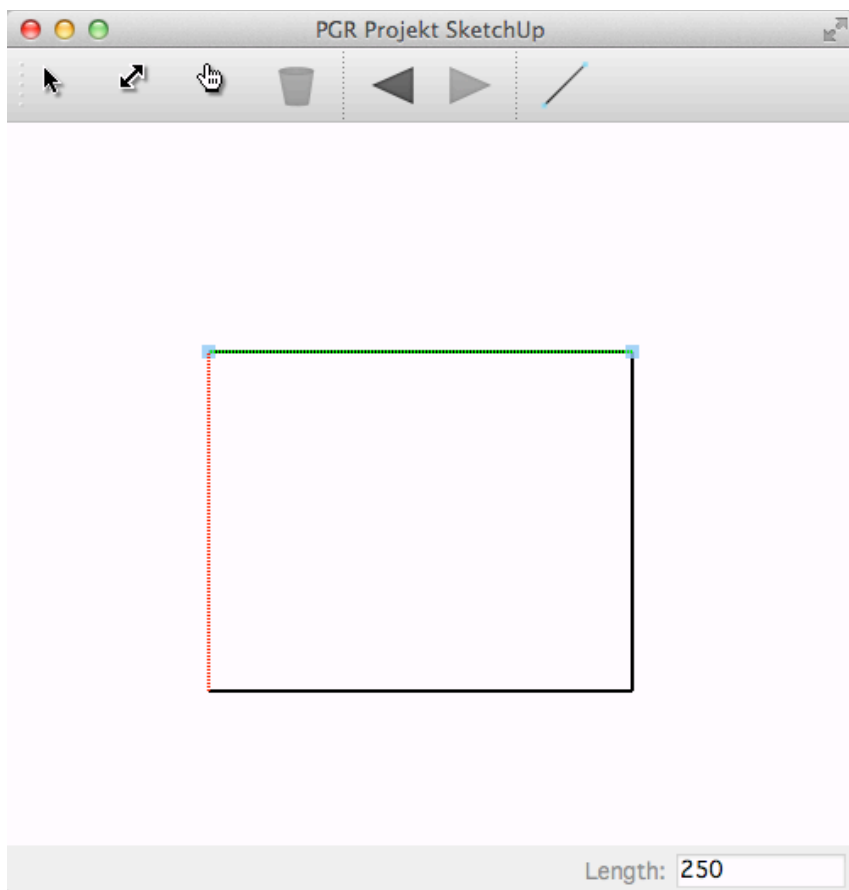
### Použité zdroje

**!!!!!!!!!!!!!!!!!!!!!!!!!!!!????????????**

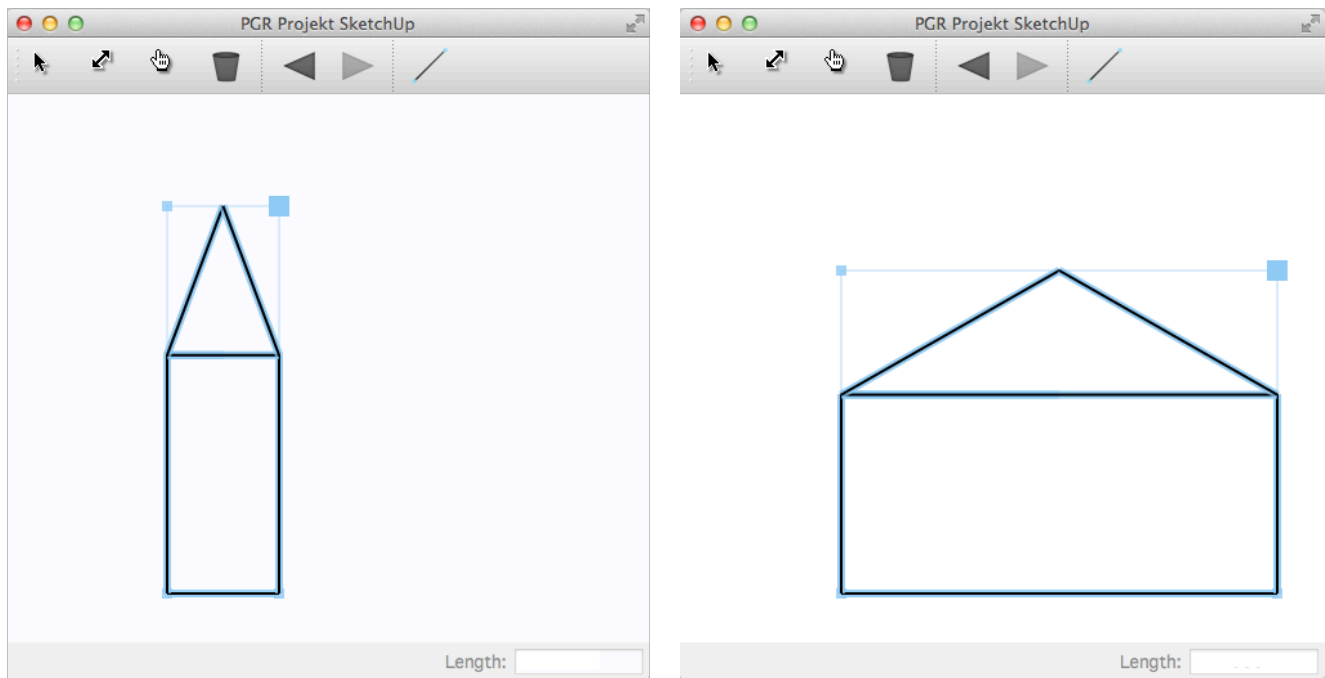
## Nejdůležitější dosažené výsledky



Obrázek 1: Nastavení délky kreslené úsečky.



