Министерство образования Республики Беларусь

Учреждение образования

«Белорусский государственный университет информатики и радиоэлектроники»

Филиал

«Минский радиотехнический колледж»

ОБУЧАЮЩАЯ ПРОГРАММА «НОТНЫЙ ГРАДИЕНТ»

Пояснительная записка

к курсовому проекту по предмету:

«Основы алгоритмизации и программирования»

КП 62491.028106.081 ПЗ

Руководитель (Назаров Л.В.)

Учащаяся гр. 62491 (Хинкель Е.Р.)

2018

Министерство образования Республики Беларусь

Учреждение образования

«Белорусский государственный университет информатики и радиоэлектроники» филиал

«Минский радиотехнический колледж»

УТВЕРЖДАЮ

Председатель ПМК \_\_\_\_\_\_\_\_\_\_\_

<<\_\_\_>>\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_г

**З А Д А Н И Е**

для курсового проектирования поа предмету *Основы алгоритмизации и программирования*

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

учащейся \_\_\_\_\_\_\_*II\_*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ курса \_\_\_62491\_\_\_\_групы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

специальности\_\_*2-40 01 01*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ *Хинкель Екатерины Райнхардовны*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(фамилия, имя, отчество)

Тема курсового проекта (работы) и исходные данные *Нотный градиент (обучающая программа) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

\_\_*исходные данные: текстовые файлы с данными,\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

При выполнении курсового проекта (работы) по названной теме должны быть представлены:

**І. Расчетно-пояснительная записка**

*Введение 1Постановка задачи 1.1 Назначение и функции программы 1.2 Математическая модель задачи 1.2.1 Математическая формулировка задачи 1.2.2 Выбор метода решения задачи 1.3 Информационная база задачи 1.3.1 Входная информация 1.3.2 Выходная информация 1.4 Система меню 1.5 Система помощи 1.6 Контрольные примеры 1.7 Алгоритм программы 2 Описание программы 3 Описание применения Заключение Список использованных источников\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

**ІІ. Графическая часть проекта (работы)**

Лист 1 Схема «Схема работы системы»

Лист 2 Схема «Схема автовоспроизведения»

Дата выдачи задания «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_ г.

Срок сдачи завершенного проекта «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_ г.

**Преподаватель-руководитель курсового проекта (работы**) **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Заведующий предметно-методической комиссии:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Содержание**

[Введение 4](#_Toc517941954)

[1. Постановка задачи 6](#_Toc517941955)

[**1.1** **Назначение и функции программы** 6](#_Toc517941956)

[1.2 Математическая модель задачи 7](#_Toc517941957)

[**1.2.1** **Математическая формулировка задачи** 7](#_Toc517941958)

[**1.2.2** **Выбор метода решения задачи** 7](#_Toc517941959)

[1.3 Информационная база задачи 8](#_Toc517941960)

[1.3.1 Входная информация 8](#_Toc517941961)

[1.3.2 Выходная информация 8](#_Toc517941962)

[1.3.3 Нормативно-справочная информация 9](#_Toc517941963)

[1.4. Система меню 10](#_Toc517941964)

[1.5 Система помощи 11](#_Toc517941965)

[2 Описание программы 15](#_Toc517941966)

[На главной форме расположены шейпы - все 36 клавиш пианино (3 октавы). 15](#_Toc517941967)

[Рисунок 3.5 Карта октав пианино 19](#_Toc517941968)

[Заключение 20](#_Toc517941969)

[Список использованных источников 21](#_Toc517941970)

[Приложение А (основные функции и процедуры)](#_Toc517941971) [Текст программы 22](#_Toc517941972)

[Приложение В (справочное) Схема алгоритма Автовоспроизведения 30](#_Toc517941973)

Изм.

Лист

№ докум.

Подпись

Дата

Лист

1

*КП62491.028106.081 ПЗ*

Разраб.

*Хинкель Е.Р.*

Провер.

Назаров Л.В.

Н. Контр.

Утв.

*Обучающая программа на тему «Нотный градиент»*

*Пояснительная записка*

Лит.

Листов

2

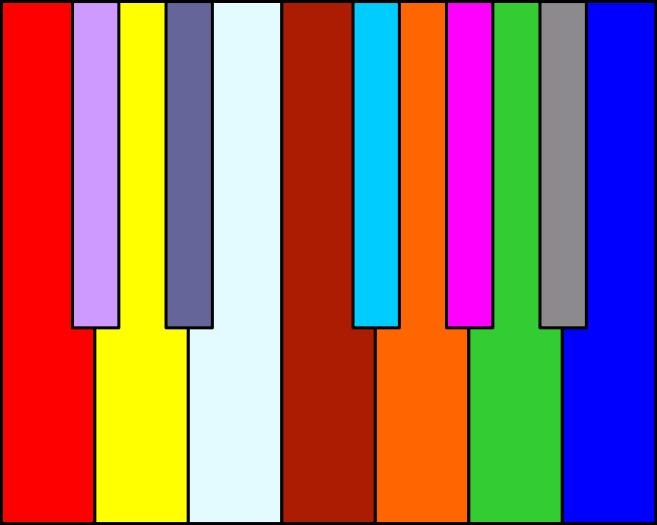
*МРК*

# Введение

«Радужный градиент» - музыкальный тренажёр для игры на пианино по цветам, нежели по нотам. Мелодию можно записать в виде цветных прямоугольников, цвет которых отвечает за саму ноту, а длина – за длительность.

В мире существует множество методов лечения:  режим, диета, лекарства, операция и т.д. Однако два из них являются особенно интересными. Первый – музыкотерапия – это исцеление музыкой. Музыкотерапия как наука стала известной в 19 веке, а в 20 веке стала невероятно популярной во многих странах. Второй же – цветотерапия – исцеление цветом. В египетских храмах археологи обнаружили помещения, конструкция которых заставляла преломлять солнечные лучи в тот или иной цвет спектра. В данной программе будут объединены эти две терапии.

Исаак Ньютон и Александр Скрябин тоже были заинтересованы в данном вопросе. Каждый имел своё представление о соотношении ноты и цвета. Например, у Ньютона система получилась больше для математиков, нежели музыкантов: он разложил радугу ровно по 7 нотам фортепиано (не затрагивая даже диезы и бемоли). Скрябин же полагался исключительно на своё световосприятие. В его системе До олицетворяло адское пламя (красный), Ми – беспристрастный разум (белый) и т.д.

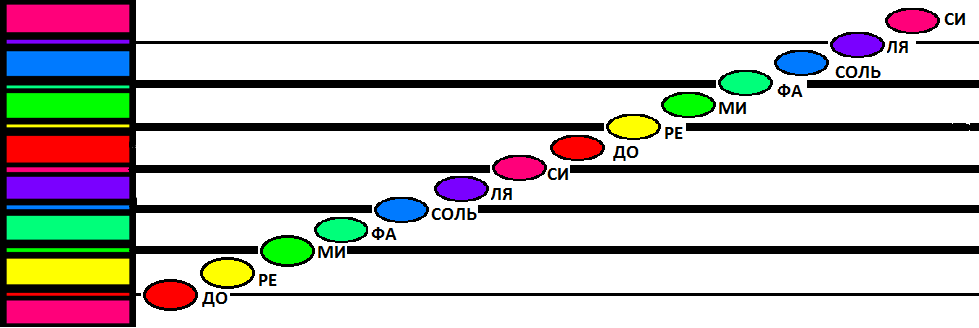
  
Рисунок «Музыкальные цветоряды Ньютона и Скрябина»

«Нотный градиент» берет за основу идею Ньютона – радужный спектр, однако распределенный не по 7-ми тонам, а по 12-ти полутонам. Оттенки получились из промежуточных Red, Green и Blue:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ЦВЕТ | НОТА | R | G | B |
| красный | До | 255 | 0 | 0 |
| Оранжевый | До-диез (Ре-бемоль) | 255 | 122 | 0 |
| Желтый | Ре | 255 | 255 | 0 |
| Салатовый | Ре-диез (Ми-бемоль) | 122 | 255 | 0 |
| Зеленый | Ми | 0 | 255 | 0 |
| Бирюзовый | Фа | 0 | 255 | 122 |
| Голубой | Фа-диез (Соль-бемоль) | 0 | 255 | 255 |
| Морской | Соль | 0 | 122 | 255 |
| Синий | Соль-диез (Ля-бемоль) | 0 | 0 | 255 |
| Фиолетовый | Ля | 122 | 0 | 255 |
| Розовый | Ля-диез (Си-бемоль) | 122 | 0 | 122 |
| Малиновый | Cи | 255 | 0 | 0 |

Рисунок «Распределение цветов по нотам»

Теперь обычные ноты превращаются в цветоноты:

Рисунок «Нотная гамма в цвете»

Пользователю будет доступно 2 режима (игра и цветоноты), смена октав (3 варианта), инструкция по игре и написанию цветонот (конкретно для данной программы). При обычной игре никаких правил нет, игрок предоставлен сам себе. В режиме цветонот, ему предстоит отыграть последовательно каждую записанную цветоноту. В качестве помощи новичкам, доступно программное воспроизведение мелодии с отображением нажатий клавиш пианино.

