Оглавление

[**1.** **Введение.** 1](#_Toc152165346)

[**1.1. Цели.** 1](#_Toc152165347)

[**1.2. Задачи.** 1](#_Toc152165348)

[**1.3. План разработки приложения.** 1](#_Toc152165349)

[**2.** **Инструкция к игре.** 2](#_Toc152165350)

[**2.1 Меню.** 2](#_Toc152165351)

[**2.2 Игровой процесс.** 2](#_Toc152165352)

[**2.3 Алгоритм угадывания числа.** 3](#_Toc152165353)

[**3.** **Техническая документация к игре.** 4](#_Toc152165354)

[**3.1 Псевдокод.** 4](#_Toc152165355)

[**3.2. Блок-схема.** 7](#_Toc152165356)

[**3.2.1 Функция main.** 7](#_Toc152165357)

[**Заключение:** 8](#_Toc152165358)

# **Введение.**

## **1.1. Цели.**

Программа для угадывания числа, которое загадал сам компьютер, основана на использовании нескольких операторов switch и функции rand(). Использован исключительно язык С/C++ , а также специальный алгоритм, чтобы узнать загаданное число, используя подсказки от компьютера.

## **1.2. Задачи.**

Были поставлены следующие задачи:

- Разработать интерфейс, который даёт возможность победы в игре;

- Разработать меню;

- При разработке использовать известные для нас команды ЯП С/С++;

- При необходимости, дополнительно изучить команды, которые помогут реализовать приложение.

## **1.3. План разработки приложения.**

Создание мини-игры «Угадай число» было разбито на два этапа:

1) Разработать интерфейс;

2) Разработать игровой процесс.

# **Инструкция к игре.**

## **2.1 Меню.**

В игровом меню пользователю предоставляется 3 варианта выбора:

- «Ввести число (количество попыток)». Это ввод числа, которое игроку предлагается угадать.

- «Подсказки». В этом разделе меню игрок имеет возможность воспользоваться подсказкой из ряда вариантов.

- «Сдаться». При нажатии на этот пункт меню, пользователь выйдет из консольного окна и вернётся в главное меню.

Переключение между пунктами меню происходит при помощи стрелок (cursor control keys) вверх (up) и вниз (down). Для выбора того или иного варианта меню используется клавиша enter. Также для досрочного выхода из консольного окна можно использовать клавишу Esc.

## **2.2 Игровой процесс.**

Первый case отвечает за то, чтобы пользователь мог ввести число, которое, на его взгляд, является тем, которое загадал компьютер.

Пользователь должен ввести число, если оно равно тому, что загадал компьютер, то программа выведет поздравление, если число не сходиться с тем, что было загадано, то программа выведет сообщение об ошибке. У пользователя есть только 3 попытки ввести число.

Во втором case уже хранится второй пункт меню, а точнее Подсказки. Когда мы в основном меню выбираем пункт 2, откроется второе меню, где будут все доступные для пользователя подсказки:

* «Это число делиться на». Позволяет узнать, на какое число делиться искомое;
* «Это число больше числа». Позволяет уточнить нижний диапазон;
* «Это число меньше числа». Позволяет уточнить верхний диапазон;
* «Узнать диапазон». Позволяет узнать примерный диапазон, в котором находится искомое число;
* «Узнать число второго разряда, если оно есть». Позволяет узнать число, которое находится во втором разряде искомого.

## **2.3 Алгоритм угадывания числа.**

Для угадывания числа сначала мы должны узнать, четное ли оно или нет. Далее узнаем диапазон. Потом узнаем цифру разряда десятых. Когда мы ее узнаем, мы узнаем три первых числа, считая слева направо, задуманной цифры. Далее в зависимости от четности, точнее, если число четное мы будем узнавать делиться ли число на 4, 8, 16 и так далее, а если оно нечетное, будем узнавать делится ли оно на 3, 7, 5 и так далее. Чем больше мы узнаем, тем меньше чисел у нас для проверки остается.

На примере мы сначала узнали, что число нечетное. Далее узнали, что но от 1771 и до 1861. Потом узнали цифру разряда десятых — это 2. То есть мы уже знаем начало этого числа — 182х. Нам осталось узнать только цифру разряда единиц. Зная, что число нечетное, мы записываем числа, которые могут подойти нам, то есть, 1821, 1823, 1825, 1827, 1829. Далее мы проверяем, делится ли число на 3. Нам сказали, что нет, тогда вычеркиваем неподходящие числа: ~~1821~~, 1823, 1825, ~~1827~~, 1829. Далее проверяем, делится ли оно на 5, нам также сказали, что нет. Остаются лишь 1823 и 1829. Далее, узнаем, больше ли задуманное число, чем число 1825. И мы получаем ответ, что 1825 меньше задуманного числа. То есть число, которое загадал компьютер — это 1829.

