Liliia Imamutdinova

+7 932 852 72 56 | lilyreber404@gmail.com | github.com/lilyreber

Education

National Research University Higher School of Economics St. Petersburg, Russia

BS in Applied Mathematics and Computer Science *2021-2025*

***Relevant courses**: Algorithms and Data Structures, C++/Python programming language, Calculus, Functional Languages, Linear Algebra and Geometry, Unix: Bash scripting, Discrete Mathematics, Mathematical Logic, Formal Languages.*

Projects

ChitChat | *C++17, Qt* March – June 2022

ChitChat is an application, that allows people to visit virtual rooms, talk with friends via voice chat and play videogames, such as Arkanoid, Hangman, etc.

- Fully implemented the game model on the client
- Made UI of registration and authentication, the playing field, the drawing of game objects relative to the player, character selection.
- Added modification of game objects on the server, including the generation of the playing field and collision of objects and their renewal.

Vector (C++ laboratory work) | *C++17, CMake* April 2022

Simplified std::vector implementation with allocator.

- Implemented STL-like vector API, which provides the strongest exception guarantee.
- Used placement new for creating and initializing elements in a vector.
- Implemented the same methods as the STL vector, including constructors, resize, reserve, and others.

BMP cropper (C++ laboratory work) | *C++17, CMake* February 2022

Utility to work with BMP formatted images

- Implemented functions to crop and rotate a BMP format image using C++ bytes casting.
- Used pragmas to pack classes that store BMP headers with specific alignment and padding.
- Added error handling and fault tolerance to prevent execution with invalid parameters.

Tic Tac Toe (C++ laboratory work) | *C++17, Boost.DLL, CMake* December 2021

Console application-game Tic Tac Toe for two players.

- Realised the entire logic of the game tic-tac-toe: storing the field, making moves, determining the current player, the winner, and the correctness of the move.
- Implemented two modes of displaying the field and entering commands, each mode corresponding to a separate shared library and switching between them.

Finite State Machine Specification Language (Formal Languages mini-project) | *Python3* October 2022

Creation of a language for describing Finite State Machine.

- Created concrete syntax.
- Implemented lexer and parser for the finite state machine description language using the ply library.

Skills

Programming languages: C++, Haskell, Python3, Bash
Development tools: Git, Docker, Qt Creator
Languages: Russian, English