МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра информатики и технологий программирования

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовому проекту

на тему

«Змейка в Borland C++»

Выполнил:

студент гр.653502

Длужневский В.Д

Проверил: Козуб В.Н.

г. Минск

2017 год

**СОДЕРЖАНИЕ**

**1. ВВЕДЕНИЕ**

**2. ЦЕЛИ И ЗАДАЧИ**

**3. ПРАВИЛА ИГРЫ**

**4. РАЗРАБОТКА ПРОГРАММЫ**

**5. ЗАКЛЮЧЕНИЕ**

**6. СПИСОК ИСТОЧНИКОВ**

**Введение.**

Игры уже давно стали неотъемлемой частью жизни людей. Наверняка игры зародились уже в древности. Уже в средние века игры были распространены среди всех слоёв населения. Эти игры были как интеллектуальные, так и спортивные. Приводя примеры спортивных игр стоит упомянуть гладиаторские бои и олимпийские игры, а среди популярных в то время интеллектуальных игр, можно упомянуть шахматы и домино.

Сегодня игры перешли на совершенно новый уровень. Рост игровой индустрии начал наблюдаться в конце 20 начале 21 века. Сначала большую популярность и спрос получили игры на телевизорах с использованием игровых приставок. Самыми примечательными из них были Sega, Dendy, первая и дефицитная Playstation и Nintendo. В наше время игры ушли далеко вперёд в развитии. Сегодня игры выпускаются на многих платформах таких как : Playstation, Xbox, Nintendo и Pc. Также не стоит забывать и про мобильные приложения.

Ещё в начале развития игровой индустрии была игра которая была портирована на несколько платформ. Этой игры и коснётся мой курсовой проект. Змейка. Возвращаясь в начало 2000 сложно представить какую либо платформу без этой игры. Змейка была на компьютерах и даже на кнопочных сотовых телефонах.

Мой курсовой проект является консольной реализацией одной из наиболее простых версий той самой змейки которая знакома многим с детства. Для написания кода я использовал Borland C++. Для курсового проекта был выбран язык программирования C++, так как он предназначен для разработки самых разнообразных приложений, но при этом сохраняет выразительность и элегантность, присущую языкам C.

Целями курсового проекта являются:

**\*** Закрепление знаний среды разработки С++.

**\*** Изучение графических возможностей консоли.

**\*** Первые шаги в разработке простейших игр

**Цели и задачи.**

Основной целью данной курсовой работы является изучение и создание игры змейка с использованием консоли, а также программирование компонентов на языке C++.

Данная работа состоит из 3 разделов: первый раздел теоретический, содержащий общие сведения по теме, второй раздел практический, где описывается код и идея игры. Третий раздел – руководство пользователя.

Основные задачи, выполнение для написания работы:

* Изучение методов для создания графических объектов в консоли.
* Изучение жанра игр-аркад.
* Изучение правил змейки.
* Написание кода программы
* Создание игры жанра аркада.
* Анализ полученной игры

**Правила игры.**

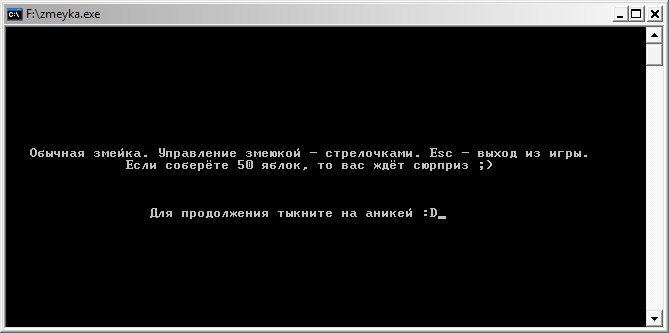
Правила игры в змейку уже давно всем нам знакомы. Начиная с первой версии целью игры всегда было есть фрукты, яблоки, кубики и расти до определённых размеров. Говоря конкретно о данной версии правила предельно просто. Для победы игроку необходимо собрать 50 фруктов, которые появляются на поле, если игрок врезается, то он проигрывает.

**Разработка программы.**

В данном разделе описаны основные функции и руководство по разработке и работе программы.

Часть 1. Запуск приложения

Данная форма первое что мы увидим после запуска программы



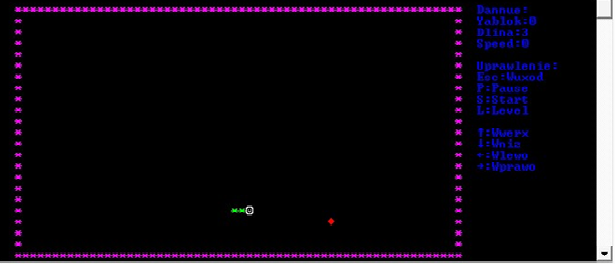
На начальном экране отображаются краткие правила игры, для начала игры достаточно нажать любую клавишу.

Вывод начального текста осуществляется довольно простым способом, мы используем GotoXY() задавая начальные координаты для написания текста после чего используем printf(“”); для непосредственного вывода текста на наш экран. И завершаем написанием строки для ожидания ввода любого символа.

