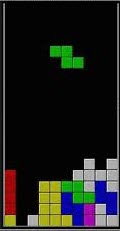
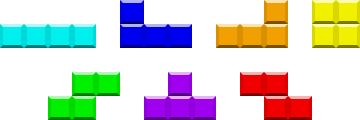
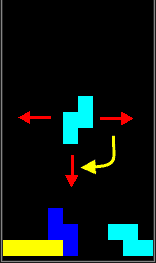
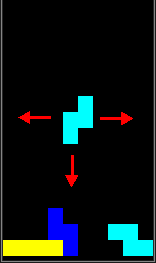
**Опис гри "Тетріс"**



Випадкові фігурки тетраміно падають зверху в прямокутний стакан (матрицю) визначеної ширини і висоти (наприклад, 10х20 клітинок). Варіанти фігур описані нижче:   


Під час польоту гравець може переміщувати фігуру вліво, вправо по горизонталі, а також вниз, прискорюючи падіння (за допомогою клавіш управління курсором). На 12 балів реалізувати також поворот фігури.   


Кожна фігурка летить зверху, поки не наштовхнеться на іншу фігурку або не впаде на дно матриці. Якщо при цьому заповниться горизонтальний ряд, тоді він пропадає і все, що було його вище, опускається на 1 клітинку. Гра закінчується, коли нова фігурка не може поміститись в матрицю.

Гравець отримує очки за кожну фігурку, тому його задача — заповнювати ряди, не заповнюючи саму матрицю (стакан) як можна довше, щоб таким чином отримати як можна більше очків. Сам підрахунок очків в різних версіях «Тетріса» різна. Очки можуть нараховуватись за прибрані лінії, за скинуті фігурки, за перехід на нову швидкість і т.д.

**Орієнтовна структура класів:**

1. Point, який описує окрему точку матриці
2. Matrix (матриця), яка має певний розмір, а також управляє фігурами, підраховує очки, перемальовує саме поле тетріса тощо. Він також може містити орієнтовно наступні методи:
   * Конструктор, який по бажанню може містити темп гри, ініціалізує очки нулем тощо.
   * Draw або Fill, який дозволяє розташовувати фігуру на полі матриці. При першому виклику цього метода, буде відображатись весь тетріс.
   * Метод генерування фігури.
   * Move, який дозволяє переміщувати фігуру вліво/вправо.
   * Rotate - метод повороту фігури.
   * Метод падіння фігури
   * Метод перевірки на заповнення горизонтального ряду в тих місцях, де розміщується щойно впавша фігура.
3. Shape (фігура), який дозволяє згенерувати рандомну фігуру. Можна також створити окремі класи для різних типів фігур.

Передбачити також можливість збереження гри в файл та завантаження її з файла.