# Введение

Выбор программного обеспечения для разработки игры был проведен самостоятельно. Vim - один из мощнейших консольных текстовых редакторов с полной свободой настройки и автоматизации, возможными благодаря расширениям и надстройкам. Выбор вспомогательных библиотек был принят на основе необходимости использовать определенные их возможности.

Программа использует несколько потоков для отображения текущего состояния объектов на экране и расчета физики управляемого пользователем объекта.

Написание курсовой работы было разделено на следующие этапы:

1. Написание технического задания
2. Разработка логической части программы
3. Разработка графической части программы
4. Тестирование полученной программы.
5. Написание отчета.

# Техническое задание

## Введение

Данная курсовая работа предусмотрена учебным планом по специальности и выполняется в третьем семестре. Целью данной работы является закрепление практических навыков по курсу "Алгоритмические языки и программирование, Язык С++", расширение эрудиции в профессиональной области и получении и закрепление навыков самостоятельного решения инженерных задач на всех этапах.

## 1.1 Аннотация

1.1.1 Наименование программы

Название данного приложения будет напрямую перекликаться с темой курсовой работы: «Ncurses Jumps». Данное название целиком и полностью отражает функционал будущего приложения.

1.1.2 Краткая характеристика области применения

Программа предназначена для сферы игровой индустрии.

* 1. Основание для разработки

Учебный план по направлению 01.03.02 «Прикладная математика и информатика» (бакалавры). Создание приложения, способного дать пользователю возможность играть в игру.

## 1.2 Назначение разработки

Программа предназначена для игры одного пользователя в игру: «Ncurses Jumps».

## 1.4 Требования, предъявляемые к программе

1. Требования к функциональным характеристикам программы

Данный проект является компьютерной игрой, вследствие чего предусматривается одна категория пользователей - игроки. В процессе работы приложения пользователь является непосредственным участник игрового процесса и оказывает непосредственное влияние на него

Программы должна обладать следующим функционалом:

1. Графический функционал
2. Система взаимодействия игровых объектов
3. Графический интерфейс пользователя;
   * 1. Требования к техническим средствам, используемым при работе программы

Для нормального функционирования программы необходим IBM-совместимый компьютер под управлением операционной системы семейства Linux, монитор, около 1 МБ свободного места на жёстком диске для установки и работы программы и хранения пользовательских файлов и клавиатура.

* + 1. Требования к языкам программы и среде разработки программы

Для разработки используется язык программирования C++, в качестве инструментов разработки выступают текстовый редактор Vim, компилятор GNU GCC, поддерживающий стандарт языка C++11 и утилита make.

## 1.5 Требования к программной документации

1. Пояснительная записка в соответствии со стандартами РТУ МИРЭА (СМКО МИРЭА 7.5.1/04.И.05-16);
2. Проектная документация, составленная в соответствии с ГОСТ.

В процессе создания приложения вся проделанная работа документируется, должны быть сохранены все детали разработки, а также трудности, с которыми пришлось столкнуться. Всё выше перечисленное должно быть отражено в пояснительной записке, которая прилагается проекту.

## 1.6 Этапы разработки

В «жизненный цикл» приложения включены следующие этапы:

1. Написание технического задания
   1. Составление диаграммы классов
   2. Согласование архитектуры с преподавателем
   3. Написание текстовой части технического задания
2. Разработка логической части программы
   1. Написание классов для хранения и обработки информации о текущей игре
   2. Написание классов, отвечающих за взаимодействие объектов на игровом поле
3. Разработка графической части программы
   1. Написание классов для взаимодействия с окном ncurses
   2. Написание классов реализующих анимацию передвижения объектов по игровому полю
4. Тестирование полученной программы.
5. Написание отчета.
6. Оформление проектной документации; 14.09.2018-29.12.2018
7. Сдача проекта и защита курсовой работы. 29.12.2018

# 2**. Обзор способов организации данных и обоснование выбора структуры данных для эффективного выполнения операций.**

Реализация игры содержит 4 основных класса.

1. CANVAS – класс реализующий связь между экраном, представленном в виде двумерного массива символов, и отображающими функциями библиотеки ncurses. Через него проходит взаимодействие большинства объектов в программе.
2. GAME – основной класс отвечающий за инициализацию игровой сессии и карты препятствий, угроз и наград на игровом поле.
3. SPRITE – основной класс отвечающий за создание и отображение объектов на игровом поле.
4. MOVABLE\_SPRITE – вспомогательный класс, наследуемый от SPRITE, реализующий функционал движущегося управляемого объекта на игровом поле.

Пример инициализации игровой сессии приведен на рисунке (рис 2.2)



Рис 2.2.

# 3. Описание программы

## 3.1 Общие сведения

1. Наименование программы

Название разрабатываемой программы – игра «Ncurses Jumps».

1. Программное обеспечение, необходимое для функционирования программы

Требования к программному обеспечению:

Операционная система семейства Linux;

1 МБ свободного места на жестком диске.