Obrázek 2 Přichytávání kreslené úsečky k již vloženým objektům (body, osy, rovnoběžky, kolmice)



Obrázek 3 Transformace vybraných úseček.

## Ovládání vytvořeného programu

- Kreslení úsečky:
  - Tlačítko s úsečkou (Line) – přepnutí do stavu kreslení úsečky, zobrazují se vodící linky k již existujícím objektům.
  - Po kliknutí se začne vykreslovat úsečka a přichytávat se k osám X, Y a existujícím bodům.
  - Klávesa P : přichytávání úsečky k rovnoběžkám. Esc nebo P : ukončení přichytávání k rovnoběžkám.
  - Klávesa R : přichytávání úsečky ke kolmici. Esc nebo P : ukončení přichytávání ke kolmici.
  - Klávesy 0 – 9 : přepnutí do módu zadávání délky úsečky. Q nebo Enter : potvrzení zadané délky, Esc : storno zadávání délky.
    - Po zadání nové délky úsečka drží svoji novou velikost. Esc : ukončení pevné délky úsečky.
  - Po druhém kliknutí se ukončí kreslení úsečky a automaticky se začne kreslit druhá úsečka z koncového bodu úsečky předchozí.
  - Mezerník – ukončení kreslení úsečky.
- Výběr objektu:
  - Tlačítko se šipkou (Select) – klasický výběr vykreslených objektů.
  - Vybrané objekty lze:
    - transformovat (chytnutím rohu a tažením),
    - smazat (tlačítkem s košem - Delete),
    - pokud je vybrána pouze jedna úsečka, lze nastavit její délku (potvrzení nové délky klávesou Q nebo Enter)
- Posun kreslicí plochy:
  - Tlačítko s ručičkou (Pan).
  - Kliknutí a tažení.

- Zpět a vpřed:
  - Šipka zpět a vpřed (Undo a Redo).

## Rozdělení práce v týmu

Vojtěch Šimetka:

- Práce s kreslicí plochou (scaling, posun),
- výběr objektů,
- transformace objektů.

Klára Vaňková:

- Přichytávání kreslených objektů,
- změna délky úsečky,
- zpět a vpřed.

## Co bylo nejpracnější

Popište, co vám při řešení nejvíce komplikovalo život, s čím jste se museli potýkat, co zabralo čas.

Rozsah: 5-10 řádků

Složitější věci jsi podle mě řešil ty, tak to musíš říct, co bylo nejhorší..

## Zkušenosti získané řešením projektu

Popište, co jste se řešením projektu naučili. Zahrňte dovednosti obecně programátorské, věci z oblasti počítačové grafiky, ale i spolupráci v týmu, hospodaření s časem, atd.

Rozsah: formulujte stručně, uchopte cca 3-5 věcí

Možná tak GIT :D nevím, fakt nevím.. 2D kreslení OpenGL? To bych mu asi radši neříkala, matika? to radši taky ne :D

## Autoevaluace

**Technický návrh: 50%** (analýza, dekompozice problému, volba vhodných prostředků, ...)

Stručně (1-2 řádky) komentujte hodnocení.

**Programování: 90%** (kvalita a čitelnost kódu, spolehlivost běhu, obecnost řešení, znovupoužitelnost, ...)

Objektový návrh připravený k dalšímu rozšíření. GUI by bylo lepší psát ručně místo klikání v editoru Qt.

**Vzhled vytvořeného řešení: 90%** (uvěřitelnost zobrazení, estetická kvalita, vzhled GUI, ...)

Jednoduché GUI a kreslicí plátno. Bylo by dobré doplnit měřítko.

**Využití zdrojů: ---** (využití existujícího kódu a dat, využití literatury, ...)

Museli jsme si především oprášit středoškolskou matematiku. Jinak jsme algoritmy vymýšleli a psali sami. Nepoužili jsme žádný existující kód, ale vycházeli pouze z učebnic středoškolské matematiky.

**Hospodaření s časem: 60%** (rovnoměrné dotažení částí projektu, míra spěchu, chybějící části řešení, ...)

Začali jsme poměrně pozdě po zadání projektu. Stanovili jsme si cíle, které jsme v tomto čase schopni zvládnout v dobré kvalitě a tyto cíle jsme splnili. Kdybychom začali dříve, bylo by možné aplikaci rozšířit o další funkčnost.

**Spolupráce v týmu: 100%** (komunikace, dodržování dohod, vzájemné spolehnutí, rovnoměrnost, ...)

Na projektu jsme pracovali převážně společně v knihovně. Používali jsme verzovací systém GIT. Vzhledem k objektovému návrhu jsme museli velmi dobře sladit naši práci, pracovat paralelně bylo většinou nemožné.

**Celkový dojem: 80%** (pracnost, získané dovednosti, užitečnost, volba zadání, cokoliv, ...)

Byli jsme spokojeni se zadáním, protože jsme se vyhnuli pokročilému programování v OpenGL. Nakonec jsme v něm vykreslovali pouze jednoduché čáry a je na zvážení, zda je pro naše účely knihovna OpenGL vhodná. Projekt nás bavil, vytvořili jsme jednoduchý kreslicí program s omezenou funkcí, ale příjemným ovládáním. Rádi bychom se mu, nebýt ostatních školních povinností, věnovali déle a funkčnost rozšířili.

## Doporučení pro budoucí zadávání projektů

Po přečtení zadání nám nebylo jasné, zda je úkolem v OpenGL vykreslit ideální GUI pro kreslicí aplikaci, nebo naprogramovat kreslicí aplikaci. Nebylo nám zřejmé, jaké technologie můžeme použít a zda je cílem projektu vytvořit dobrou aplikaci nebo se naučit OpenGL. Na konzultacích jsme si vše ujasnili, ale něco takového by mohlo být jasné již ze zadání.