Пояснительная записка содержит 3 раздела:

1. «Постановка задачи» – назначение данной программы, метод решения задачи, входные, выходные данные и конкретные примеры
2. «Описание программы» – минимальные требования для работы программы, её интерфейс (внешнего вида) и возможности.
3. «Описание применения» – правила подготовки исходной информации, руководство по использованию каждой функции приложения.

1. **Постановка задачи**

Разработка игры «Нотный градиент» подразумевает реализацию игры на виртуальном пианино, отображение цветонот на экран и их дальнейшее программное воспроизведение.

Предметной областью данной программы является алгоритмизация игр. Будут используются алгоритмы высчитывающие номер ноты и ее длительность, а также осуществляющие цветонотную запись песен.

* 1. **Назначение и функции программы**

В данной программе реализована игра пользователя и программное воспроизведение заготовленных цветонот.

Для воспроизведение нот предусмотрено нажатие как по клавиатуре, так и мышкой. Пользователю разрешается зажимать одновременно несколько нот, образуя аккорд (количество нот в нем ограничено возможностями клавиатуры). Также на мониторе показано 3 октавы (от До до Си), которые можно переключать между собой, повышая или понижая их.

Программное воспроизведение поможет игроку проследить за длительностями нот, а также порядком их нажатия, благодаря отображению игры на экран. Для того, чтобы послушать мелодию, надо подготовить цветоноты в текстовом файле и открыть их. Для этого в справке подробно описывается алгоритм записи цветонот, используя последовательности чисел.

* 1. **Математическая модель задачи** 
     1. **Математическая формулировка задачи**

Для прорисовки цветонот потребуется формула, в которой описывается, что длина прямоугольника ноты прямо пропорциональна ее длительности. Для целой ноты берется 256 пунктов (при многократном делении на 2 получаются целые значения), для половинной будет – 128, для четвертой – 64 и т.д. Если встречается аккорд, то цветонота делится поровну на количество нот в этом аккорде:

L – длина для одной ноты

A – количество нот в аккорде (1 если одна нота)

T – длительность ноты (половинная – 2, четвертая – 4 и т.д.)

* + 1. **Выбор метода решения задачи**

Зажатая нота проверяется на соответствие с нотой в цветонотах. Если результат положительный, эта нота будет воспроизведена и выдана следующая. При автовоспроизведении поочередно проигрываются все ноты по цветонотам. Для аккордов предусмотрено одновременное воспроизведение сразу нескольких нот путем запуска независимых друг от друга аудио-потоков (wav) через консоль Windows.

* 1. **Информационная база задачи**
     1. **Входная информация**

Цветоноты представляют собой текстовый файл со следующими данными: название мелодии, номера нот, длительности нот, кол-во нот. После описания каждой информации ставится 0 (ноль).

* Название – любые символы.
* Номера нот – числа от 1 до 36 (все ноты от До 1-ой октавы до Си 3-ей октавы).
* Длительности нот – числа от 1 до 16 (желательно больше не ставить). 1 – целая, 2 – половинная, 4 – четвертая, 8- восьмая, 16 – шестнадцатая.
* Количество нот – сколько нот будет зажато одновременно. То есть, если нужен аккорд из 3 нот, то пишется 3, если нужна просто одиночная нота, то – 1.



Рисунок 1.1 Текстовый файл с цветонотами мелодии «Lalala»

* + 1. **Выходная информация**

Каждая цветонота представляет собой цветной прямоугольник, ограниченный черной линией. В аккорде, как видно ниже, этих разграничений нет между его отдельными составляющими.

  
Рисунок 1.2 Исходные данные «по цветам»

F:\RGB-PIANO\Снимок2.JPG  
Рисунок 1.3 Нотная запись по цветам

* + 1. **Нормативно-справочная информация**

Для корректной работы программе требуются некоторые вспомогательные данные:

* RGB-коды для каждой клавиши пианино (colors.txt)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Клавиша | Зажатая | | | Незажатая | | |
| R | G | B | R | G | B |
| 1 | 255 | 0 | 0 | 255 | 255 | 255 |
| 2 | 255 | 125 | 0 | 0 | 0 | 0 |
| 3 | 255 | 255 | 0 | 255 | 255 | 255 |
| 4 | 125 | 255 | 0 | 0 | 0 | 0 |
| 5 | 0 | 255 | 0 | 255 | 255 | 255 |
| 6 | 0 | 255 | 125 | 255 | 255 | 255 |
| 7 | 0 | 255 | 255 | 0 | 0 | 0 |
| 8 | 0 | 125 | 255 | 255 | 255 | 255 |
| 9 | 0 | 0 | 255 | 0 | 0 | 0 |
| 10 | 125 | 0 | 255 | 255 | 255 | 255 |
| 11 | 255 | 0 | 255 | 0 | 0 | 0 |
| 12 | 255 | 0 | 125 | 255 | 255 | 255 |
| 13 | 255 | 0 | 0 | 255 | 255 | 255 |
| 14 | 255 | 125 | 0 | 0 | 0 | 0 |
| 15 | 255 | 255 | 0 | 255 | 255 | 255 |
| 16 | 125 | 255 | 0 | 0 | 0 | 0 |
| 17 | 0 | 255 | 0 | 255 | 255 | 255 |
| 18 | 0 | 255 | 125 | 255 | 255 | 255 |
| 19 | 0 | 255 | 255 | 0 | 0 | 0 |
| 20 | 0 | 125 | 255 | 255 | 255 | 255 |
| 21 | 0 | 0 | 255 | 0 | 0 | 0 |
| 22 | 125 | 0 | 255 | 255 | 255 | 255 |
| 23 | 255 | 0 | 255 | 0 | 0 | 0 |
| 24 | 255 | 0 | 125 | 255 | 255 | 255 |
| 25 | 255 | 0 | 0 | 255 | 255 | 255 |
| 26 | 255 | 125 | 0 | 0 | 0 | 0 |
| 27 | 255 | 255 | 0 | 255 | 255 | 255 |
| 28 | 125 | 255 | 0 | 0 | 0 | 0 |
| 29 | 0 | 255 | 0 | 255 | 255 | 255 |
| 30 | 0 | 255 | 125 | 255 | 255 | 255 |
| 31 | 0 | 255 | 255 | 0 | 0 | 0 |
| 32 | 0 | 125 | 255 | 255 | 255 | 255 |
| 33 | 0 | 0 | 255 | 0 | 0 | 0 |
| 34 | 125 | 0 | 255 | 255 | 255 | 255 |
| 35 | 255 | 0 | 255 | 0 | 0 | 0 |
| 36 | 255 | 0 | 125 | 255 | 255 | 255 |

* Коды для клавиш клавиатуры, записанные последовательно от 1-ой до 36-ой ноты (keys.txt) :

{90,83,88,68,67,86,71,66,72,78,74,77,188,76,190,186,191,81,50,87,51,69,52,82,84,54,89,

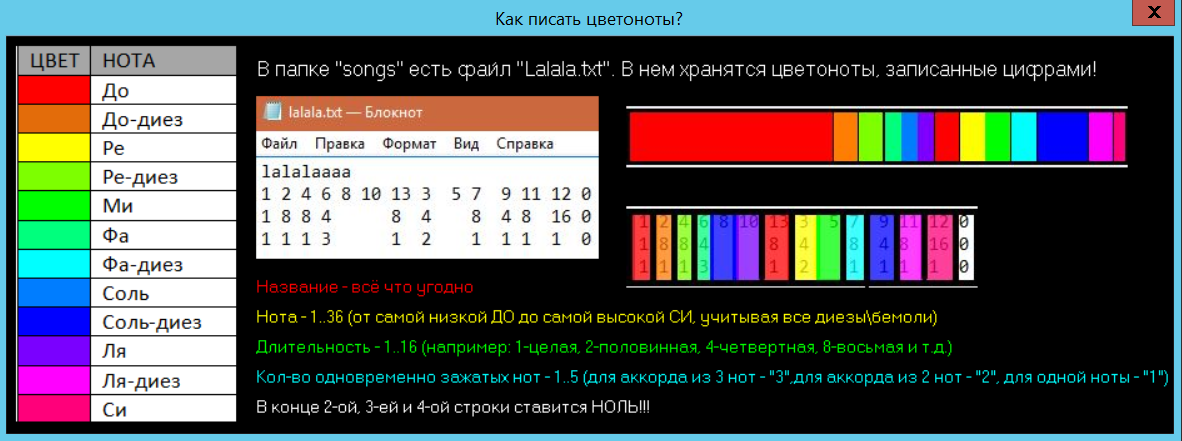
55,85,73,57,79,48,80,189,219}

* wav-файл для каждой ноты в 5-ти октавах, пронумерованных от 1 до 60 (папка «notes»)
  1. **Система меню**
* Режим игры
  + - Игра
    - Цветоноты
* Октава
  + - Первая
    - Вторая
    - Третья
* Помощь
  + - Как играть?
    - Как писать цветоноты
    - Автор и его друзья
  1. **Система помощи**
* Как играть?