Часть 2. Экран игры и игровой процесс

После ввода символа начинается сама игра

Ниже представлен вид главного экрана игры:



1)Первая и самая простоя вещь, хотя и не маловажная – это стены.

Стены задаются поэтапно с использованием в коде всё того же элемента GotoXY(). Для этого нам достаточно просто повторить его несколько раз задавая координаты на которых располагается символ “\*” являющийся частью нашей стенки. Создав все четыре стены мы обозначили область игрового пространства.

2)Следующее что нужно для нашей игры – это ресурсы(пропитание для змеи).Для этого мы создаём рандом который будет заполнять наше поле яблоками. Для этого мы используем отдельный рандом для каждой из координат, а именно x и y. Так как наш рандом ограничен то нам не стоит беспокоится о том что яблоко окажется на стенке.

Однако остаётся вероятность того что яблоко может сгенерироваться на нашу змею. Для этого нам необходимо использовать строку проверки:

if (x == g.gaduka.t[i].X && y == g.gaduka.t[i].Y

Предварительно мы вводим переменную I, после чего используем её для создания цикла. Так что программа будет генировать новые значения пока не найдёт то которое не будет накладываться на позицию змеи.

После того как наша программа сгенерировала и проверила координаты для нашего яблока нам необходимо закрепить его на этом месте. Для этого мы присваиваем эти координаты до тех пор пока змея его не проглотит. После чего позиция яблока сбрасывается.

3)Третья часть самая ёмкая – создание самой змеи. Первое что нужно реализовать это функция, которая будет определять стартовую позицию, скорость и размер змеи. Мы реализуем всё достаточно просто, задав начальные координаты старта и фиксированный размер змеи равный 3. После прорисовки змеи сразу же запускается процесс создания яблока.

Следующая функция, которую необходимо реализовать – функция передвижения змеи. Чтобы изобразить передвижение змеи мы будем использовать две основные строки:

SetConsoleCursorPosition(hConsole, head);

SetConsoleTextAttribute(hConsole, 0x0a);

printf("\*");

И

SetConsoleCursorPosition(hConsole, next);

SetConsoleTextAttribute(hConsole, 0x0f);

printf("%c", 1);

Первая используется для перекраски клетки в хвост змеи, а вторая для перекраски клетки в голову змеи. Аналогичный код используется для закрашивания той ячейки в которой больше не находиться змея, единственным отличием будет то что вместо символов в строке printf(“ ”) мы просто оставляем пустое место. После того как все эти операции выполнены программа возобновляет функцию движения. Данная функция используется также при поедании змеёй яблок, но в этом случае при каждом поедании создаётся новый массив, который больше предыдущего на одну единицу.

Теперь вернёмся к передвижению змеи. Не стоит забывать, что наша змея при своём передвижении может упереться в стену или же в саму себя. Чтобы это узнать нам нужно узнать следующую координату в направлении, которое задал пользователь. Для определения координат нам понадобятся две строки:

next.X = head.X + g.dx;

next.Y = head.Y + g.dy;

Таким образом наша программа определяет следующую координату направления, но этого недостаточно. Чтобы определить врежется ли наша змея в стену, в себя или же соберёт яблоко нам нужно добавить 3 простых условия.

Первое будет сравнивать новые координаты с координатами стены. Второе условие будет сравнивать новые координаты с остальными координатами змеи. Третье условие будет сравнивать новые координаты с координатами яблока, если они совпадут, то запустится функция роста змеи, которая обсуждалась ранее. В первых двух функция в случае совпадения игра заканчивается проигрышем.

4)Следующий пункт – это панель информации находящаяся справа от игрового поля. После каждого хода на эту панель выводятся обновлённые данные о количестве яблок, скорости и длине змеи. Также на правой панели указаны клавиши управления. Мы уже говорили о том, что в случае нажатия ESC мы выходим из игры, но кроме этого есть ещё две функции это pause и start. Легко догадаться что первая функция останавливает программу, а вторая запускает её. Всё это выполняется после каждого хода благодаря функции plus.

5) Победа или проигрыш. Победа или же проигрыш игрока определяется в основном благодаря 3 и 4 пункту. Если в функции проверки в одном из первых условий координаты совпадут, то игрок проигрывает и на экран выводиться надпись.



Победа наступает тогда, когда параметр яблок доходит до 50 единиц, только в этом случае игрок побеждает и на экран выводиться:



**Заключение.**

В результате работы над данным курсовым проектом была разработана игровая программа змейка и документация к ней. В процессе работы над программой были развиты навыки работы с языком программирования С++.

**Список источников:**

[http://www.cyberforum.ru](http://www.cyberforum.ru/)

h[ttps://en.wikipedia.org](https://en.wikipedia.org/)

https://www.microfocus.com/jajp/borland/?utm\_medium=301&utm\_source= borland.com

<https://msdn.microsoft.com/en-us/library/60k1461a.aspx>