1. Язык программирования, на котором написана программа

Программа написана на языке С++. Данный язык программирования хорошо подходит для работы с классами. С его помощью мы легко можем осуществить взаимодействие всех классов в программе. Так, для реализации курсовой работы был выбран набор инструментов для создания ПО на языке С/C++ состоящий из текстового редактора Vim, компилятора GNU GCC, поддерживающего стандарт языка C++11, и утилиты make.

## 3.2 **Функциональное назначение программы**

Игра создавалась для реализации шаблонной игры-платформера. Платформер - жанр компьютерных игр, в которых основной чертой игрового процесса является прыгание по платформам, лазанье по лестницам, собирание предметов, обычно необходимых для завершения уровня. Пример игрового процесса приведен на рисунке (рис 3.1)

Рис 3.1.

## 3.3 Описание логической структуры программы:

3.3.1 Алгоритмы, используемые в программе

3.3.2 Структура программы с описанием функций составных частей и связей между ними

За логику и взаимодействие объектов отвечают 3 основных класса.

1. GAME – класс, реализующий механики создания и хранения препятствий и наград на игровом поле, а также отвечающий за текущую игровую сессию и вызывающий метод для её завершения при выходе из игры или “смерти” игрового персонажа. Пример экрана завершения приведен на рисунке (Рис 3.2)
2. SPRITE – основной класс отвечающий за хранение информации о любом отдельно взятом объекте на игровом поле и отвечающий за его отображение на этом поле.
3. MOVABLE\_SPRITE – вспомогательный класс, наследуемый от класса SPRITE, который реализует функционал движущегося управляемого объекта на игровом поле и отвечает за ключевую логику коллизий объектов на поле.

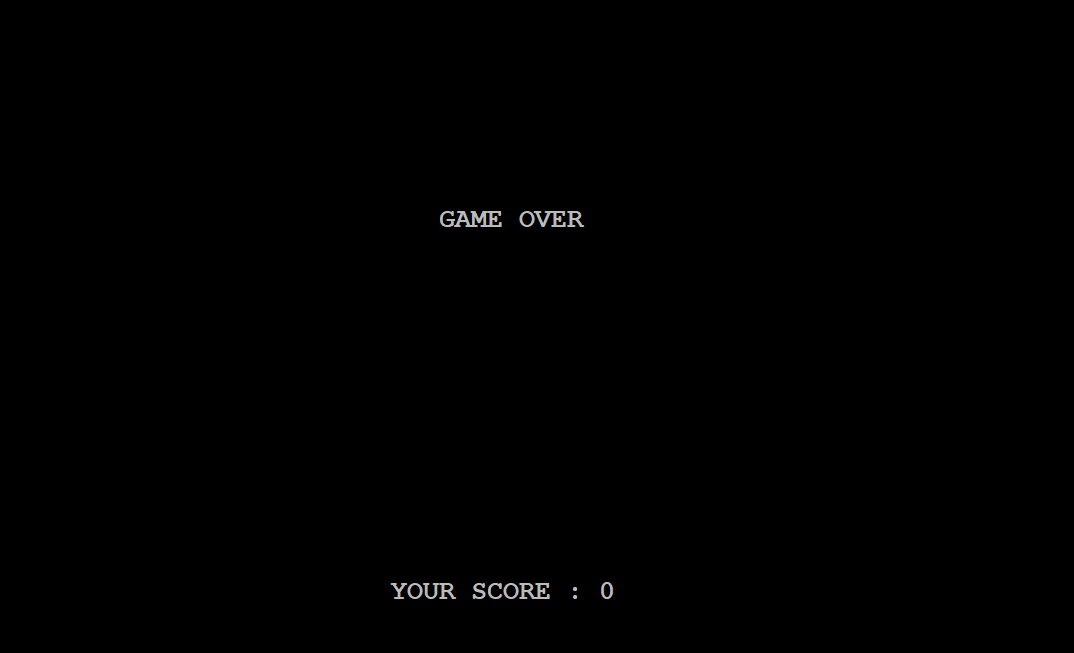


Рис 3.2

Графическая часть представлена одним главным классом CANVAS. Этот класс служит связующим звеном между логической частью программы и непосредственно объектами и методами библиотеки ncurses, которая используется для графического отображения объектов и их взаимодействий друг с другом на игровом поле.

## 3.4 **Технические средства**

Для нормального функционирования программы необходим IBM-совместимый компьютер под управлением операционной системы семейства Linux, монитор с разрешением экрана не ниже 1024х768, около 1 МБ свободного места на жёстком диске для установки и работы программы и хранения пользовательских файлов и клавиатура.