Рисунок 1.4 Помощь (Как играть?)

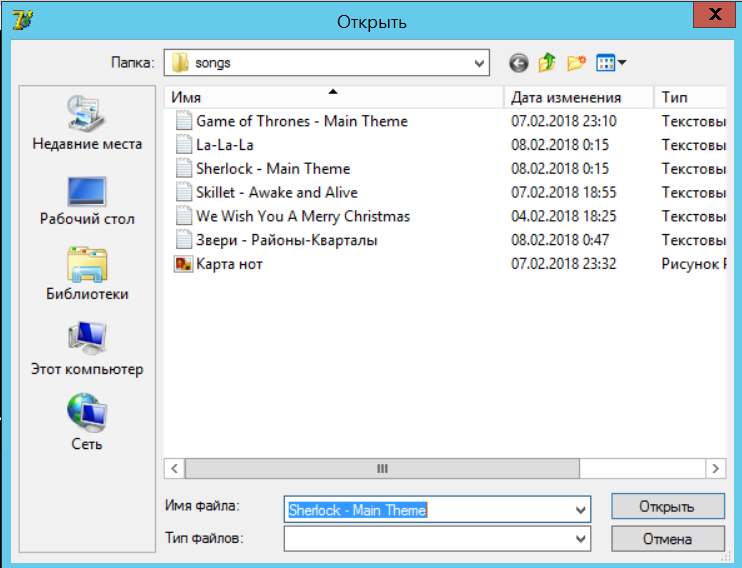
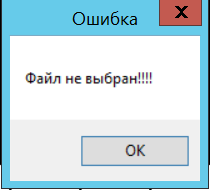
* Как писать цветоноты?

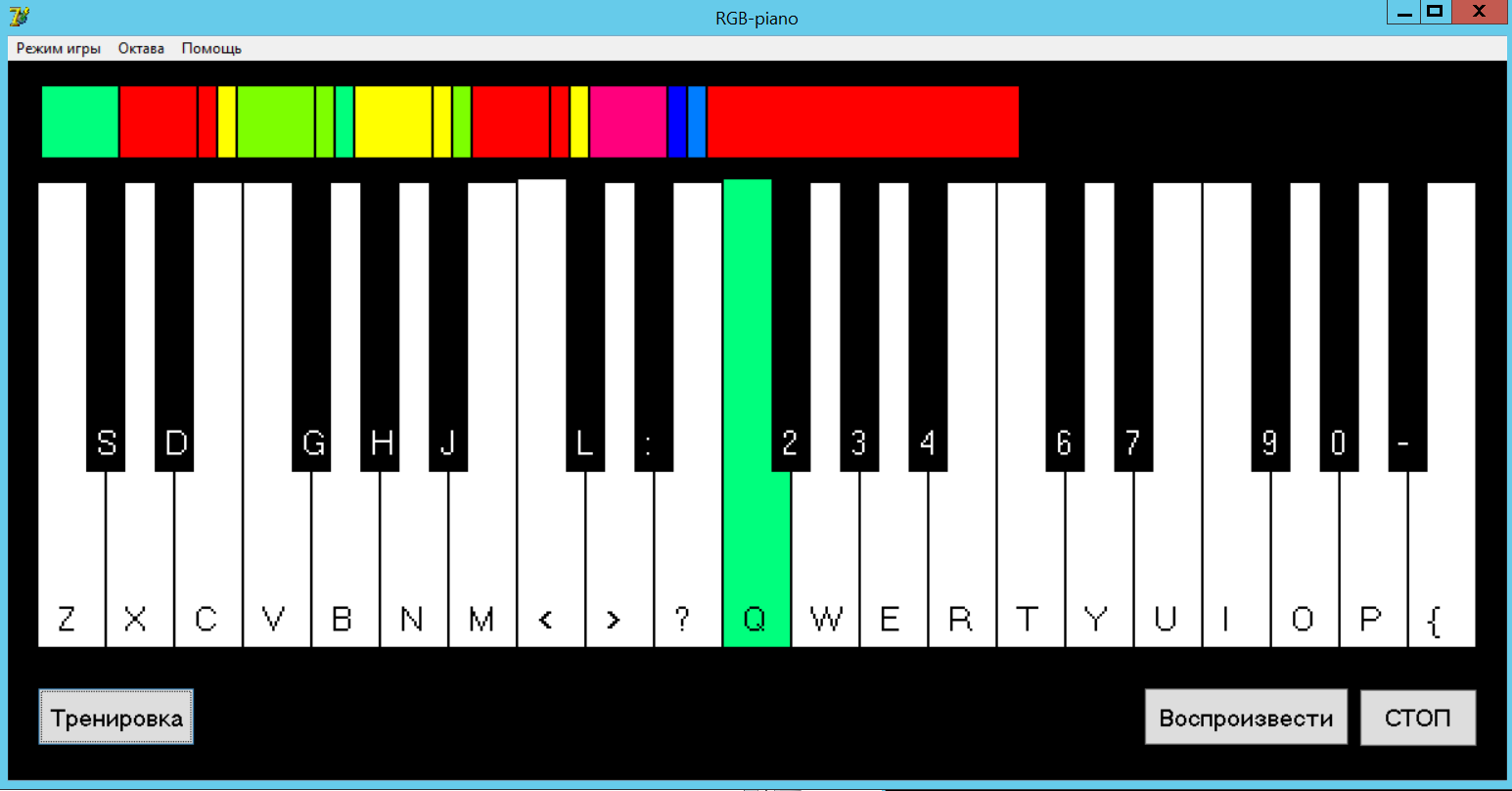
Рисунок 1.5 Помощь (Как писать цветоноты?)

* Автор и его друзья

Рисунок 1.6 Помощь (Автор и его друзья)

* 1. **Контрольные примеры**

Рисунок 1.7 Выбор файла с цветонотами  
Рисунок 1.8 Попытка начать тренироваться, не выбрав файл с цветонотами

Рисунок 1.9 Игра в режиме тренировки мелодии из «Шерлока»

