**Министерство образования и науки Российской Федерации**

**Федеральное государственное автономное образовательное**

**учреждение высшего образования**

**«Национальный исследовательский технологический университет**

**«МИСИС»**

**Институт информационных технологий и компьютерных наук**

**Кафедра инженерной кибернетики**

**Прикладная математика**

**КУРСОВАЯ РАБОТА**

**по дисциплине «Объектно-ориентированное программирование»**

**Консольное приложение «Игры»**

Выполнили:

Студентки 1 курса группы БПМ-24-1

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ К.Е. Тимофеева

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ М.Г. Тестова

Преподаватель:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ С.В. Еременко

**Москва 2025**

## Содержание

1. Введение……………………………………………………………….…...
2. Цель работы…………………………………………………………….….
3. Задачи……………………………………………………………….……...
4. Общая структура проекта…………………………………………………
5. Описание работы……..……………………………………………………
   1. Главное меню……………….……………………….……………..
   2. Игра «Угадай число»………………..……………………………..
   3. Игра «Змейка»…………….………………………………………..
   4. Обработка некорректного ввода…………………………………...
6. Перспектива доработки проекта………………………………………....
7. Заключение………………………………………………………………...
8. Список использованных источников ……………………………………

**Введение**

В рамках данной курсовой работы применяются накопленные знания из дисциплины «Объектно-ориентированное программирование» при разработке консольного приложения «Игры». Программа предназначена для развлекательных целей.

Консольное приложение "Игры" представляет собой программу, написанную на языке C++, которая включает в себя две классические игры:

* "Угадай число" – игра, в которой пользователь должен угадать загаданное компьютером число.
* "Змейка" – классическая аркадная игра, где игрок управляет змейкой, собирая еду и избегая столкновений с границами поля и собственным телом.

Приложение разработано с использованием стандартной библиотеки C++ и работает в консольном режиме.

**Цель работы**

Создать консольное приложение, демонстрирующее реализацию двух игр ("Угадай число" и "Змейка") с использованием возможностей языка C++ и объектно-ориентированного программирования.

**Задачи**

1. Изучить материалы, необходимые для написания программы
2. Разработать структуру приложения, продумать логику работы
3. Реализовать игру "Угадай число"
4. Реализовать игру "Змейка" с управлением через клавиши
5. Обеспечить обработку пользовательского ввода и валидацию данных
6. Организовать систему команд, понятных программе

**Общая структура проекта**

project/ (корневая директория проекта)

│

├── game1/ (директория первой игры - "Угадай число")

│ ├── CMakeLists.txt (файл сборки для первой игры)

│ ├── game1.cpp (реализация логики игры)

│ ├── game1.hpp (заголовочный файл)

│ └── game1\_rules.txt (правила игры)

│

├── game2/ (директория второй игры - "Змейка")

│ ├── CMakeLists.txt (файл сборки для второй игры)

│ ├── game2.cpp (реализация логики игры)

│ └── game2.hpp (заголовочный файл)

│ └── game2\_rules.txt (правила игры)

│

├── CMakeLists.txt (главный файл сборки проекта)

├── main.cpp (основной исполняемый файл)

├── help\_message.txt (файл со справочной информацией)

├── start\_message228.txt (стартовое сообщение)

├── .gitignore (файл исключений Git)

└── README.md (документация проекта)

Файл main.cpp является основным исполняемым компонентом. Он содержит точку входа программы и обрабатывает основной пользовательский ввод.

Папка game1 содержит реализацию первой игры («Угадай число»). Заголовочный файл game1.hpp содержит объявление класса и функций для первой игры, game1.cpp – реализацию методов класса и игровой логики в целом. В отдельный файл вынесены правила игры.

Аналогично организована реализация игры «Змейка». Заголовочный файл game2.hpp содержит объявление класса и функций для первой игры, game2.cpp содержит реализацию методов класса и игрового процееса. В файле game2\_rules.txt содержатся правила игры.