## 3.5 **Вызов программы**

Запуск программы осуществляется путём запуска исполняемого файла из терминала операционной системы Linux, содержащего название программы. После запуска появится приветствующее послание (рис 3.3)

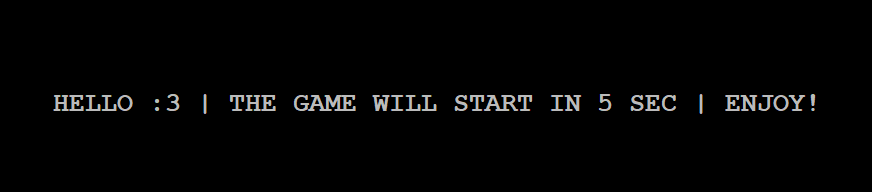


Рис 3.3.

По прошествии пяти секунд запустится и будет готова к использованию сама игра (рис 3.4)

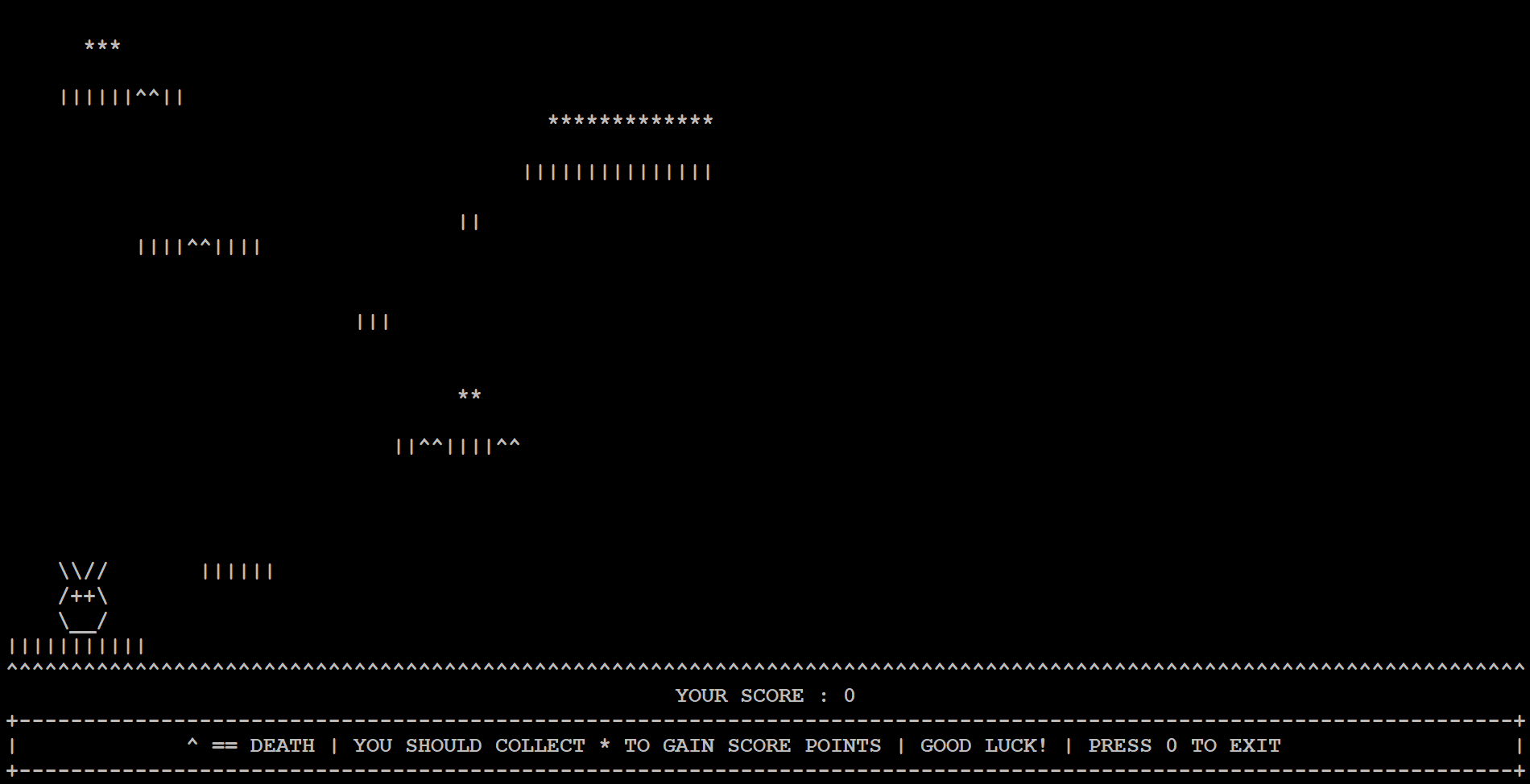


Рис 3.4.

# Заключение

В ходе написания курсовой работы с использованием полученных знаний была создана игра ««Ncurses Jumps»». В ходе проекта были задействованы навыки объектно-ориентированного программирования на языке C++ и использования текстового редактора Vim, компилятора GNU GCC, поддерживающего стандарт языка C++11, и утилиты make.

Готовый программный продукт полностью удовлетворяет требованиям технического задания.