Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Саратовский государственный технический университет имени Гагарина Ю.А.»

Институт прикладных информационных технологий и коммуникаций

Кафедра «Информационная безопасность автоматизированных систем»

Направление подготовки 10.03.01 Информационная безопасность

Расчетно-графическая работа по дисциплине «Языки программирования»

Игра Digger

Выполнил: студент первого курса

Учебной группы с-ИБС-12

Очной формы обучения

Соколовский Роман Владимирович

Проверил: ассистент каф. ИБС

Романчук Сергей Петрович

Саратов 2020

# Аннотация

На прямоугольной карте, представляющей собой «шахту» перемещается диггер. Он может как двигаться по уже существующим ходам, так и прокапывать новые. Цель игры – собрать все разбросанные по карте сокровища. Игроку активно мешают монстры, которые также перемещаются по карте. Столкновение с монстром означает «гибель» игрока. Монстры могут преследовать игрока.

# Содержание

[Аннотация 2](#_Toc41566810)

[Содержание 3](#_Toc41566811)

[Введение 4](#_Toc41566812)

[Теоретическая часть 5](#_Toc41566813)

[Практическая часть 7](#_Toc41566814)

[Заключение 9](#_Toc41566815)

[Приложения 10](#_Toc41566816)

[Литература 24](#_Toc41566817)

# Введение

Тема: Игра Digger

Цель: разработать аналог известной аркадной игры Digger.

Задачи:

Реализовать карту, содержащую диггера, монстров и сокровища.

Реализовать возможность управления диггером.

Реализовать поведение агрессивное монстров по отношению к диггеру.

Реализовать условия победы и поражения.

# Теоретическая часть

C++ — компилируемый, статически типизированный язык программирования общего назначения.

Поддерживает такие парадигмы программирования, как процедурное программирование, объектно-ориентированное программирование, обобщённое программирование. Язык имеет богатую стандартную библиотеку, которая включает в себя распространённые контейнеры и алгоритмы, ввод-вывод, регулярные выражения, поддержку многопоточности и другие возможности. C++ сочетает свойства как высокоуровневых, так и низкоуровневых языков. В сравнении с его предшественником — языком C, — наибольшее внимание уделено поддержке объектно-ориентированного и обобщённого программирования.

C++ широко используется для разработки программного обеспечения, являясь одним из самых популярных языков программирования. Область его применения включает создание операционных систем, разнообразных прикладных программ, драйверов устройств, приложений для встраиваемых систем, высокопроизводительных серверов, а также игр. Существует множество реализаций языка C++, как бесплатных, так и коммерческих и для различных платформ. Например, на платформе x86 это GCC, Visual C++, Intel C++ Compiler, Embarcadero (Borland) C++ Builder и другие. C++ оказал огромное влияние на другие языки программирования, в первую очередь на Java,C#, Golang, Solidity, PHP.

Синтаксис C++ унаследован от языка C. Одним из принципов разработки было сохранение совместимости с C. Тем не менее, C++ не является в строгом смысле надмножеством C; множество программ, которые могут одинаково успешно транслироваться как компиляторами C, так и компиляторами C++, довольно велико, но не включает все возможные программы на C.

Code::Blocks — свободная кроссплатформенная среда разработки. Code::Blocks написана на С++ и использует библиотеку wxWidgets. Имея открытую архитектуру, может масштабироваться за счёт подключаемых модулей. Поддерживает языки программирования С, С++, D (с ограничениями), Fortran.

Code::Blocks разрабатывается для Windows, Linux и Mac OS X. Среду можно собрать из исходников практически под любую Unix-подобную систему, например FreeBSD, PC-BSD.

# Практическая часть

Для людей, никогда не игравших в игру, введены инструкции, которые можно прочесть перед началом. Все действия в программе происходят внутри массива 15\*10, являющегося картой. На карте в виде цифр располагаются сам диггер, монстры, его преследующие, тоннели и изумруды, которые нужно собрать.

Условия игры простые: нужно либо одолеть всех монстров, либо собрать все находящиеся на карте изумруды. Как только одно из условий выполнено, игрок объявляется победителем.

Игровой процесс пошаговый: сначала ходит игрок, затем монстры. Сначала для игрока выстраивается карта, после чего игрок может в свой ход путем ввода соответствующего числа совершить одно из восьми действий: переместиться на одну клетку вправо, влево, вверх или вниз либо атаковать монстра, находящегося на соседней с ним клетке (диагональные клетки соседними не считаются). После этого все монстры на карте по одному двигаются в сторону игрока, ориентируясь на его координаты.