* 1. **Алгоритм программы**

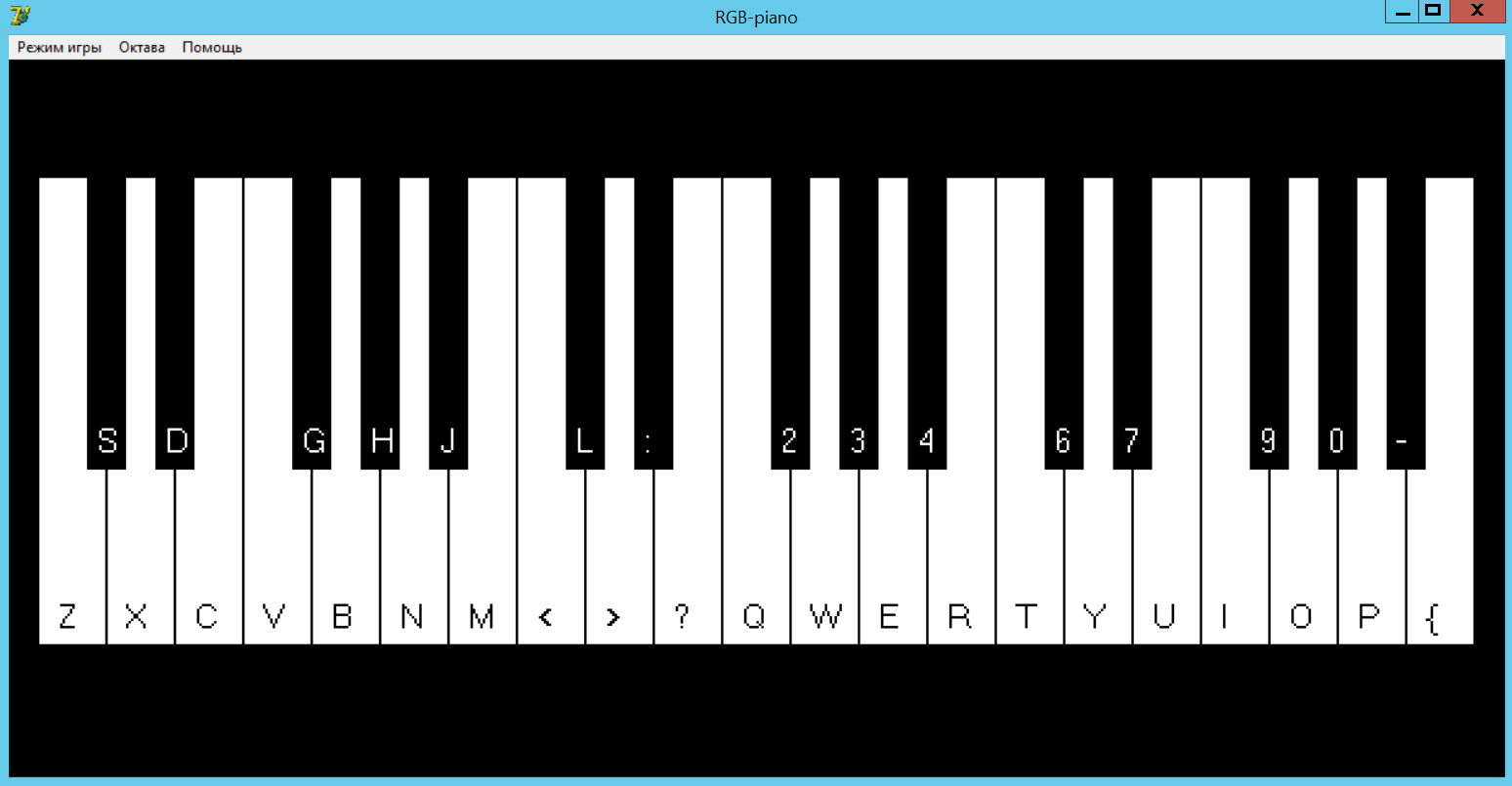
В данной программе отсутствуют какие-либо сложные алгоритмы с рекурсиями и бэкрекингами. Самой большой процедурой можно считать событие KeyDown, ибо там осуществляется множество проверок: на соответствие клавиши с доступными для нажатия, на соответствие с требуемой нотой (в режиме «Цветоноты»), на количество сыгранных нот, на количество сыгранных нот в аккорде. А также производятся действия окрашивания шейпов (клавиш пианино) на форме в соответствующие им цвета (зажатые\незажатые).

Реализация аккордов в цветонотах происходит с помощью указания их в текстовом файле цветонот. Если 3 ноты, значит будет «3», если одона – «1». Благодаря этому, программа отсчитывает ноты по блокам и отдельно по нотам одновременно, что позволяет ей во внутреннем цикле проиграть сразу несколько нот как одну, но в то же время отобразить их пользователю как 3 разные ноты. Для этого создаётся 2 счетчика – I и J.

Чтобы проверить, зажаты ли все ноты в аккорде и можно ли переходить к следующему, создается множество с нотами этого аккорда, и с каждым нажатием одной из него ноты, она исключается. Таким образом, когда множество становится пустым, делается переход на следующий аккорд (блок нот).

1. **Описание программы**

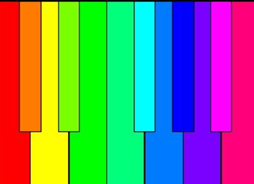
На главной форме расположены шейпы - все 36 клавиш пианино (3 октавы).

Рисунок 2.1 Главная форма с клавишами

На них отображены символы клавиатуры, чтобы пользователю было легче сориентироваться, на что ему надо нажать.

  
Рисунок 2.2 Карта клавиш

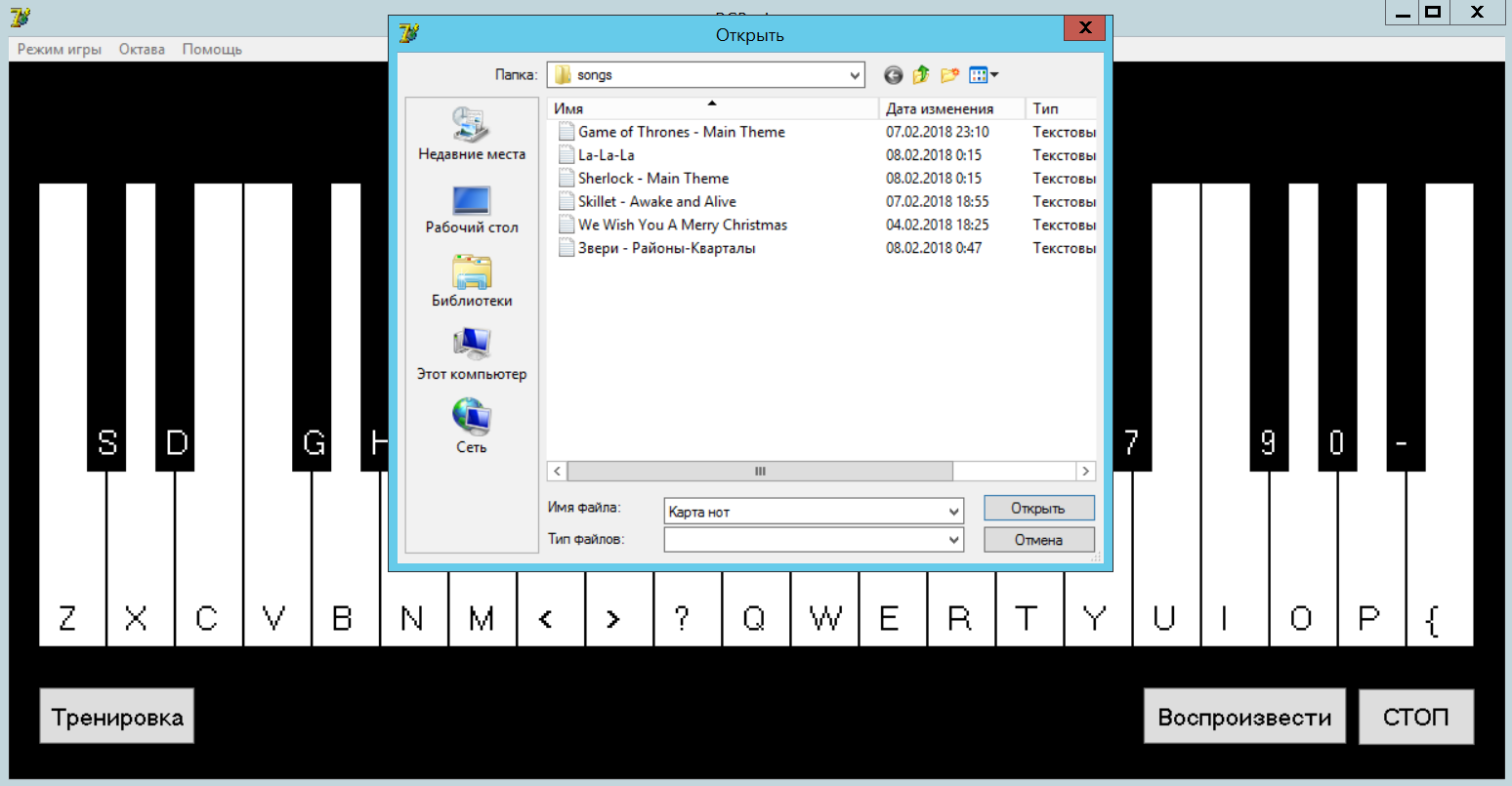
Нажимая на клавишу, она окрашивается в свой цвет, назначенный её ноте. Осуществить это можно путем смены цвета для шейпа в процедуре KeyDown и KeyUp.

  
Рисунок 2.3 Карта цветонот

Для того, чтобы воспроизведение ноты не осуществлялось ежемилисекундно во время нажатия клавиши, надо установить флаг нажатия. Он окрашивается в моменты нажатия и отпускания клавиши. Так программа будет понимать, когда включить аудио и когда его можно возобновить.

Чтобы воспроизвести сразу несколько нот одновременно, можно воспользоваться свойством аудио-потоков формата «wav». Вызывая их консольно, создается обособленный поток, который не зависит от остальных, а значит можно запустить сразу несколько. Для этого надо указать параметр «shareble» в консольном запросе «open waveaudio shareable». Если этого не сделать, работа производится только над одним потоком, а соответственно одним файлом, что позволяет только последовательное воспроизведение.