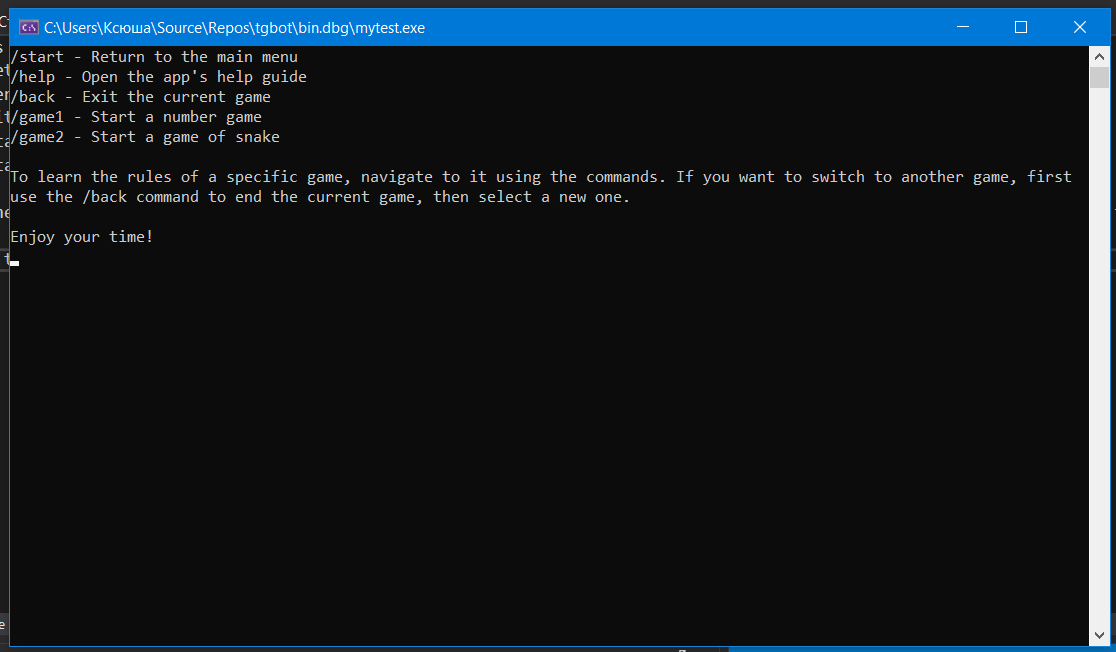
Также в папке project лежат вспомогательные файлы start\_message.txt и help\_message.txt, которые содержат стартовое сообщение и справочную информацию по приложению соответственно. Текст вынесен в отдельные файлы для удобства редактирования.

В проекте используется система сборки CMake, с помощью которой все файлы объединяются в общую систему с четкими зависимостями и простым процессом сборки на любой поддерживаемой платформе. Преимущество использования CMake в том, что каждая игра собирается в отдельную библиотеку, возможна независимая компиляция объектов и мы можем с легкостью добавлять новые игры.

