

GYMNÁZIUM JÍROVCOVA 8

MATURITNÍ PRÁCE

Programování grafické aplikace v C++

René Čakan

vedoucí práce: Dr. rer. nat. Michal Kočer

V Českých Budějovicích 14. února 2025

2025/2026

Prohlášení

Prohlašuji, že jsem tuto práci vypracoval samostatně s vyznačením všech použitých pramenů.

V Českých Budějovicích dne podpis

René Čakan

Abstrakt

Klíčová slova

Poděkování

Obsah

Úvod	1
I Teorie k vývoji hry v OpenGL	2
1 Programovací jazyky	3
1.1 Programovací jazyk C	3
1.2 Programovací jazyk C++	5
2 Počitačová grafika	7
2.1 Historie a architektura grafických karet	7
2.2 Grafická pipeline	8
2.3 Transformace a lineární algebra	8
3 OpenGL	9
3.1 Historie OpenGL	9
3.2 OpenGL pipeline	9
3.3 Shadery	9
3.4 Textury	9
4 to do	10
II Vývoj hry v OpenGL	11
5 Grafika a zvuk	12
5.1 Aseprite	12
5.2 Bosca Ceoil	12

6 Použité knihovny	13
6.1 GLAD	13
6.2 GLFW	13
6.3 stb_image	13
7 Herní scény	14
7.1 Scéna hlavního menu	14
7.2 Scéna pozastavené hry	14
7.3 Scéna hry	14
8 Serní mechaniky	15
8.1 Generování objektů	15
8.2 Pohyb hráče	15
8.3 Detekce kolize	15
Závěr	16
Bibliografie	19
Zkratky	20
Přílohy	23
A Fotky z pokusu	24
B Příloha další	25

Úvod

Část I

Teorie k vývoji hry v OpenGL

1 Programovací jazyky

1.1 Programovací jazyk C

C je středněúrovňový programovací jazyk, tedy jazyk, který je podobou blízko strojovému kódu, ale má už prvky vyššího programovacího jazyka jako jsou funkce, datové struktury nebo to že je strukturovaný. Je komplikaný a statický, což znamená, že se program musí nejdříve přeložit do strojového kódu a až pak se může spustit. Datové typy jsou známy v čase komplikace, proto všechny proměnné musí být v kódu deklarovány, jelikož vkládání vstupních dat do programu probíhá až při běhu programu. Programuje se v něm strukturovaně a procedurálně, tedy kód se píše pomocí řídících struktur(if, while, for atd.) a pomocí funkcí, které umožňují používat části kódu vícekrát. C nemá automatický správce paměti, takže je potřeba uvolňovat paměť manuálně. C má střídmou standardní knihovnu, která obsahuje základní matematické operace a funkce pro práci s pamětí a soubory, takže jakékoli složitější datové struktury či funkce si člověk musí naprogramovat sám. Tato strohost a blízkost ke strojovému kódu z C dělá jeden z nejrychlejších programovacích jazyků. [10, 8, 5, 12, 14]

C bylo vytvořeno Dennisem Ritchiem na počátku 70. let 20. století v AT&T Bellových laboratořích. Jeho předchůdci byly jazyky ALGOL, CPL, BCPL a B. Jeho prvním účelem bylo přepsat operační systém UNIX do použitelnějšího jazyka než Assembly a B. Už koncem 70. let bylo C populární, ale nebylo standardizované a vznikalo mnoho různých variant. Na začátku 80. let tedy Americký národní institut pro standardy(ANSI) zahájil práci na formální standardizované verzi. Tu dokončili v roce 1989 a je známa pod jménem C89. V průběhu let vycházely další verze, které jazyk zlepšovaly a modernizovali. Nejdůležitější verze byly C99, C11 a C17. Norma C23 byla nedávno schválena a ted' se implementuje do komplikátorů. V součastnosti mezi nejpoužívanější komplikátory patří GCC, Clang a MSVC. Jelikož bylo C velice populární, ovlivnilo řadu jiných programovacích jazyků, jako C, C#, Java, Rust, Go atd. [3, 6, 10, 8]