Для выбора песни в режиме «Тренировка» открывается окошко OpenDialog в папке с песнями (/songs):

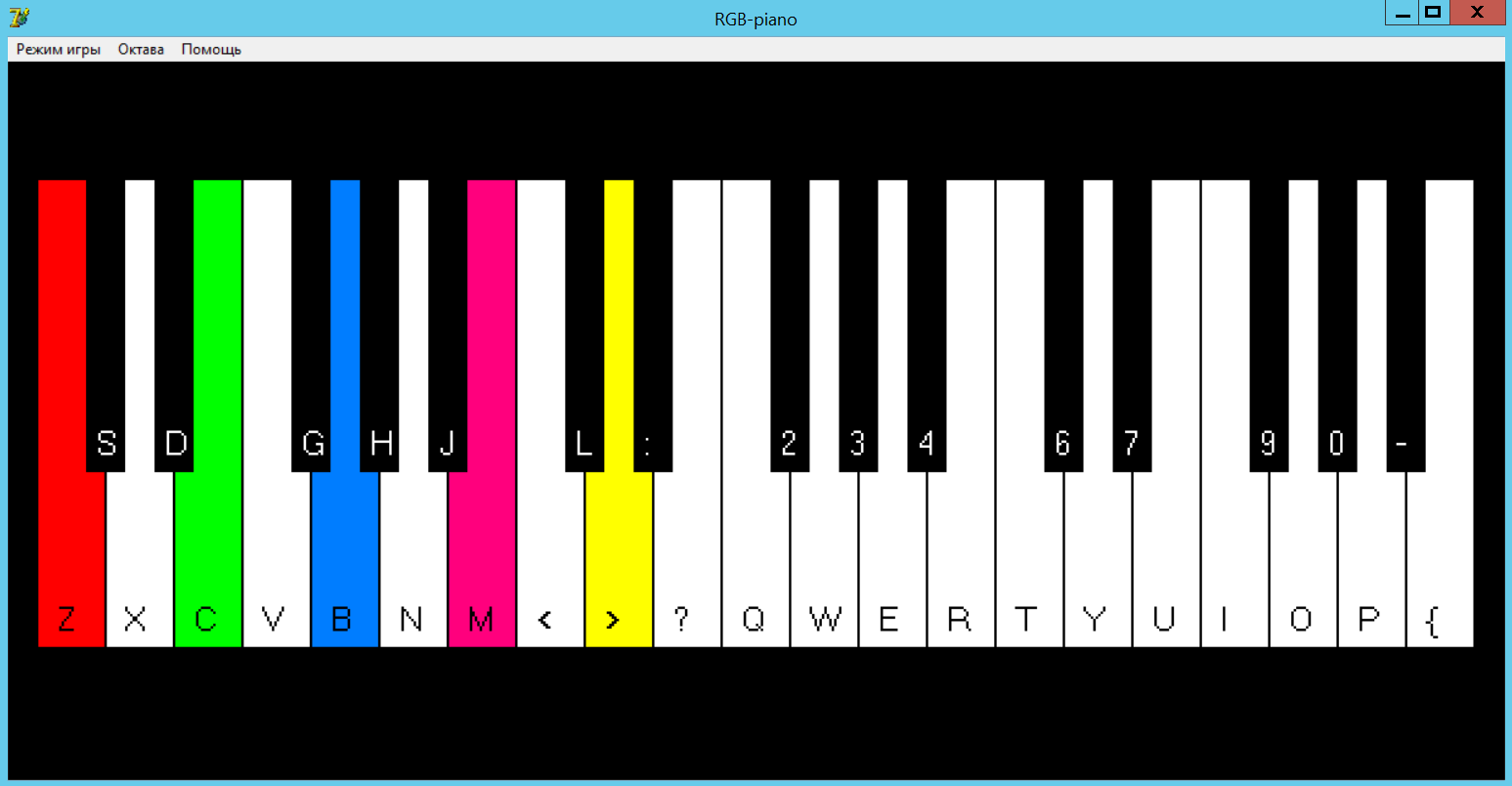
Рисунок 2.4 Окно выбора файла с цветонотами

Если файл не будет выбран выведется окошко ShowMessage с сообщением «Выберите файл!».

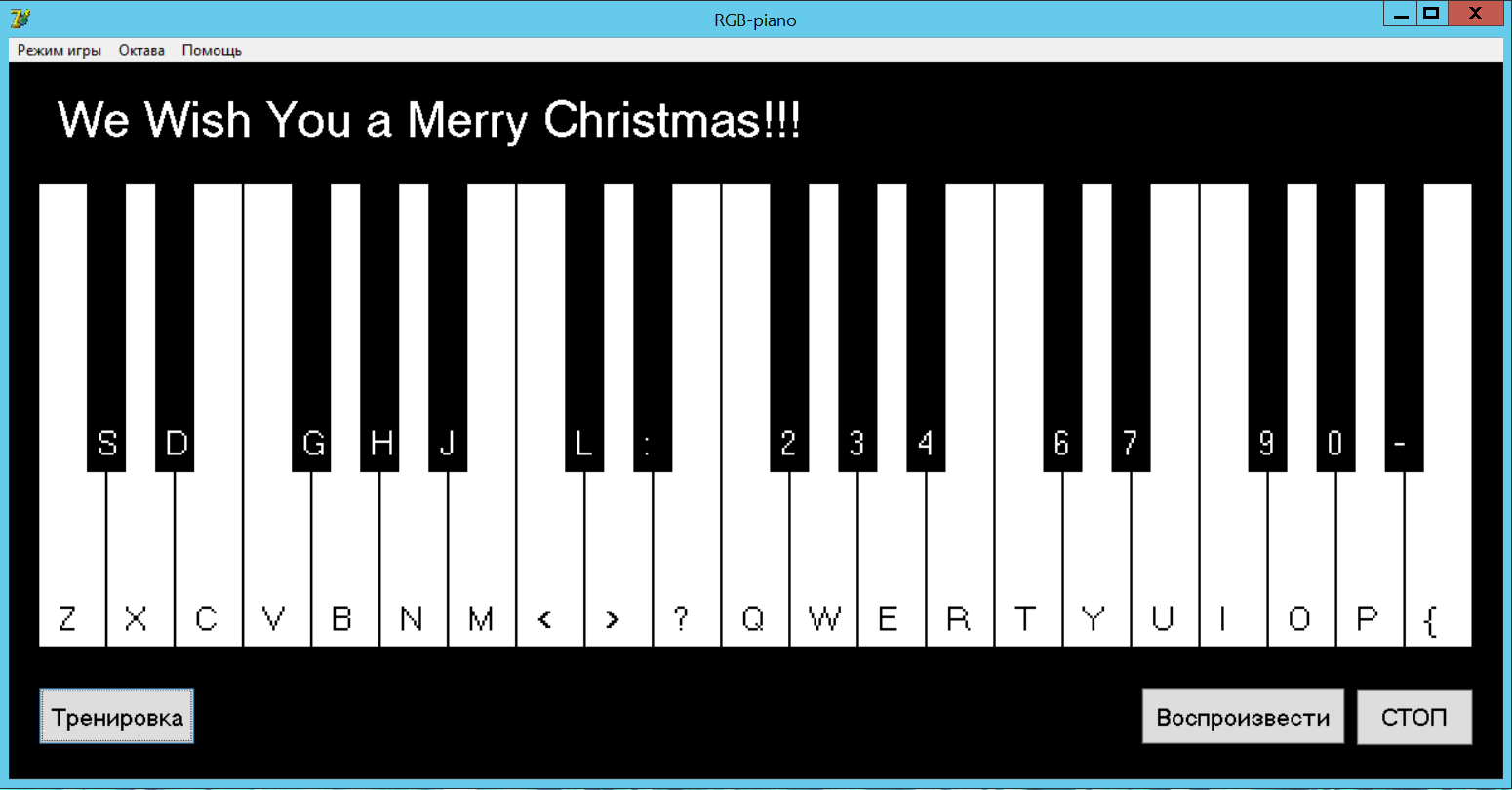
Обязательна возможность остановки автовоспроизведения. Делается это с помощью кнопки и флага. В цикле воспроизведения проверяется значение этого флага, и если он переходит в окрашенное состояние, воспроизведение останавливается и выводится «Game over».