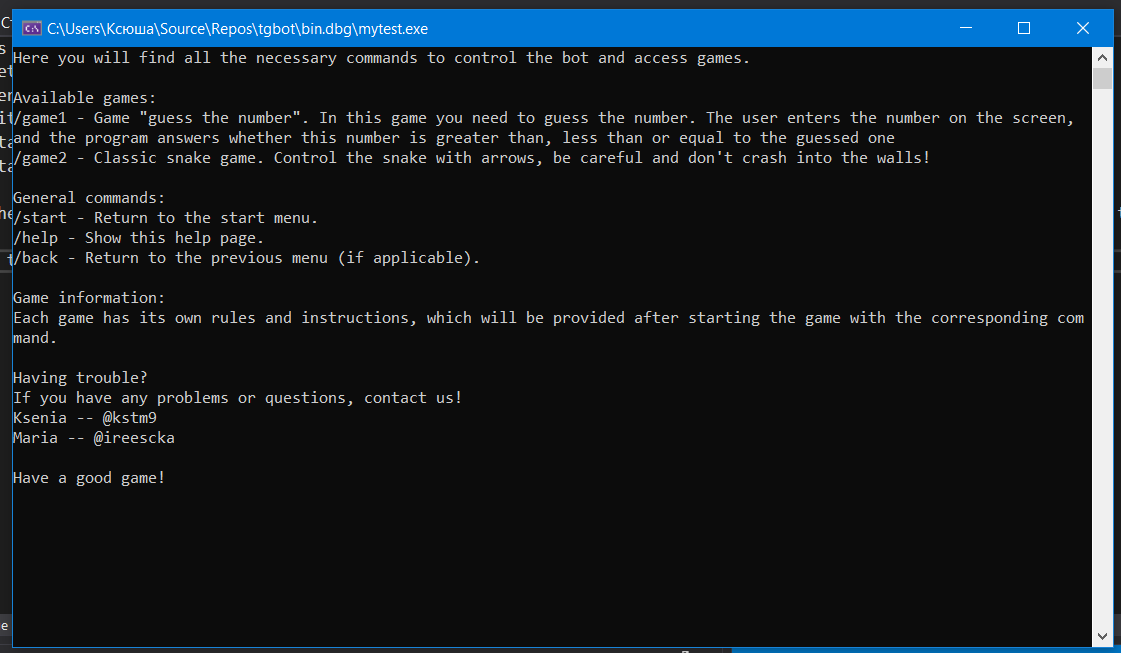
**Описание работы**

**Главное меню**

При открытии приложения пользователь видит стартовое сообщение, в котором рассказывается, какие существуют команды для управления.



При переходе по команде /help открывается сообщение со справочной информацией, где кратко рассказывается про каждую игру, а также указаны контакты разработчиков приложения. Консоль каждый раз очищается.

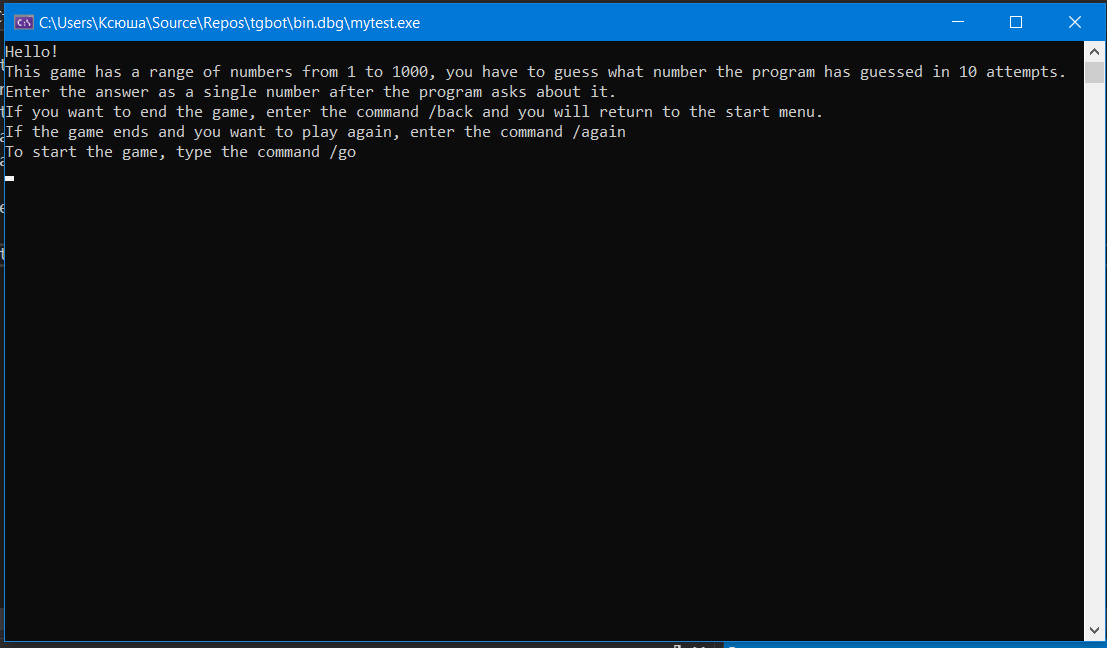


Команда /back возвращает на предыдущую страницу, то есть если пользователь находится в первой игре, то по этой команде он вернется на страницу с правилами, оттуда по команде /back он вернется на стартовое меню.

