

# GYMNÁZIUM JÍROVCOVA



## MATURITNÍ PRÁCE

Jak psát práci v L<sup>A</sup>T<sub>E</sub>Xu

Arnošt Thor

vedoucí práce: Mgr. František Kantor, PhD.

# Prohlášení

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

V Českých Budějovicích dne ..... podpis .....

Arnošt Thor

# Abstrakt

## Klíčová slova

## Poděkování

# Obsah

Úvod	1
<b>I Teorie k vývoji hry v OpenGL</b>	<b>2</b>
<b>1 Programovací jazyky</b>	<b>3</b>
1.1 Programovací jazyk C . . . . .	3
1.2 Programovací jazyk C++ . . . . .	5
<b>2 Grafická karta</b>	<b>6</b>
2.1 Historie grafických karet . . . . .	6
2.2 Architektura grafických karet . . . . .	6
2.3 Grafická pipeline . . . . .	6
2.4 Transformace a lineární algebra . . . . .	6
<b>3 OpenGL</b>	<b>7</b>
3.1 Historie OpenGL . . . . .	7
3.2 OpenGL pipeline . . . . .	7
3.3 Shadery . . . . .	7
3.4 Textury . . . . .	7
<b>II Vývoj hry v OpenGL</b>	<b>8</b>
<b>4 Grafika a zvuk</b>	<b>9</b>
4.1 Aseprite . . . . .	9
4.2 Bosca Ceoil . . . . .	9

<b>5</b>	<b>Použité knihovny</b>	<b>10</b>
5.1	GLAD . . . . .	10
5.2	GLFW . . . . .	10
5.3	stb_image . . . . .	10
<b>6</b>	<b>Herní scény</b>	<b>11</b>
6.1	Scéna hlavního menu . . . . .	11
6.2	Scéna pozastavené hra . . . . .	11
6.3	Scéna hry . . . . .	11
<b>7</b>	<b>Herní mechaniky</b>	<b>12</b>
7.1	Generování objektů . . . . .	12
7.2	Pohyb hráče . . . . .	12
7.3	Detekce kolize . . . . .	12
	<b>Bibliografie</b>	<b>15</b>
	<b>Zkratky</b>	<b>16</b>
	<b>Přílohy</b>	<b>19</b>
<b>A</b>	<b>Fotky z pokusů</b>	<b>20</b>
<b>B</b>	<b>Příloha další</b>	<b>21</b>

# 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 kompilovaný 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 kompilace, 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 řídicích 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 strídmou standardní knihovnu, která obsahuje základní matematické operace a funkce pro práci s pamětí a soubory, takže jakékoliv 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. [8, 6, 3, 9, 10]

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 prvotním 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 teď se implementuje do kompilátorů. V současnosti mezi nejpoužívanější kompilátory patří GCC a Clang. Jelikož bylo C velice populární, ovlivnilo řadu jiných programovacích jazyků, jako C, C#, Java, Rust, Go atd. [2, 4, 8, 6]

POZNAMKA přidej ještě k historii historii compileru, ať tam nejsou jen současně.

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. [2, 8, 14]

Jednoduchý program, který načte ze vstupu počet čísel, která chce uživatel seřadit. 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, kompilovaný, statický, má datové typy známy v době kompilace a nemá automatický správce paměti. V C++ se také programuje strukturovaně a procedurálně, ale na rozdíl od C také umožňuje programovat objektově. Objektové programování umožňuje používat objekty, které jsou vytvořeny pomocí tříd. Tyto třídy a jejich hierarchie dělí kód přehledněji a usnadňuje budoucí rozšiřování a debugování. 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. Proto se doporučuje programovat v C tak, aby daný C kód byl podmnožinou C++, což umožňuje jeho použití v jiném C++ programu. [13, 12]

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. Stroustrupovi přišlo objektově-orientované velmi užitečné, ale `Simula 67` byl příliš pomalý pro větší projekty. Proto se rozhodl že vytvoří nadmnožina jazyka C, která by umožňovala objektově-orientované programování a zároveň si zachoval rychlost C, s názvem `C with Classes`.

## 2 Grafická karta

### 2.1 Historie grafických karet

### 2.2 Architektura grafických karet

### 2.3 Grafická pipeline

### 2.4 Transformace a lineární algebra

## **3    OpengGL**

### **3.1   Historie OpenGL**

### **3.2   OpenGL pipeline**

### **3.3   Shadery**

### **3.4   Textury**

## Část II

# Vývoj hry v OpenGL

## 4 Grafika a zvuk

### 4.1 Aseprite

### 4.2 Bosca Ceoil

## 5 Použité knihovny

### 5.1 GLAD

### 5.2 GLFW

### 5.3 stb\_image



## **6 Herní scény**

### **6.1 Scéna hlavního menu**

### **6.2 Scéna pozastavené hra**

### **6.3 Scéna hry**

## 7 Serní mechaniky

### 7.1 Generování objektů

### 7.2 Pohyb hráče

### 7.3 Detekce kolize

## Závěr

# Bibliografie

1. AKENINE-MÖLLER, Tomas et al. *Real-Time Rendering*. 4th. A K Peters/CRC Press, 2018.
2. BANAHAN, Mike; BRADY, Declan; DORAN, Mark. *The C Book: Featuring the ANSI C standart*. 2nd. Addison-Wesley, 1991.
3. *Features of C* [Online]. 2025. Dostupné také z: <https://www.geeksforgeeks.org/c/features-of-c-programming-language/>. [citováno 2025-10-12].
4. HARBISON, Samuel P.; STEELE Guy L., Jr. *C: A Reference Manual*. 5th. Pearson, 2002.
5. HUGHES, John F. et al. *Computer Graphics: Principles and Practice*. 3rd. Addison-Wesley Professional, 2013.
6. KERNIGHAN, Brian W.; RITCHIE, Dennis M. *The C Programming Language*. 2nd. Pearson, 1988.
7. KIRK, David B.; HWU, Wen-mei W. *Programming Massively Parallel Processors: A Hands-on Approach*. 3rd. Morgan Kaufmann, 2016.
8. KOCHAN, Stephen G. *Programming in C*. 4th. Oxford University Press, 2012.
9. MORTENSEN, Peter. *What's the difference between a low-level, midlevel and high-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].
10. 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].
11. STROUSTRUP, Bjarne. *Design and Evolution of C++*. 1st. Addison-Wesley Professional, 1994.

12. STROUSTRUP, Bjarne. *Programming: Principles and Practice Using C++*. 2nd. Addison-Wesley Professional, 2014.
13. STROUSTRUP, Bjarne. *The C++ Programming Language*. 4th. Addison-Wesley Professional, 2013.
14. *Top Applications of C programming* [Online]. 2024. Dostupné také z: <https://www.wscubetech.com/resources/c-programming/applications>. [citováno 2025-10-09].

# Zkratky

**atd.** a tak dále. 3, 10

**tj.** to jest. 10

## Seznam obrázků

## Seznam tabulek



# Přílohy

## A Fotky z pokusů

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant 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 eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

## B Příloha další