1. **Описание применения**

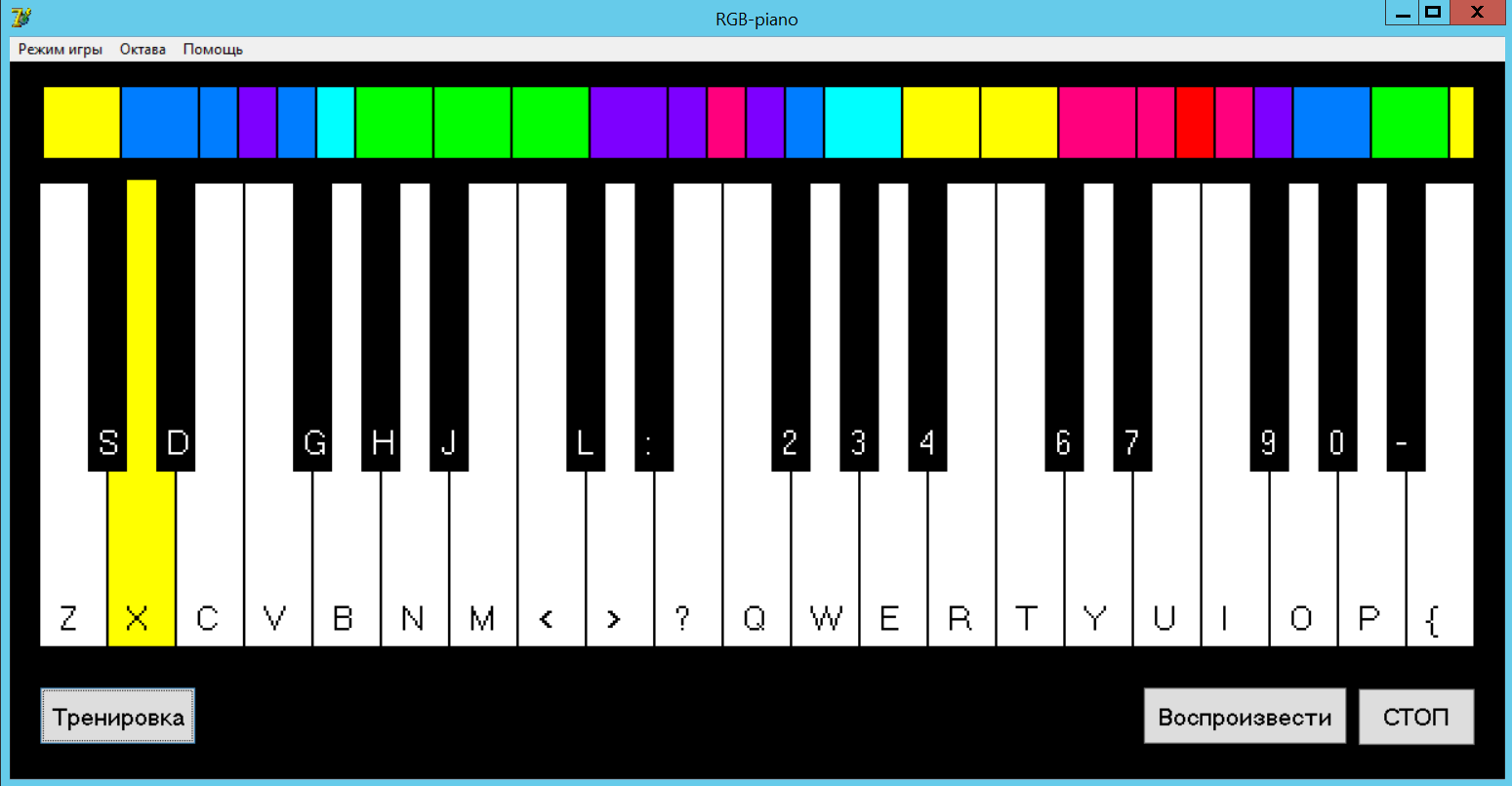
Режим «Игра» предоставляется сразу при запуске программы. Для игры достаточно нажимать на клавиатуру или мышкой по клавишам на экране. Можно зажимать несколько клавиш одновременно для получения аккордов, однако на мембранных клавиатурах недоступны некоторые сочетания клавиш, из-за чего не все аккорды доступны.

  
Рисунок 3.1 Игра в режиме «Игра»

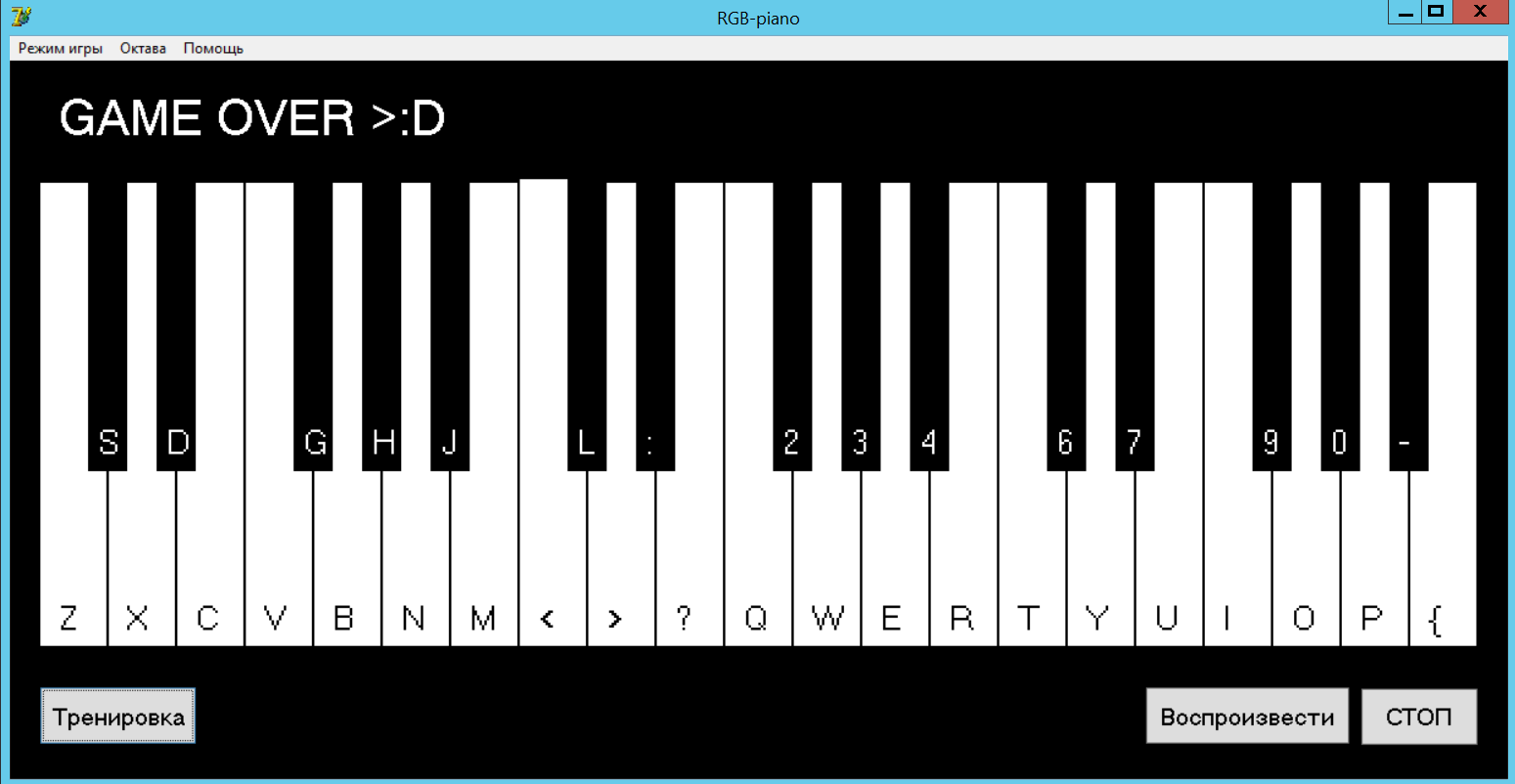
Если переключится на режим «Тренировка» пользователю предлагается выбор песни. Они представлены текстовыми файлами в папке «songs».

Рисунок 3.2 Игра в режиме тренировки

Нажав на «Тренировку», высвечиваются цветоноты, по которым пользователь будет играть. На виртуальном пианино будет подсвечиваться нужная клавиша, при нажатии по ней, цветоноты продвигаются на следующую цветоноту. Если пользователь нажимает не соответствующую цветонотам клавишу, ничего не происходит. Игра заканчивается, когда закончатся цветоноты.

 Рисунок 3.3 Процесс игры в режиме Тренировка

При нажатии на «Воспроизвести», программа играет за пользователя. Процесс «игры» выводится на экран (цветонотная дорожка «бежит», клавиши «нажимаются»).