# **Техническая документация к игре.**

## **3.1 Псевдокод.**

Начало

В number записывается любое число от 1 до 2000

country и countHelp хранят в себе количество попыток вводить число и количество подсказок

Запуск Цикла

Выбор пункта

Если 1

Counttry уменьшается на 1

Ввод числа

Ответ программы, то число, или нет

Если да

Active = true

Выход из switch()

Если нет

Выход из switch

Конец Если 1

Если 2

Если counthelp = 0

Выход из switch()

Уменьшение countHelp на 1

Выбор пункта меню

Если 1

Ввод любого числа

Проверка, делится ли загаданное число на введенное

Если да

Сказать пользователю

Если нет

Сказать пользователю, что не делится

Выход из switch

Если 2

Ввод любого числа

Проверка, больше ли оно заданного числа

Если да

Сказать пользователю

Если нет

Сказать пользователю, что не больше

Выход из switch()

Если 3

Ввод любого числа

Проверка, меньше ли оно заданного числа

Если да

Сказать пользователю

Если нет

Сказать пользователю, что не меньше

Выход из switch()

Если 4

Вывод диапазона. То есть чисел, между которыми находится загаданное число.

Выход из switch();

Если 5

Вывод задуманного числа деленного на 10 и после этого деленного по модулю на 10.

Выход из switch()

Выход из switch()

Конец Если 2

Если 3

Active = true

Выход из switch()

Конец Если 3

Если country = 0

Active = true;

Конец если

Если active = true

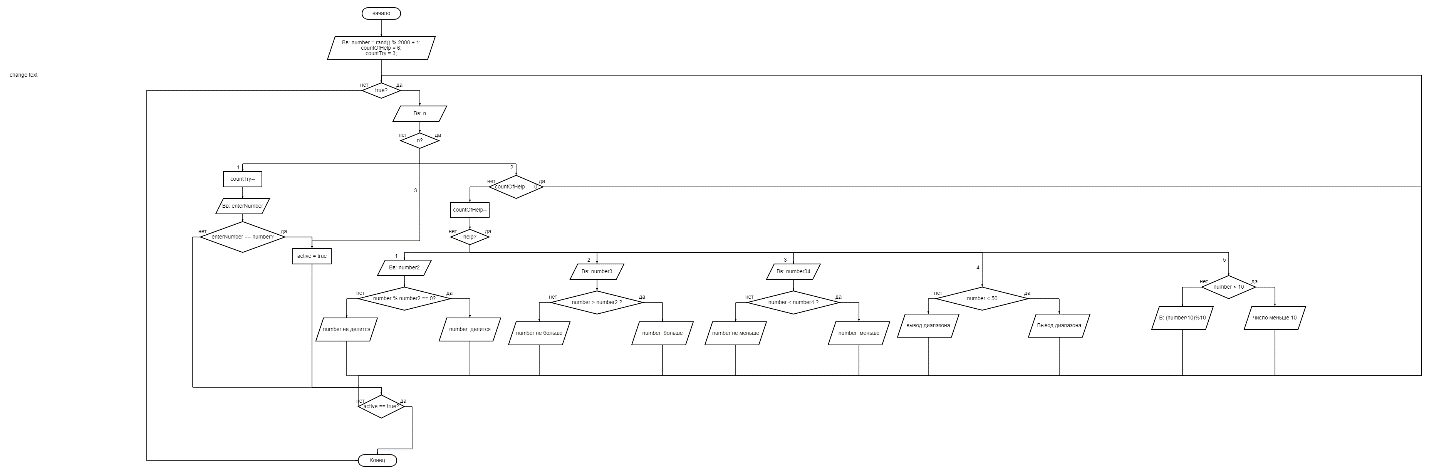
Выход из цикла

Конец если

Конец программы

## **3.2. Блок-схема.**

### **3.2.1 Функция main.**



# **Заключение:**

В ходе выполнения данной работы, я познакомился и освоил некоторые команды ЯП С/С++. Научился создавать меню, изменять цвет в консоли, создавать динамику в 2D пространстве. Пользоваться функциями очистки консольного окна, функцией отслеживания нажатия клавиш на клавиатуре, функцией для задания координат и др. При помощи ASCII-символов нарисовали графические объекты.

При выполнении программы сделали следующие **выводы**:

1. Для постоянного вызова функций удобнее использовать цикл while;
2. Чтобы код легко читался, необходимо следовать рекомендациям по хорошему стилю программирования;
3. Модульное программирование, создание дополнительных функций -повышает эффективность нашего кода и программы;
4. При планировании задачи, лучше всего заранее строить алгоритм, после чего, приступать непосредственно к работе.