Тетрис

1. Размер поля – 8\*8
2. Размеры игрового поля 20\*20 символов
3. Строительный материал: ║,╔╗,╤, ╬,

Ход игры:

Фигуры выпадают в случайном порядке, следующая на очереди фигура отображается в верхнем правом угру.После заполнения строки она пропадает. В процессе падения возможно повернуть фигуру вокруг своей оси. Игра заканчивается как только самая верхняя точка строения касается верхнего края игрового поля.

Клавиши управления:

Enter – поставить фигуру на место, единовременно с вызовом следующей фигуры.

Ctrl – повернуть фигуру вокруг оси

Курсор ← → – сдвиг фигуры влево, право

Пример готового варианта программы (рис 1.)

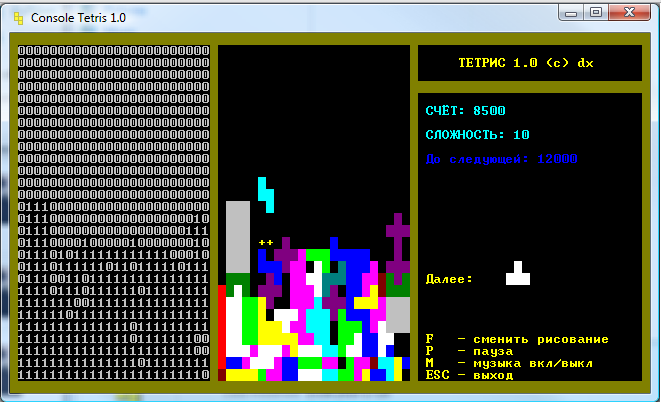


Рис.1

Архитектура

Программа представляет несколько классов

Класс Тетрис (Main)

Основной класс, который управляет другими классами и отвечает за подсчет очков (переменная типа int)

Методы:

Создается матрица 20\*20 –> const (игровое поле), с нулевыми элементами.

* int[,] CreateEmptyField(byte vertSize, byte horizontSize) ?
* GameField CreateEmptyField(byte vertSize, byte horizontSize) ?

Класс поля struct GameField

Игровое поле представляет собой матрицу 20\*20 символов. Пустота «0» заполненное место «1» очевидно должен быть enum **Color**

Методы для поля: удаление массивов «1» из матрицы нулей и сдвиг вниз на одну строку

Класс Фигур struct Figure

Создается enum фигур (вид фигуры) из которого генерируется выпадение фигур в рандомном порядке. + цвет + координаты

типы данных

Методы

Метод генерирования фигурки (Figure CreateRandomFigure()), метод передвижения влево/право. Метод падения фигурки, будет проверка на свободное место для фигурки, если она падает на самую верхнюю лежащую фигурку, и места для нее нет – это означает конец игры.

Класс для работы с графикой в консоли

Цвет, размер поля, рамки, вспомогательный текст

Модули - ? (то, что реализуется через class)