Программа функционирует при помощи функций, которые производят определенные операции, которые составляют логику игры, ее функционирование.

Функция generate\_start\_number:

Генерирует случайное число, проверяет его на то, чтобы оно было четырехзначным, цифры в этом числе должны быть различными

int correct\_number = rand() % 10000;

bool b = false;

while (! b)

{

if ((correct\_number / 1000) != ((correct\_number / 100) % 10) &&

(correct\_number / 1000) != ((correct\_number % 100) / 10) &&

(correct\_number / 1000) != (correct\_number % 10) &&

((correct\_number / 100) % 10) != ((correct\_number % 100) / 10) &&

((correct\_number / 100) % 10) != (correct\_number % 10) &&

((correct\_number % 100) / 10) != (correct\_number % 10) &&

correct\_number >= 1000)

{

b = true;

}

else

{

correct\_number = rand() % 10000;

}

}

return correct\_number;

Переменная correct\_number – это и есть то число, которое сгенерировала программа (число, которое пользователь будет угадывать).

Функция cows

Вычисляет число коров (цифр, совпадающих по значению, но отличным по местоположению) в введенным пользователем числе и числе, сгенерированным программой.

int cows\_counter;

cows\_counter = 0;

if ((correct\_number / 1000) == (my\_number / 1000) ||

(correct\_number / 1000) == ((my\_number / 100) % 10) ||

(correct\_number / 1000) == ((my\_number % 100) / 10) ||

(correct\_number / 1000) == (my\_number % 10))

{

cows\_counter++;

}

if (((correct\_number / 100) % 10) == (my\_number / 1000) ||

((correct\_number / 100) % 10) == ((my\_number / 100) % 10) ||

((correct\_number / 100) % 10) == ((my\_number % 100) / 10) ||

((correct\_number / 100) % 10) == (my\_number % 10))

{

cows\_counter++;

}

if (((correct\_number % 100) / 10) == (my\_number / 1000) ||

((correct\_number % 100) / 10) == ((my\_number / 100) % 10) ||

((correct\_number % 100) / 10) == ((my\_number % 100) / 10) ||

((correct\_number % 100) / 10) == (my\_number % 10))

{

cows\_counter++;

}

if ((correct\_number % 10) == (my\_number / 1000) ||

(correct\_number % 10) == ((my\_number / 100) % 10) ||

(correct\_number % 10) == ((my\_number % 100) / 10) ||

(correct\_number % 10) == (my\_number % 10))

{

cows\_counter++;

}

return cows\_counter;

cows\_counter – переменная, содержащая в себе число коров

correct\_number – число, сгенерированное программой

my\_number – введенное пользователем число

функция bulls

Вычисляет число быков (цифр, совпадающих по значению и по местоположению) в введенным пользователем числе и числе, сгенерированным программой.

int bulls\_counter;

bulls\_counter = 0;

if ((correct\_number / 1000) == (my\_number / 1000))

{

bulls\_counter++;

}

if (((correct\_number / 100) % 10) == ((my\_number / 100) % 10))

{

bulls\_counter++;

}

if (((correct\_number % 100) / 10) == ((my\_number % 100) / 10))

{

bulls\_counter++;

}

if ((correct\_number % 10) == (my\_number % 10))

{

bulls\_counter++;

}

return bulls\_counter;

bulls\_counter – переменная, содержащая в себе число быков

correct\_number – число, сгенерированное программой

my\_number – введенное пользователем число

Функция Game\_new

Производит игру в нормальном режиме (пользователю заранее не известно число, он имеет неограниченное количество попыток его угадать)

int my\_number;

bool win, correct;

win = false;

correct = false;

system("CLS");

cout << "The specified number, good luck!" << endl;

do

{

do

{

cout << "Enter number: ";

cin >> my\_number;

if ((my\_number >= 1000) && (my\_number <= 9999))

correct = true;

else

cout << "Error!Incorrect number!" << endl;

} while (correct == false);

cout << "++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++" << endl;

cout << "Bulls:" << bulls(correct\_number, my\_number);

cout << " Cows:" << cows(correct\_number, my\_number) << endl;

if (my\_number == correct\_number)

win = true;