Для остановки «Воспроизведения» можно нажать на «СТОП», тогда игра закончится.Рисунок 3.4 Конец игры в режиме «Тренировка»

Пользователю предоставлен выбор, с какой октава будет по центру его виртуального пианино. Всего в нем помещается 3 октавы, так что доступно следующие варианты:

* Малая, Первая, Вторая
* Первая, Вторая, Третья
* Вторая, Третья, Четвертая



Рисунок 3.5 Карта октав пианино

**Заключение**

Когда пользователь начнет свободно ориентироваться в нотной записи по цветам и наоборот, можно считать, что цель достигнута! Возможно после ознакомления с данной системой, кто-то окончательно перейдет на цветоноты, что поможет быстрее читать ноты (предварительно окрасив их в соответствующие цвета). В качестве обычного виртуального пианино игра полностью оправдывает себя, предоставляя возможность игры даже 5-нотных аккордов (на механической клавиатуре удастся сыграть даже в 10 нот)!

Усовершенствовать «Нотный градиент» возможно добавив варианты звуков (рояль, флейта, собачка, орган), проблема лишь в поиске wav-файлов для каждой ноты по отдельности… Также можно осуществить подключение к MIDI-клавиатуре.

Интерфейс данной программы достаточно простой. Например, дорожку цветонот можно было бы анимировать и сделать медленное затухание во время проигрывания ноты. Однако, самым достойным усовершенствованием «Нотного градиента» станет добавление новых мелодий! Изначально в базе хранится 5 песен, но этого не достаточно для настоящих любителей поиграть.

Есть надежда, что найдутся люди, которые вдохновятся идеей цветовой записи нот и сделают эту программу (мир) лучше!

**Список использованных источников**

***Основная***

1. [*Чиртик А. А., Борисок В. В., Корвель Ю. И.*](http://progbook.ru/delphi/1256-chirtik-delphi-tryuki-i-effekty.html)Delphi. Трюки и эффекты – 2010
2. *Бобровский С.И.* Delphi 7. Учебный курс. – СПб:Питер, 2003.
3. *Вирт Н.* Алгоритмы и структуры данных: Пер. с англ. – М.: Мир, 1980.

***Дополнительная***

1. *Озеров В.* Delphi. Советы программистов. –СПб:Символ-Плюс, 2004. – 976с., ил.
2. *Макконнелл С.* Совершенный код: Пер. с англ. – 2010

# Приложение А (основные функции и процедуры)

# Текст программы

{Обучающая программа «Нотный градиент»

Выполнила: учащаяся гр. 62491 Хинкель Екатерина}

//-----------------------------------------------------------------------------

// FORM CREATE

//-----------------------------------------------------------------------------

procedure TForm1.FormCreate(Sender: TObject);

var

i: Integer;

aTmp: array[0..255] of Char;

begin

InputColors(); // инициализация массива с RGB-кодами цветов

InputKeys(); // инициализация массива с кодами клавиш клавиатуры

k.o := 24; // установка октавы во «Вторую»

fplay := True; // установка режима «Игра»

for i := 1 to 36 do // окрашивание клавиш пианино в «незажатые»

(FindComponent('Shape' + inttostr(i)) as TShape).

Brush.Color := RGB(c[i].r[1], c[i].g[1], c[i].b[1]);

mciSendString(PChar('open waveaudio shareable'), nil, 0, 0); // открытие аудио-потока

GetShortPathName(PChar(extractFilePath(paramStr(0))), aTmp, Sizeof(aTmp));

k.d := StrPas(aTmp); // получение и присваивание короткого пути к проекту

end;

//-----------------------------------------------------------------------------

// KEY DOWN

//-----------------------------------------------------------------------------

procedure TForm1.FormKeyDown(Sender: TObject; var Key: Word;

Shift: TShiftState);

var

n, i, j, l: Integer;

begin

n := k.n[Key]; // зажатая нота

j := s.j; // счетчик по нотам

if fplay and k.f[Key] and (n <> 0) and ((n in note) or (mniPiano.Checked)) then

begin

// остановка играющего и загрузка нового аудио ноты

mciSendString(PChar('stop "' + k.d + 'notes\' + inttostr(n + k.o) +

'.wav"'), nil, 0, 0);

mciSendString(PChar('play "' + k.d + 'notes\' + inttostr(n + k.o) +

'.wav"'), nil, 0, 0);

k.f[Key] := False;

// установка флага зажатия клавиши (чтобы не случилось залипания)

if mniPiano.Checked then

(FindComponent('Shape' + inttostr(n)) as TShape).

Brush.Color := RGB(c[n].r[2], c[n].g[2], c[n].b[2])

else

begin

(FindComponent('Shape' + inttostr(n)) as TShape).

Brush.Color := RGB(c[n].r[1], c[n].g[1], c[n].b[1]);

// если конец цветонот

if j = s.sj then

begin

CleanForm();

Form1.Label1.Caption := 'GAME OVER >:D';

fplay := False;

for i := 1 to 36 do

(FindComponent('Shape' + inttostr(i)) as TShape).

Brush.Color := RGB(c[i].r[1], c[i].g[1], c[i].b[1]);

exit;

end;

inc(s.j);

Exclude(note, n);

//если все ноты из аккорда нажаты, переход к следующей

if note = [] then

begin

inc(s.i);

GetRectangle(s.i, s.j);

for l := 0 to s.a[s.i] - 1 do

begin

n := s.n[s.j + l];

Include(note, n);

(FindComponent('Shape' + inttostr(n)) as TShape).

Brush.Color := RGB(c[n].r[2], c[n].g[2], c[n].b[2]);

end;

end;

end;

end;

end;

//-----------------------------------------------------------------------------

// KEY UP

//-----------------------------------------------------------------------------

procedure TForm1.FormKeyUp(Sender: TObject; var Key: Word;

Shift: TShiftState);

var

n: Integer;

begin

n := k.n[Key];

// если режим «Игра», то окрашивание клавиши в «незажатую»

if n <> 0 then

begin

k.f[Key] := True;

if (mniPiano.Checked) then

(FindComponent('Shape' + inttostr(n)) as TShape).

Brush.Color := RGB(c[n].r[1], c[n].g[1], c[n].b[1]);

end;

end;

//-----------------------------------------------------------------------------

// MOUSE DOWN+UP

//-----------------------------------------------------------------------------

procedure TForm1.Shape1MouseDown(Sender: TObject; Button: TMouseButton;

Shift: TShiftState; X, Y: Integer);

begin

with (Sender as TShape) do

case Tag of

1: keybd\_event(Ord('Z'), 0, 0, 0);

2: keybd\_event(Ord('S'), 0, 0, 0);

3: keybd\_event(Ord('X'), 0, 0, 0);

4: keybd\_event(Ord('D'), 0, 0, 0);

5: keybd\_event(Ord('C'), 0, 0, 0);

6: keybd\_event(Ord('V'), 0, 0, 0);

7: keybd\_event(Ord('G'), 0, 0, 0);

8: keybd\_event(Ord('B'), 0, 0, 0);

9: keybd\_event(Ord('H'), 0, 0, 0);

10: keybd\_event(Ord('N'), 0, 0, 0);

11: keybd\_event(Ord('J'), 0, 0, 0);

12: keybd\_event(Ord('M'), 0, 0, 0);

13: keybd\_event(Ord(188), 0, 0, 0);

14: keybd\_event(Ord('L'), 0, 0, 0);

15: keybd\_event(Ord(190), 0, 0, 0);

16: keybd\_event(Ord(186), 0, 0, 0);

17: keybd\_event(Ord(191), 0, 0, 0);

18: keybd\_event(Ord('Q'), 0, 0, 0);

19: keybd\_event(Ord('2'), 0, 0, 0);

20: keybd\_event(Ord('W'), 0, 0, 0);

21: keybd\_event(Ord('3'), 0, 0, 0);

22: keybd\_event(Ord('E'), 0, 0, 0);

23: keybd\_event(Ord('4'), 0, 0, 0);

24: keybd\_event(Ord('R'), 0, 0, 0);

25: keybd\_event(Ord('T'), 0, 0, 0);

26: keybd\_event(Ord('6'), 0, 0, 0);

27: keybd\_event(Ord('Y'), 0, 0, 0);

28: keybd\_event(Ord('7'), 0, 0, 0);

29: keybd\_event(Ord('U'), 0, 0, 0);

30: keybd\_event(Ord('I'), 0, 0, 0);

31: keybd\_event(Ord('9'), 0, 0, 0);

32: keybd\_event(Ord('O'), 0, 0, 0);