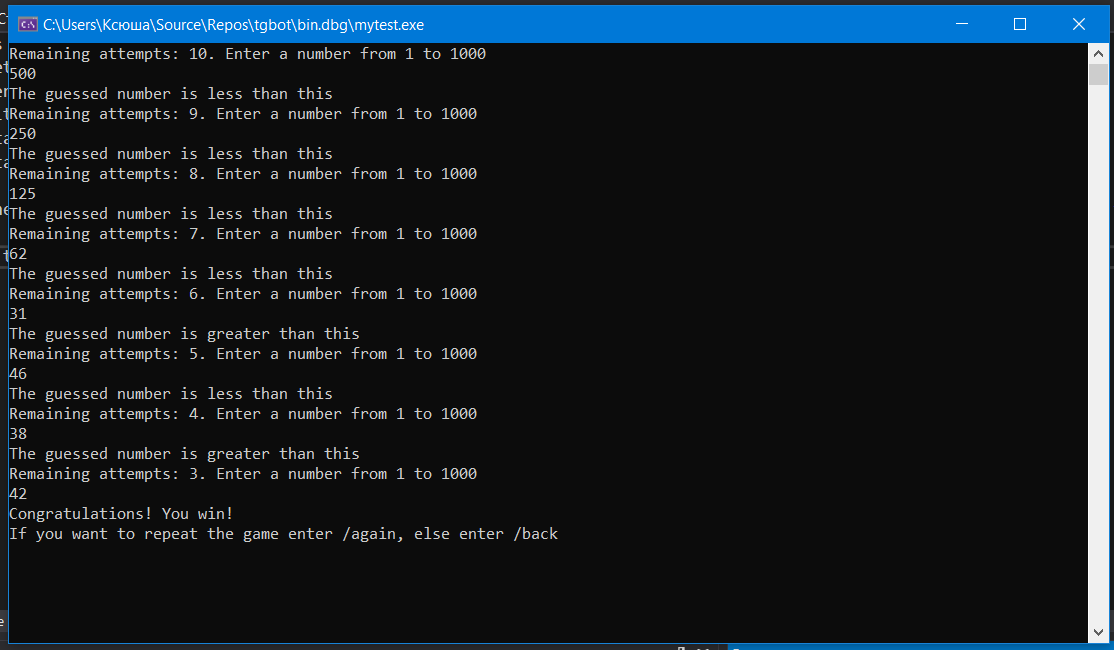
**Игра «Угадай число»**

В этой игре пользователю нужно определить загаданное программой число в диапазоне от 1 до 1000. На это дается 10 попыток. Предполагается, что смышленый игрок будет действовать по алгоритму бинарного поиска.

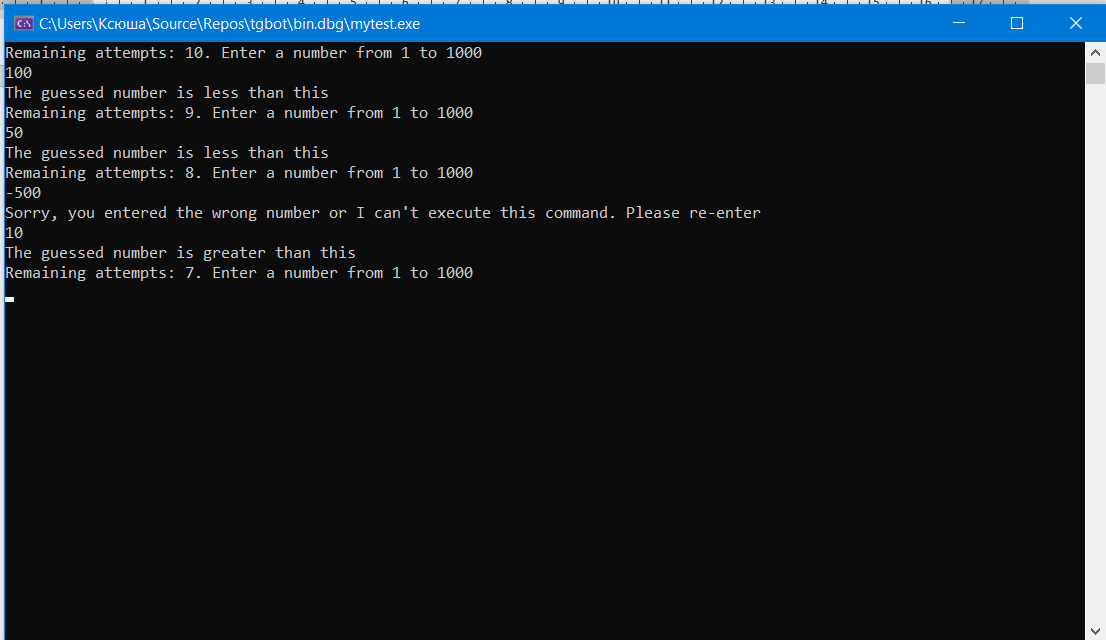
По команде /game1 пользователь видит в консоли правила и доступные во время игры команды.