Если монстры окажутся на той же клетке, что и диггер, то игроку будет объявлено поражение. Для предотвращения такого исхода игрок может атаковать монстра, но это действие доступно только раз в пять ходов, после чего умение уходит на перезарядку. Монстры могут собираться в кучи, в этом случае, если атаковать кучу, умрут все монстры в ней. Монстры могут перемещаться только по уже вырытым туннелям, в том время как диггер может прокапывать новые; по этой причине монстры никогда не взаимодействуют с изумрудами, так как все они считаются зарытыми в земле.

Программа работает засчет бесконечного цикла, ее завершение осуществляется с помощью специальной функции, вызываемой в конце хода игрока. Эта функция приводит монстров в движение, а также проверяет, остались ли на карте изумруды и монстры и не достигли ли монстры диггера, при выполнении одного из этих условий функция выводит сообщение о победе либо поражении и возвращает значение истины, через которое и завершится функция main.

Во время написания программы я пользовался исключительно процедурным программированием и поэтому не создавал классов.

Я использовал только стандартную библиотеку std, а именно <iostream> .

# Заключение

В ходе выполнения расчетно-графической работы мной успешно была завершена работа по воссозданию аркадной игры, а также я получил возможность заглянуть «за кулисы» игрового процесса. В результате я получил немного измененную версию, в которой процесс происходит не в режиме real-time, а пошагово, что добавляет в нее элемент стратегии.

# Приложения

**Main.cpp:**

#include <iostream>

#include "everything\_but\_main.h"

using namespace std;

int main()

{

int t;

int timer=6;

int digger\_i=4;

int digger\_j=7;

int map[10][15]={

{2,2,2,2,2,2,2,3,2,2,2,2,2,2,2},

{2,1,1,2,2,2,2,0,2,2,2,2,1,1,2},

{2,1,1,2,2,2,2,0,2,2,2,2,1,1,2},

{2,2,2,2,2,2,2,0,2,2,2,2,2,2,2},

{3,0,0,0,0,0,0,9,0,0,0,0,0,0,3},

{2,2,2,2,2,2,2,0,2,2,2,2,2,2,2},

{2,2,2,2,2,2,2,0,2,2,2,2,2,2,2},

{2,1,1,2,2,2,2,0,2,2,2,2,1,1,2},

{2,1,1,2,2,2,2,0,2,2,2,2,1,1,2},

{2,2,2,2,2,2,2,3,2,2,2,2,2,2,2}

};

cout << "\t\t\t\t\tDIGGER" << endl;

cout << "\tWould you like to read the instructions? Type 1 if so or 0 if not. ";

cin >> t;

if (t) show\_instructions();

while (true)

{

show\_map(map);

show\_menu(map, digger\_i, digger\_j, &timer);

cin >> t;

switch(t)

{

case 0:

if(endstep(map, digger\_i, digger\_j, &timer)) return 0;

break;

case 1:

if (map[digger\_i][digger\_j-1]<3&&(digger\_i>=0)&&(digger\_j-1>=0)&&(digger\_i<10)&&(digger\_j-1<15))

{

map[digger\_i][digger\_j]=0;

map[digger\_i][digger\_j-1]=9;

digger\_j--;

if(endstep(map, digger\_i, digger\_j, &timer)) return 0;

break;

}

else

{

cout << "\tSuch command number isn't available.";

break;

}

case 2:

if (map[digger\_i][digger\_j+1]<3&&(digger\_i>=0)&&(digger\_j+1>=0)&&(digger\_i<10)&&(digger\_j+1<15))

{

map[digger\_i][digger\_j]=0;

map[digger\_i][digger\_j+1]=9;

digger\_j++;

if(endstep(map, digger\_i, digger\_j, &timer)) return 0;

break;

}

else

{

cout << "\tSuch command number isn't available.";

break;

}

case 3:

if (map[digger\_i-1][digger\_j]<3&&(digger\_i-1>=0)&&(digger\_j>=0)&&(digger\_i-1<10)&&(digger\_j<15))

{

map[digger\_i][digger\_j]=0;

map[digger\_i-1][digger\_j]=9;

digger\_i--;

if(endstep(map, digger\_i, digger\_j, &timer)) return 0;

break;