C je univerzální programovací jazyk, má tedy širokou škálu využití. Jeho první využití bylo k napsání UNIXu, který později ovlivnil operační systémy jako Linux, macOS, iOS a Android. Používá se v programování softwaru s omezenou pamětí a výkonem, jako je firmware aut či v zařízeních chytrých domácností. Dále se využívá pro tvoření kompilátorů a interpreterů jako je GCC nebo interpreter Pythonu. Také jsou v něm napsané systémové databáze MySQL a Oracle Database. Kvůli jeho rychlosti jsou v něm napsané knihovny pro jiné programovací jazyky jako je NumPy, OpenGL či GLFW. [3, 10, 24]

Jednoduchý program, který načte ze vstupu počet čísel, která chce uživatel setřídit. Následně daná čísla načte a vytiskne je seřazená:

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int compare(const void *a, const void *b)
5 {
6     return (*(int *)a - *(int *)b);
7 }
8
9 int main(void)
10 {
11     int sizeOfArray;
12     scanf("%d", &sizeOfArray);
13     int *array = malloc(sizeOfArray * sizeof(int));
14
15     for(int i = 0; i < sizeOfArray; ++i)
16         scanf("%d", &array[i]);
17
18     qsort(array, sizeOfArray, sizeof(int), compare);
19
20     for(int i = 0; i < sizeOfArray; ++i)
21         printf("%d\n", array[i]);
22
23     free(array);
24     return 0;
25 }
```

Zdrojový kód 1.1: sort_n_numbers.c

1.2 Programovací jazyk C++

Programovací jazyk C++ je v mnoha ohledech podobný jazyku C. Je stejně jako C středně-úrovňový, komplikaný, statický, má datové typy známé v době komplikace a nemá automatický správce paměti. V C++ se také programuje strukturovaně a procedurálně, ale na rozdíl od C, také umožnuje programovat objektově. Objektové programování umožnuje používat objekty, které jsou instance tříd. Tyto třídy umožňují dědičnost, polymorfismus a zapouzdření, což dělá kód přehlednější a usnadňuje budoucí rozšiřování a debuggování. Dalším rozdílem je standardní knihovna, kterou má C++ rozsáhlejší. Obsahuje nové kontejnery jako vector, map, a priority_queue, které jsou tvořeny pokročilejšími datovými strukturami jako binární vyhledávací strom nebo heap. Dále obsahuje nové algoritmy, například sort, find nebo count. Kvůli velké podobnosti C a C++ se často může C kód používat v C++, ale není tomu tak vždy. Například tento kód:

```
1| int class(int new, int bool);
```

Zdrojový kód 1.2: incompatibility_example.c

V C tento kód vytvoří funkci class, která vrací int a má 2 parametry new a bool. V C++ jsou ale class, new a bool klíčová slova, která nelze použít v názvu proměnných a funkcí. Pokud chce programátor napsat C kód, který se bude jednoduše v C++ programech, doporučuje se programovat v C tak, aby daný C kód byl podmnožinou C++. [22, 21]

C++ bylo vytvořeno Bjarnem Stroustrupem v roce 1979 v AT&T Bellových laboratořích. Před vytvořením C++ pracoval Stroustrup s programovacím jazykem Simula 67, který byl objektově orientovaný a sloužil primárně k vytváření simulací. Stroustupovi přišlo objektově orientované programování velmi užitečné, ale Simula 67 byl příliš pomalý pro větší projekty. Rozhodl se vytvořit nadmnožinu jazyka C, která by umožňovala objektově orientované programování a zároveň si zachoval rychlosť C, s názvem C with Classes. V roce 1982 byl Stroustrup se stavem C with Classes zklamán. Nepríšlo mu, že oproti C přináší významné zlepšení a rozhodl se jazyk dále vylepšovat nad rámec objektově orientovaného programování. V roce 1983 byl jazyk přejmenován z C with Classes na C++. Dále bylo C++ v roce 1985 oficiálně vydáno a začalo se používat komerčně. V roce 1998 byla vydána první standardizovaná verze s jménem C++98. Další významné verze, které jazyk modernizovaly a přidávaly mu nové funkce, byly C++03, C++11, C++14, C++17, C++20 a nejnovější

verze C++23. C++ se kompiluje pomocí stejných kompilátorů jako C, tedy GCC, Clang a MSVC. [21, 20, 2, 25]

C++ je stejně jako C univerzální programovací jazyk, a využívá se v široké škále odvětví. První využití je ve videoherním průmyslu. V C++ jsou napsané populární hrací enginy jako Unity nebo Unreal Engine. Dále se v něm vytváří aplikace jako Photoshop nebo Blender. Využívá se v částech operačních systémů jako Apple macOS nebo Microsoft Windows OS. Dále se využívá při vytváření internetových prohlížečů, například Firefox nebo Google Chrome. C++ se využívá i ve vědě, například v CERNu nebo v NASA. [22, 23, 19]

Program se stejnou funkcí jako z kapitoly o C, ale napsán v C++