} while (win == false);

cout << "++++++++++++++++++++++++++++++Achievement+++++++++++++++++++++++++++++++++++++" << endl;

system("PAUSE");

correct\_number – число, сгенерированное программой

my\_number – введенное пользователем число

correct – переменная, определяющая, корректно ли введенное пользователем число

win – переменная, определяющая, угадал ли пользователь задуманное программой число

Функция New\_game\_easy

Производит игру в легком режиме (пользователю заранее известно число, он имеет неограниченное количество попыток его угадать)

int my\_number;

bool win, correct;

win = false;

correct = false;

system("CLS");

cout << "The specified number:" << correct\_number << endl;

do {

do {

cout << "Enter number:";

cin >> my\_number;

if ((my\_number >= 1000) && (my\_number <= 9999))

correct = true;

else

cout << "Error!Incorrect number!" << endl;

} while (correct == false);

cout << "++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++" << endl;

cout << "Bulls:" << bulls(correct\_number, my\_number);

cout << " Cows:" << cows(correct\_number, my\_number) << endl;

if (my\_number == correct\_number)

win = true;

} while (win == false);

cout << "+++++++++++++++++++++++++++++++Achievement++++++++++++++++++++++++++++++++++++++" << endl;

system("PAUSE");

correct\_number – число, сгенерированное программой

my\_number – введенное пользователем число

correct – переменная, определяющая, корректно ли введенное пользователем число

win – переменная, определяющая, угадал ли пользователь задуманное программой число

Функция New\_game\_hard

Производит игру в сложном режиме (пользователю заранее неизвестно число, он имеет ограниченное количество попыток его угадать)

int my\_number, attempt;

bool win, correct;

win = false;

correct = false;

attempt = 1;

system("CLS");

cout << "The number set you have 5 attempts, good luck!" << endl;

//printf("The specified number:%d\n",x);

do

{

cout << "Attempt:" << endl;

cout << attempt;

do

{

cout << "Enter the number:" << endl;

cin >> my\_number;

if ((my\_number >= 1000) && (my\_number <= 9999))

correct = true;

else

cout << "Error!Incorrect number!" << endl;

} while (correct == false);

cout << "++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++" << endl;

cout << "Bulls:" << bulls(correct\_number, my\_number);

cout << " Cows:" << cows(correct\_number, my\_number) << endl;

if (my\_number == correct\_number)

win = true;

if (attempt == 5)break;

attempt++;

} while (win == false);

if (my\_number == correct\_number)

cout << "+++++++++++++++++++++++++++++++Achievement+++++++++++++++++++++++++++++++" << endl;

if (attempt == 5)

cout << "+++++++++++++++++++++++++++++++YUO LOSE++++++++++++++++++++++++++++++++++++" << endl;

system("PAUSE");

correct\_number – число, сгенерированное программой

my\_number – введенное пользователем число

correct – переменная, определяющая, корректно ли введенное пользователем число

win – переменная, определяющая, угадал ли пользователь задуманное программой число

attempt – переменная, определяющая количество введенных чисел пользователем (попыток)

Функция runTest

Производит тестирование игры, т.е. рассматривает различные случаи игры.

Проверяет работу функций, вычисляющих количество быков и коров.

assert(cows(1234, 4321) == 4);

puts("Test 1 passed");

assert(cows(4568, 2931) == 0);

puts("Test 2 passed");

assert(cows(1234, 8921) == 2);

puts("Test 3 passed");

assert(cows(1987, 1997) == 3);

puts("Test 4 passed");

assert(cows(1234, 4987) == 1);

puts("Test 5 passed");

assert(bulls(1234, 4321) == 0);

puts("Test 6 passed");

assert(bulls(1234, 1658) == 1);

puts("Test 7 passed");

assert(bulls(4895, 4823) == 2);

puts("Test 8 passed");

assert(bulls(1234, 1235) == 3);

puts("Test 9 passed");

assert(bulls(1234, 1234) == 4);

puts("Test 10 passed");

getch();

Меню программы :

while (1)

{

system("CLS");

puts(" 1 - New Game (easy)");

puts(" 2 - New Game (normal)");

puts(" 3 - New Game (hard)");

puts(" 4 - Test Game");

puts(" 0 - Exit");

c = getch();

switch (c)

{

case '1':New\_game\_easy(j);break;

case '2':Game\_new(j);break;

case '3':New\_game\_hard(j);break;

case '4':runTest();break;

case '0':return 0;

default: puts("Error!");

}