Задание 3

Игра «угадай число» с таблицей рекордов

Закрепим полученные на текущий момент знания и навыки, а также немного развлечёмся.

Игра «угадай число» - компьютер загадывает число, пользователь пытается угадать. При этом подсчитывается количество попыток и формируется таблица рекордов.

В материалах к этой самостоятельной работе можно найти несколько заготовок кода, объединив которые (с небольшими доработками) можно получить базовую реализацию приложения.

Суть игры. Приложение «загадывает» некоторое число и ждёт в цикле ввода от пользователя. В случае, если пользователь ввёл число, меньшее загаданного, выводится подсказка «less than». В случае, если пользователь ввёл число, большее заданного, выводится подсказка «greater than». В случае, когда пользователь, наконец, угадывает число, выводится сообщение «you win!», и цикл ввода завершается. В качестве примера см пример `check_value.cpp` из материалов к самостоятельной работе.

Для того, чтобы игра была более интересной, в приложении стоит использовать генератор случайных чисел вместо «угадывания» константы. При этом рекомендуется использовать самый простой генератор случайных чисел из стандартной библиотеки – std::rand. Есть ряд нюансов:

- по умолчанию std::rand выдаёт значения от 0 до константы `RAND_MAX` (конкретное значение специфично для разных платформ, однако стандартном гарантируется, что это значение не менее `32767`)
- угадывать столь большое число может быть трудновато (хотя через бинарный поиск – даже интересно), поэтому мы берём в качестве загаданного числа не результат выполнения std::rand, а остаток от деления результата на некоторую константу, которой и ограничиваем максимальное для загадывания значение (в примере ниже ограничением станет число 100):

```
const int max_value = 100;
const int random_value = std::rand() % 100;
```

- перед использованием генератор случайных чисел нужно инициализировать начальным значением (так называемый seed), иначе каждый раз при перезапуске приложения генератор случайных чисел будет выдавать одну и ту же последовательность значений. В качестве начального значения для инициализации хорошим вариантом является текущее время с точностью до секунды. Делается это следующим образом:

```
std::srand(std::time(nullptr));
```

В качестве примера работы с генератором случайных чисел см пример `random_value.cpp`.

Для реализации таблицы рекордов пользователю при старте приложения выводится предложение представиться. Введенное значение запоминается в переменную строкового типа. Далее приложение считает количество попыток, которое потребовалось пользователю для угадывания числа. Это число так же запоминается. Затем имя пользователя и число попыток дописываются в файл с таблицей рекордов и выводится полное текущее содержимое таблицы рекордов. См пример `high_scores.cpp`.

Пример работы приложения:

```
> guess_the_number
Hi! Enter your name, please:
Username
Enter your guess:
42
less than 42
1
greater than 1
33
you win! attempts = 3

High scores table:
SomeName 13
SomeOtherName 24
Username 9
Username 3
```

Дополнительное задание 1. Задание опциональное и для формальной сдачи работы не обязательное.

Добавить обработку параметра командной строки “-max” с одним числовым аргументом, с помощью которого можно задать максимальное значения для загаданного числа. См пример `argument.cpp`. В случае, если параметр не передаётся, ограничением по умолчанию выбирается некоторая константа на усмотрение автора (например, 100). Пример запуска приложения с этим аргументом:

> guess_the_number –max 42

Дополнительное задание 2. Задание опциональное и для формальной сдачи работы не обязательное.

Добавить обработку параметра командной строки “-table” без аргументов, с помощью которого можно вывести таблицу рекордов сразу, без необходимости игры. При этом приложение выводит текущие значения из таблицы рекордов и сразу завершается. Пример запуска с этим аргументом:

> guess_the_number -table

Дополнительное задание 3. Задание опциональное и для формальной сдачи работы не обязательное.

При чтении таблицы рекордов для каждого пользователя определять минимальное значение числа попыток и выводить только их.

Например, содержимое таблицы рекордов:

```
A 15
B 12
```

```
A 10
B 1
A 3
```

Для пользователей A и B имеем 3 и 2 записи соответственно. Но в конце работы приложения выводим только минимальные значения (это же все-таки рекорды):

```
High scores table:
A 3
B 1
```

Дополнительное задание 4. (со звёздочкой) Задание опциональное и для формальной сдачи работы не обязательное.

При формировании таблицы рекордов перезаписывать предыдущий результат пользователя, если текущий результат стал лучше. Пример:

- текущие значения таблицы рекордов:

```
B 13
A 12
C 5
```

- далее предположим, что пользователь A сыграл ещё раз и выполнил задание за 5 попыток, тогда содержимое таблицы рекордов должно измениться следующим образом:

```
B 13
C 5
A 5
```

Дополнительное задание 5. (на случай, если станет скучно) Вариация дополнительного задания 1 – добавить обработку параметра командной строки ”-level” с одним аргументом численного типа, через который можно задать уровень сложности. От уровня сложности зависит максимальное значение загаданного числа. Например, добавим 3 уровня сложности:

1 – масимальное значение 10 (таким образом отгадывать будет в диапазоне от 0 до 9)

2 – максимальное значение 50 (диапазон станет от 0 до 49)

3 – максимальное значение 100 (диапазон от 0 до 99)

Пример запуска приложения с этим аргументом:

> guess_the_number – level 3

В случае выполнения дополнительного задания 1, считаем ввод параметров “-level” и “-max” одновременно ошибочной ситуацией. Допустимо либо указать уровень сложности, либо явно задать максимальное значение.

В самостоятельной работе требуется:

1. создать приложение «угадай число»
2. для сборки использовать CMake
3. выгрузить результат на github.com в свой аккаунт

В Чат с преподавателем в Личном кабинете отправить:

1. ссылку на репозиторий с исходниками приложения