Программа загадывает псевдорандомное число с помощью rand(), после чего прокручивает цикл на 10 повторений. Каждый раз обрабатываются всевозможные варианты ввода пользователя (/back, /again, /start, /go, число, некорректный ввод). С помощью команды /again пользователь может начать игру заново: будет загадано новое число и количество попыток вернется к начальным 10.



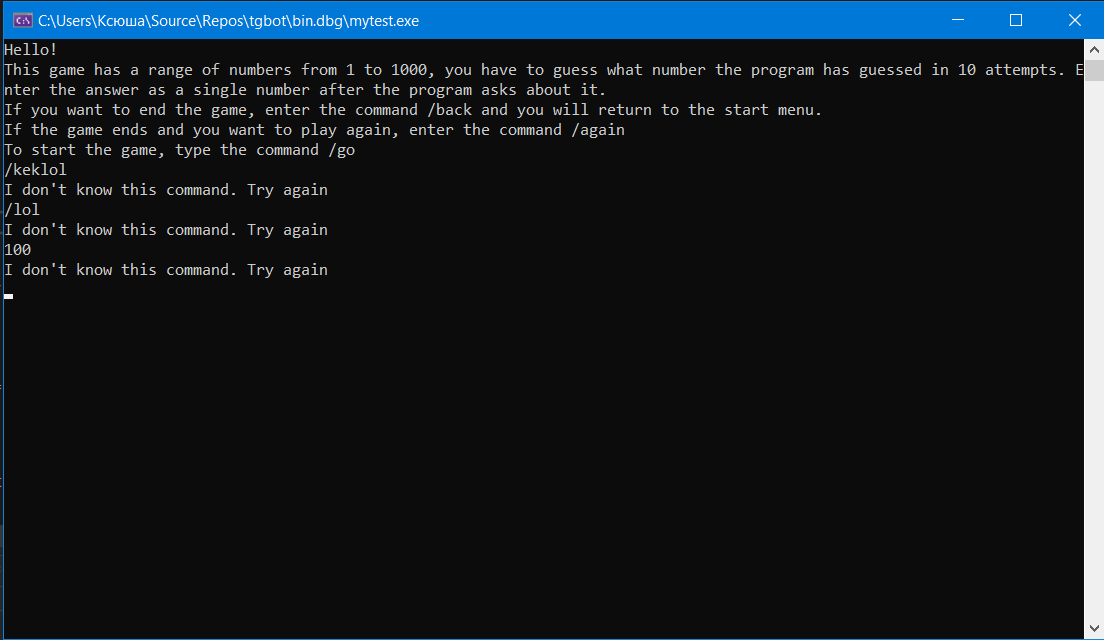
В игре предусмотрен ввод натуральных чисел и перечисленных в правилах команд.