```
1 #include <iostream>
2 #include <vector>
3 #include <algorithm>
4
5 int main()
6 {
7     int sizeOfArray;
8     std::cin >> sizeOfArray;
9     std::vector<int> arr(sizeOfArray);
10
11    for(int i = 0; i < sizeOfArray; ++i)
12        std::cin >> arr[i];
13
14    std::sort(arr.begin(), arr.end());
15
16    for(int i : arr)
17        std::cout << i << '\n';
18
19    return 0;
20 }
```

Zdrojový kód 1.3: sort_n_numbers.cpp

2 Počitačová grafika

2.1 Historie a architektura grafických karet

Není úplně jasné, který počítač jako první využíval počitačovou grafiku, ale začnu počítačem Small-Scale Experimental Machine(SSEM). Tento počítač byl vytvořen v roce 1948 na Manchesterské univerzitě a jeho tvůrci byli Frederic C. Williams, Tom Kilburn a Geoff Tootill. Tento počítač byl první počítač s uloženým programem, tedy počítač, který měl svůj program uložen ve stejné paměti jako data, se kterými počítač pracoval. Data uchovával na katodové trubici(CRT), které se později začalo říkat Williamsova trubice. Tato trubice si dokázala pamatovat až 2048 bitů, které byly uchovávány jako elektrické náboje. Součástí trubice byl i displej, který na svém fosforovém povrchu promítal oblasti s nábojem. Takto vznikl první počítač s digitálním displejem. [16, 13]

Americký Office of Naval Research a U.S. Air Force chtěli vytvořit počítač, který by dokázal běžet letecký simulátor. Tak tedy v roce 1944 začal Jay Forrester pracovat v laboratořích MIT na projektu Whirlwind. Při práci vyvinul Forrester paměť s náhodným přístupem(RAM), tvořenou magnetickými jádrami, skrz které proudil koindidenční proud. Počítač byl dokončen v roce 1951. Využíval CRT podobně jako baby a dokázal na svém displeji řešit rovnice, později i simulovat karetní hru blackjack. Následně se projekt Whirlwind stal součástí projektu Semi-Automatic Ground Environment(SAGE). Ten měl za úkol vytvořit počitačový systém, který by pomocí radarů dokázal odhalovat letadla a řídit obrané síly proti případným letadlům. Projekt SAGE byl jeden z prvních systémů, které využívaly interaktivní ovládání pomocí klávesnice či speciálního laserového pera. Pokud bylo světlem z tohoto pera posvíceno na ikonu letadla, počítač zobrazil informace o daném letadle jako rychlosť a směr jeho letu. [16, 11, 18]

2.2 Grafická pipeline

2.3 Transformace a lineární algebra

3 OpenGL

3.1 Historie OpenGL

3.2 OpenGL pipeline

3.3 Shadery

3.4 Textury

4 to do

istorie compileru C CRT

Část II

Vývoj hry v OpenGL

5 Grafika a zvuk

5.1 Aseprite

5.2 Bosca Ceoil

6 Použité knihovny

6.1 GLAD

6.2 GLFW

6.3 stb_image

7 Herní scény

7.1 Scéna hlavního menu

7.2 Scéna pozastavené hry

7.3 Scéna hry

8 Serní mechaniky

8.1 Generování objektů

8.2 Pohyb hráče

8.3 Detekce kolize

Závěr

Bibliografie

1. AKENINE-MÖLLER, Tomas et al. *Real-Time Rendering*. 4th. A K Peters/CRC Press, 2018.
2. ALBATROSS. *History of C++* [Online]. [B.r.]. Dostupné také z: <https://cplusplus.com/info/history/>. [citováno 2025-10-17].
3. BANAHAN, Mike; BRADY, Declan; DORAN, Mark. *The C Book: Featuring the ANSI C Standard(Instruction Set)*. 2nd. Addison-Wesley, 1991. ISBN 9780201544336.
4. ENGEL, Wolfgang. *The History of the GPU - New Developments*. 1st. Springer International Publishing, 2023. ISBN 9783031140464.
5. *Features of C* [Online]. 2025. Dostupné také z: <https://www.geeksforgeeks.org/c-features-of-c-programming-language/>. [citováno 2025-10-12].
6. HARBISON, Samuel P.; STEELE, Guy L. *C, a Reference Manual*. 5th. Prentice-Hall, 2002. ISBN 9780130895929.
7. HUGHES, John F. et al. *Computer Graphics: Principles and Practice*. 3rd. Addison-Wesley Professional, 2013.