33: keybd\_event(Ord('0'), 0, 0, 0);

34: keybd\_event(Ord('P'), 0, 0, 0);

35: keybd\_event(Ord(189), 0, 0, 0);

36: keybd\_event(Ord(219), 0, 0, 0);

end;

end;

procedure TForm1.Shape1MouseUp(Sender: TObject; Button: TMouseButton;

Shift: TShiftState; X, Y: Integer);

begin

with (Sender as TShape) do

case Tag of

1: keybd\_event(Ord('Z'), 0, KEYEVENTF\_KEYUP, 0);

2: keybd\_event(Ord('S'), 0, KEYEVENTF\_KEYUP, 0);

3: keybd\_event(Ord('X'), 0, KEYEVENTF\_KEYUP, 0);

4: keybd\_event(Ord('D'), 0, KEYEVENTF\_KEYUP, 0);

5: keybd\_event(Ord('C'), 0, KEYEVENTF\_KEYUP, 0);

6: keybd\_event(Ord('V'), 0, KEYEVENTF\_KEYUP, 0);

7: keybd\_event(Ord('G'), 0, KEYEVENTF\_KEYUP, 0);

8: keybd\_event(Ord('B'), 0, KEYEVENTF\_KEYUP, 0);

9: keybd\_event(Ord('H'), 0, KEYEVENTF\_KEYUP, 0);

10: keybd\_event(Ord('N'), 0, KEYEVENTF\_KEYUP, 0);

11: keybd\_event(Ord('J'), 0, KEYEVENTF\_KEYUP, 0);

12: keybd\_event(Ord('M'), 0, KEYEVENTF\_KEYUP, 0);

13: keybd\_event(Ord(188), 0, KEYEVENTF\_KEYUP, 0);

14: keybd\_event(Ord('L'), 0, KEYEVENTF\_KEYUP, 0);

15: keybd\_event(Ord(190), 0, KEYEVENTF\_KEYUP, 0);

16: keybd\_event(Ord(186), 0, KEYEVENTF\_KEYUP, 0);

17: keybd\_event(Ord(191), 0, KEYEVENTF\_KEYUP, 0);

18: keybd\_event(Ord('Q'), 0, KEYEVENTF\_KEYUP, 0);

19: keybd\_event(Ord('2'), 0, KEYEVENTF\_KEYUP, 0);

20: keybd\_event(Ord('W'), 0, KEYEVENTF\_KEYUP, 0);

21: keybd\_event(Ord('3'), 0, KEYEVENTF\_KEYUP, 0);

22: keybd\_event(Ord('E'), 0, KEYEVENTF\_KEYUP, 0);

23: keybd\_event(Ord('4'), 0, KEYEVENTF\_KEYUP, 0);

24: keybd\_event(Ord('R'), 0, KEYEVENTF\_KEYUP, 0);

25: keybd\_event(Ord('T'), 0, KEYEVENTF\_KEYUP, 0);

26: keybd\_event(Ord('6'), 0, KEYEVENTF\_KEYUP, 0);

27: keybd\_event(Ord('Y'), 0, KEYEVENTF\_KEYUP, 0);

28: keybd\_event(Ord('7'), 0, KEYEVENTF\_KEYUP, 0);

29: keybd\_event(Ord('U'), 0, KEYEVENTF\_KEYUP, 0);

30: keybd\_event(Ord('I'), 0, KEYEVENTF\_KEYUP, 0);

31: keybd\_event(Ord('9'), 0, KEYEVENTF\_KEYUP, 0);

32: keybd\_event(Ord('O'), 0, KEYEVENTF\_KEYUP, 0);

33: keybd\_event(Ord('0'), 0, KEYEVENTF\_KEYUP, 0);

34: keybd\_event(Ord('P'), 0, KEYEVENTF\_KEYUP, 0);

35: keybd\_event(Ord(189), 0, KEYEVENTF\_KEYUP, 0);

36: keybd\_event(Ord(219), 0, KEYEVENTF\_KEYUP, 0);

end;

end;

//------------------------------------------------------------------------------------------------------------------------+

// АВТОВОСПРОИЗВЕДЕНИЕ //////////////////////////////////////////////////////////////////////////////////////////////////

//------------------------------------------------------------------------------------------------------------------------+

procedure TForm1.btnCompClick(Sender: TObject);

var

i, j, l, n: integer;

begin

mniTrain.Checked := True;

fplay := False;

fstop := False;

j := 1;

GetRectangle(1, 1);

for i := s.i to s.si do // проход по каждой ноте в цветонотах

begin

for l := 0 to (s.a[i] – 1) do // одновременное проигрывание всех нот в аккорде

begin

n := s.n[j + l];

mciSendString(PChar('stop "' + k.d + 'notes\' + inttostr(n + k.o) +

'.wav"'), nil, 0, 0);

mciSendString(PChar('play "' + k.d + 'notes\' + inttostr(n + k.o) +

'.wav"'), nil, 0, 0);

(FindComponent('Shape' + inttostr(n)) as TShape).

Brush.Color := RGB(c[n].r[2], c[n].g[2], c[n].b[2]);

end;

Application.ProcessMessages;

for l := 0 to (s.a[i] - 1) do // проприсовка аккорда

begin

n := s.n[j + l];

(FindComponent('Shape' + inttostr(n)) as TShape).

Brush.Color := RGB(c[n].r[1], c[n].g[1], c[n].b[1]);

end;

j := j + s.a[i];

if i = s.si then

Break;

GetRectangle(i + 1, j); // отрисовка следующей ноты

Sleep(1200 div s.t[i]); // ожидание всей длительности ноты

Application.ProcessMessages; // вывести на экран, что нота «сыграна»

if fstop then

Break;

end;

CleanForm();

Form1.Label1.Caption := 'GAME OVER >:D';

end;

//-------------------------------------------------------------------------------------------------------------------------

// ~~ ~~ ~~ ~~ ~~ ~~ ~~ ~~ ~~ ~~ ~~ ЦВЕТОНОТЫ ~~ ~~ ~~ ~~ ~~ ~~ ~~ ~~ ~~ ~~ ~~ ~~

//-------------------------------------------------------------------------------------------------------------------------

procedure GetRectangle(i, j: integer);

var

x1, x2, y1, y2, istep, jstep, l, n: Integer;

begin

CleanForm(); //очистка формы от старых цветонот

istep := 256; // шаг для отрисовки одной целой ноты

x2 := 27;

y1 := 20;

y2 := 80;

while (x2 < 1200) and (i <= s.si) do // пока не закончится место на экране или сами цветоноты

begin

x1 := x2;

Form1.Canvas.Brush.Style := bsSolid;

jstep := (istep div s.t[i]) div s.a[i];

for l := 0 to (s.a[i] - 1) do // прорисовка всех нот в аккорде

begin

x2 := x2 + jstep;

if x2 > 1200 then

x2 := 1200;

n := s.n[j + l]; // количество сыгранных нот

Form1.Canvas.Pen.Color := RGB(c[n].r[2], c[n].g[2], c[n].b[2]);

Form1.Canvas.Brush.Color := RGB(c[n].r[2], c[n].g[2], c[n].b[2]);

Form1.Canvas.Rectangle(x1 + (jstep \* l), y1, x2, y2);

end;

j := j + s.a[i]; //переход на следующий аккорд (j)

Form1.Canvas.Brush.Style := bsClear;

Form1.Canvas.Pen.Color := clBlack;

Form1.Canvas.Rectangle(x1, y1, x2, y2);

inc(i); //переход на следующий аккорд (i)

end;

end;

//-----------------------------------------------------------------------------------------------------------------------

// FORM\_DESTROY

//-----------------------------------------------------------------------------------------------------------------------

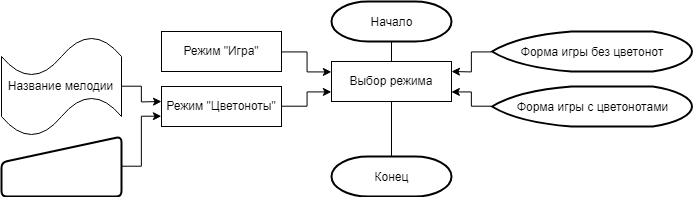
procedure TForm1.FormDestroy(Sender: TObject);

begin

mciSendString(PChar('close waveaudio'), nil, 0, 0); // закрытия аудио-потока

end;

# Приложение Б (обязательное) Схема работы системы

****

Текстовый файл   
с цветонотами

# Приложение В (справочное) Схема алгоритма Автовоспроизведения

Цикл l  
Увеличить l на 1

# 

Цикл l  
Увеличить l на 1

Цикл i  
Увеличить i на 1