}

else

{

cout << "\tSuch command number isn't available.";

break;

}

case 4:

if (map[digger\_i+1][digger\_j]<3&&(digger\_i+1>=0)&&(digger\_j>=0)&&(digger\_i+1<10)&&(digger\_j<15))

{

map[digger\_i][digger\_j]=0;

map[digger\_i+1][digger\_j]=9;

digger\_i++;

if(endstep(map, digger\_i, digger\_j, &timer)) return 0;

break;

}

else

{

cout << "\tSuch command number isn't available.";

break;

}

case 5:

if (map[digger\_i][digger\_j-1]%3==0&&map[digger\_i][digger\_j-1]!=0&&(digger\_i>=0)&&(digger\_j-1>=0)&&(digger\_i<10)&&(digger\_j-1<15)&&timer==6)

{

map[digger\_i][digger\_j-1]=0;

timer=0;

if(endstep(map, digger\_i, digger\_j, &timer)) return 0;

break;

}

else

{

cout << "\tSuch command number isn't available.";

break;

}

break;

case 6:

if (map[digger\_i][digger\_j+1]%3==0&&map[digger\_i][digger\_j+1]!=0&&(digger\_i>=0)&&(digger\_j+1>=0)&&(digger\_i<10)&&(digger\_j+1<15)&&timer==6)

{

map[digger\_i][digger\_j+1]=0;

timer=0;

if(endstep(map, digger\_i, digger\_j, &timer)) return 0;

break;

}

else

{

cout << "\tSuch command number isn't available.";

break;

}

break;

case 7:

if (map[digger\_i-1][digger\_j]%3==0&&map[digger\_i-1][digger\_j]!=0&&(digger\_i-1>=0)&&(digger\_j>=0)&&(digger\_i-1<10)&&(digger\_j<15)&&timer==6)

{

map[digger\_i-1][digger\_j]=0;

timer=0;

if(endstep(map, digger\_i, digger\_j, &timer)) return 0;

break;

}

else

{

cout << "\tSuch command number isn't available.";

break;

}

break;

case 8:

if (map[digger\_i+1][digger\_j]%3==0&&map[digger\_i+1][digger\_j]!=0&&(digger\_i+1>=0)&&(digger\_j>=0)&&(digger\_i+1<10)&&(digger\_j<15)&&timer==6)

{

map[digger\_i+1][digger\_j]=0;

timer=0;

if(endstep(map, digger\_i, digger\_j, &timer)) return 0;

break;

}

else

{

cout << "\tSuch command number isn't available.";

break;

}

break;

default:

cout << "\tSuch command number isn't available.";

}

}

}

**everything\_but\_main.h:**

#ifndef EVERYTHING\_BUT\_MAIN\_INCLUDED

#define EVERYTHING\_BUT\_MAIN\_INCLUDED

void show\_instructions();

void show\_map(int map[10][15]);

void show\_menu(int map[10][15], int digger\_i, int digger\_j, int \*timer);

bool endstep(int map[10][15], int digger\_i, int digger\_j, int \*timer);

#endif // EVERYTHING\_BUT\_MAIN\_INCLUDED

**everything\_but\_main.cpp:**

#include <iostream>

#include "everything\_but\_main.h"

using namespace std;

void show\_instructions()

{

cout << "\tIn this game you control a machine called digger. It is marked" << endl;

cout << "\ton the map as number 9. It can move through corridors (number " << endl;

cout << "\t0 on the map) or dig tunnels though dirt (number 2). Your goal " << endl;

cout << "\tis to either collect all emeralds (number 1, they can be" << endl;

cout << "\tdug out by coming to the same square) or kill all monsters" << endl;

cout << "\twho try to destroy your digger (num. 3,6, etc). If they come" << endl;

cout << "\tto the same square as your digger, you will lose. However, you" << endl;

cout << "\tcan fight monsters which kills them once in 5 turns. Good luck!" << endl;

}

void show\_map(int map[10][15])

{

for (int i=0; i<10; i++)

{

cout << "\n\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"<< endl;

for (int j=0; j<15; j++)

{

cout << "|" << map[i][j] << "|";

}

}

cout << endl;

}

void show\_menu(int map[10][15], int digger\_i, int digger\_j, int \*timer)