8. KERNIGHAN, Brian W.; RITCHIE, Dennis M. *The C Programming Language*. 2nd. Pearson, 1988. ISBN 9780131103627.
9. KIRK, David B.; HWU, Wen-mei W. *Programming Massively Parallel Processors: A Hands-on Approach*. 3rd. Morgan Kaufmann, 2016.
10. KOCHAN, Stephen G. *Programming in C*. 4th. Pearson Education, 2015. ISBN 9780321776419.
11. *Milestones: Whirlwind Computer, 1944-59* [Online]. 2024. Dostupné také z: https://ethw.org/Milestones:Whirlwind_Computer,_1944-59. [citováno 2025-11-06].

12. MORTENSEN, Peter. *What's the difference between a low-level, midlevel and hight-level language* [Online]. 2023. Dostupné také z: <https://stackoverflow.com/questions/3468068/whats-the-difference-between-a-low-level-midlevel-and-high-level-language>. [citováno 2025-10-13].
13. NAPPER, Brian. *The Manchester Small Scale Experimental Machine – "The Baby"* [Online]. [B.r.]. Dostupné také z: <https://web.archive.org/web/20120604211339/http://www.computer50.org/mark1/new.baby.html>. [citováno 2025-11-05].
14. NOLLE, Tom. *What is structured programming(modular programming)* [Online]. 2023. Dostupné také z: <https://www.techtarget.com/searchsoftwarequality/definition/structured-programming-modular-programming>. [citováno 2025-10-14].
15. PEDDIE, Jon. *The History of the GPU - Eras and Environment*. 2nd. Springer Nature, 2023. ISBN 9783031135811.
16. PEDDIE, Jon. *The History of the GPU - New Developments*. 1st. Springer International Publishing, 2023. ISBN 9783031140464.
17. PEDDIE, Jon. *The History of the GPU - Steps to invention*. 1st. Springer International Publishing, 2023. ISBN 9783031109676.
18. SAGE: SEMI-AUTOMATIC GROUND ENVIRONMENT AIR DEFENSE SYSTEM [Online]. [B.r.]. Dostupné také z: <https://www.ll.mit.edu/about/history/sage-semi-automatic-ground-environment-air-defense-system>. [citováno 2025-11-06].
19. SRUTHY. *What Is C++ Used For? Top 12 Real-World Applications And Uses Of C++* [Online]. 2025. Dostupné také z: <https://www.softwaretestinghelp.com/cpp-applications/>. [citováno 2025-10-22].
20. STROUSTRUP, Bjarne. *Design and Evolution of C++, The*. 1st. Addison-Wesley Professional, 1994. ISBN 9780201543308.
21. STROUSTRUP, Bjarne. *Programming Principles and Practice Using C++*. 2nd. Pearson Education, 2014. ISBN 9780321992789.
22. STROUSTRUP, Bjarne. *The C++ Programming Language*. 4th. Pearson Education, 2013. ISBN 9780321563842.

23. *Top 25 C++ Applications in Real World[2025]* [Online]. 2025. Dostupné také z: <https://www.geeksforgeeks.org/blogs/top-applications-of-cpp-in-real-world/>. [citováno 2025-10-22].
24. *Top Applications of C programming* [Online]. 2024. Dostupné také z: <https://wscubetech.com/resources/c-programming/applications>. [citováno 2025-10-09].
25. VOLLE, Adam. *C++ computer language* [Online]. 2025. Dostupné také z: <https://www.britannica.com/technology/C-computer-language>. [citováno 2025-10-17].

Zkratky

CRT katodová trubice(cathode ray tube). 7, 10

RAM pamět s náhodným přístupem(random access memory). 7

Seznam obrázků

Seznam tabulek

Přílohy

A Fotky z pokusů

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, place-
rat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy
egest, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque ha-
bitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo.
Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus
eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra
ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla,
malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci
egest risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit
amet orci dignissim rutrum.

B Příloha další