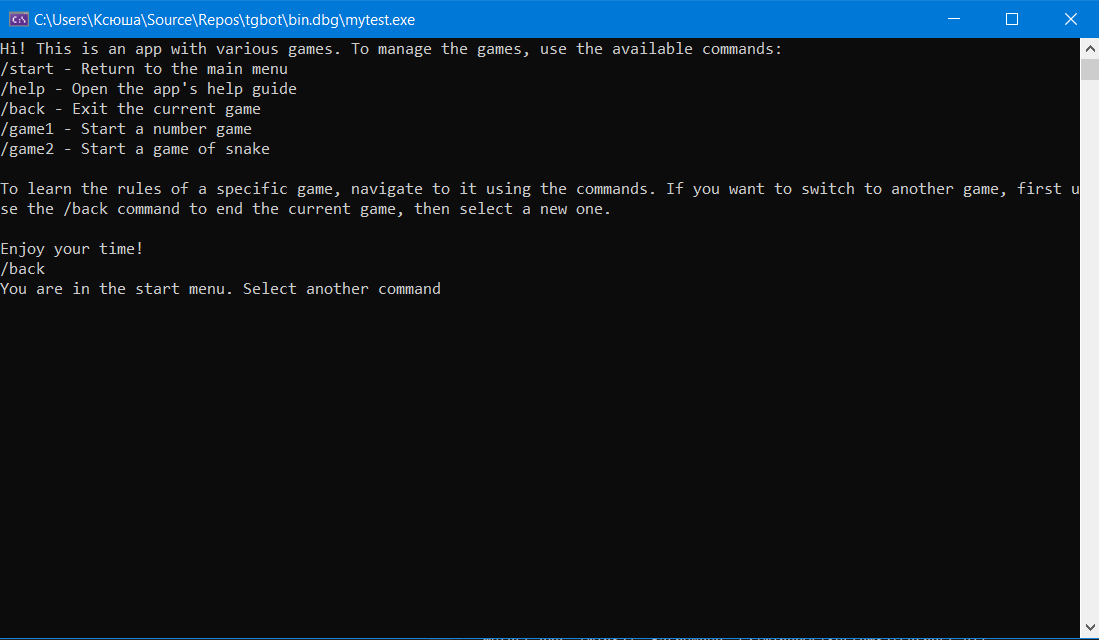
**Игра «Змейка»**

**Обработка некорректного ввода**

В каждом окне, где требуется ввод существует проверка на введенные данные. Если строка на входе не соответствует ни одной из принимаемых команд, то программа вывод сообщение о том, чтобы пользователь повторил ввод.



Если пользователь введет команду /back, находясь в стартовом меню, программа скажет, что назад вернуться нельзя, и попросит повторить ввод.



**Перспектива доработки проекта**

1. Расширение функциональности игр

Для «Угадай число» можно добавить выбор уровней, на каждом уровне будет различный диапазон значений. Также можно добавить уровень «с промахом», где программа может не более одного раза соврать пользователю об отношении загаданного и полученного чисел. Также возможно добавление режима, где пользователь будет загадывать число, а компьютер наоборот угадывать.

Для «Змейки» возможно добавление еды с различными эффектами: временное ускорение, увеличение длины на 2, уменьшение длина на 1 и т.д. Возможно добавление выбора размера поля и сложности карты (препятствия на самом поле, лабиринт). Также интересным обновлением может послужить сохранение рекордов пользователя локально на устройство.

1. Добавление новых игр

Так как каждая игра представляет собой отдельную библиотеку, добавлять новые игры довольно легко. Можно расширить ассортимент игр, добавив, например, игру «Города», где пользователь и программа по очереди называют города на последнюю букву.

1. Улучшение архитектуры проекта

Для еще более простого добавления новых игр можно реализовать паттерн «стратегия» и создать общий интерфейс для всех игр.

1. Улучшение пользовательского интерфейса

Для более удобного пользования программой, можно внедрить такие идеи:

* Графическое меню с навигацией стрелками
* Замена команд на кнопки
* Система настроек (цвета, управление, сложность)

**Заключение**

Приложение успешно демонстрирует работу с базовыми конструкциями C++ и файловым вводом-выводом, обработку ввода, управление состоянием программы и простую графику в консоли. В ходе создания приложения мы разработали две игры. В «Угадай число» обеспечена обработка пользовательского ввода, генерация случайных чисел и система подсказок. В "Змейке" реализовано управление, столкновения, рост змейки и подсчёт очков. Проект имеет чёткую структуру, разделённую на модули, что упрощает его дальнейшее расширение. Мы поработали с системой сборки CMake, благодаря которой приложение может быть собрано на любой доступной платформе. Пользовательский интерфейс интуитивно понятен благодаря текстовому меню и подсказкам. Приложение может быть расширено добавлением новых игр или улучшением существующих.

**Список использованных источников**

1. Лекции – С++ и основы ООП (2012) // URL: <https://youtube.com/playlist?list=PLE9F6A65165CBC023&si=wJHsc0lq_MoD9Ra2>
2. Пишем змейку на С++ // URL: <https://code-live.ru/post/cpp-oldschool-snake/>