{

cout << "\n\tList of available options:" << endl;

cout << "\t0. Do nothing" << endl;

if (map[digger\_i][digger\_j-1]<3&&(digger\_i>=0)&&(digger\_j-1>=0)&&(digger\_i<10)&&(digger\_j-1<15)) cout << "\t1. Move left" << endl;

if (map[digger\_i][digger\_j+1]<3&&(digger\_i>=0)&&(digger\_j+1>=0)&&(digger\_i<10)&&(digger\_j+1<15)) cout << "\t2. Move right" << endl;

if (map[digger\_i-1][digger\_j]<3&&(digger\_i-1>=0)&&(digger\_j>=0)&&(digger\_i-1<10)&&(digger\_j<15)) cout << "\t3. Move up" << endl;

if (map[digger\_i+1][digger\_j]<3&&(digger\_i+1>=0)&&(digger\_j>=0)&&(digger\_i+1<10)&&(digger\_j<15)) cout << "\t4. Move down" << endl;

if (map[digger\_i][digger\_j-1]%3==0&&map[digger\_i][digger\_j-1]!=0&&(digger\_i>=0)&&(digger\_j-1>=0)&&(digger\_i<10)&&(digger\_j-1<15)&&\*timer==6) cout << "\t5. Fight a monster to the left" << endl;

if (map[digger\_i][digger\_j+1]%3==0&&map[digger\_i][digger\_j+1]!=0&&(digger\_i>=0)&&(digger\_j+1>=0)&&(digger\_i<10)&&(digger\_j+1<15)&&\*timer==6) cout << "\t6. Fight a monster to the right" << endl;

if (map[digger\_i-1][digger\_j]%3==0&&map[digger\_i-1][digger\_j]!=0&&(digger\_i-1>=0)&&(digger\_j>=0)&&(digger\_i-1<10)&&(digger\_j<15)&&\*timer==6) cout << "\t7. Fight a monster above" << endl;

if (map[digger\_i+1][digger\_j]%3==0&&map[digger\_i+1][digger\_j]!=0&&(digger\_i+1>=0)&&(digger\_j>=0)&&(digger\_i+1<10)&&(digger\_j<15)&&\*timer==6) cout << "\t8. Fight a monster below" << endl;

cout << "\tType the corresponding option number: ";

}

bool endstep(int map[10][15], int digger\_i, int digger\_j, int \*timer)

{

bool emerald=true;

bool monster=true;

bool win=false;

if (\*timer<6) \*timer=\*timer+1;

cout << \*timer << endl;

for (int i=0; i<10; i++)

{

for (int j=0; j<15; j++)

if (map[i][j]%3==0&&map[i][j]!=0)

{

if (i<digger\_i&&map[i][j]%11!=0&&map[i+1][j]%3==0)

{

map[i+1][j]=(map[i+1][j]+map[i][j])\*11;

map[i][j]=0;

}

if (i>digger\_i&&map[i][j]%11!=0&&map[i-1][j]%3==0)

{

map[i-1][j]=(map[i-1][j]+map[i][j])\*11;

map[i][j]=0;

}

if (j<digger\_j&&map[i][j]%11!=0&&map[i][j+1]%3==0)

{

map[i][j+1]=(map[i][j+1]+map[i][j])\*11;

map[i][j]=0;

}

if (j>digger\_j&&map[i][j]%11!=0&&map[i][j-1]%3==0)

{

map[i][j-1]=(map[i][j-1]+map[i][j])\*11;

map[i][j]=0;

}

}

}

for (int i=0; i<10; i++)

{

for (int j=0; j<15; j++)

if (map[i][j]%11==0) map[i][j]=map[i][j]/11;

}

if (map[digger\_i][digger\_j]>9)

{

show\_map(map);

cout << "\n\tOh no! Monsters broke your digger!" << endl;

return true;

}

for (int i=0; i<10; i++)

{

for (int j=0; j<15; j++)

if (map[i][j]==1) emerald=false;

}

for (int i=0; i<10; i++)

{

for (int j=0; j<15; j++)

if (map[i][j]%3==0&&map[i][j]!=0&&map[i][j]!=9) monster=false;

}

win=emerald+monster;

if (!win) return false;

show\_map(map);

cout << "\n\tCongratulations! You won!" << endl;

return true;

}

# Литература

<https://ru.wikipedia.org/wiki/C%2B%2B>

<https://ru.wikipedia.org/wiki/Code::Blocks#cite_ref-1>

<http://www.futrega.org/digger/>