Карточная игра "Дурак"

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

Игровой процесс:

В игре используется колода из 36 карт и участвуют от двух до шести игроков. Каждому раздаётся по 6 карт, следующая карта открывается и её масть устанавливает козырь для данной игры, и остальная колода кладётся сверху, так, чтобы козырная карта была всем видна. Цель игры — избавиться от всех карт. Последний игрок, не избавившийся от карт, остаётся в «дураках». В первой сдаче первым ходит игрок с младшим козырем; в дальнейших сдачах ходят либо «под дурака», то есть игрок справа от «дурака», либо «из-под дурака», то есть игрок слева, по договору. Ход делается всегда налево, и состоит из выкладывания одной или больше карт, и попытки игрока, под которого ходят, их покрыть либо старшей картой той же масти, либо картой козырной масти, если кроющаяся карта сама не является козырной — в таком случае её можно покрыть только старшим козырем. Если игрок, под которого ходят, отбился, то он ходит следующим, но сначала все игроки добирают до 6 карт, начиная с игрока, который ходил первым, и кончая игроком, который отбился. Если он не смог отбиться, то он принимает все карты, и следующим ходит игрок слева.

При участии четырёх или шести игроков возможна игра 2х2 или 3х3, когда игроки, сидящие через одного, являются партнёрами и не ходят друг под друга.

Ходить можно любым количеством карт одного достоинства. Игрок, под которого ходят, может крыть карты, а может принять. Ходящий игрок имеет право «подкинуть» карты, совпадающие по достоинству с любой картой на столе, включая те, которыми ходили, и те, которыми крыли. Если в игре участвуют более двух игроков, то игрок, который ходил, имеет право первого слова, но когда он кончил ходить, то остальные игроки имеют право подкинуть свои карты по тем же правилам. Нельзя подкидывать больше карт, чем остаётся у отбивающегося игрока . Также нельзя подкидывать всего больше шести карт, даже если у игрока на руках больше карт. При первом «отбое» даже нельзя подкидывать отбивающему игроку всего больше, чем 5 карт.Клавиши управления:

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

Ctrl – повернуть фигуру вокруг оси

Курсор ← → – сдвиг фигуры влево, право

Разработайте программу, которая позволяет имитировать игру человека с компьютером и человека с человеком в подкидного дурака.

Функциональные требования:

Программа должна быть реализована в консоли. По желанию студента возможно использование визуальных средств среды разработки.

Программа должна реализовывать эффективный искусственный интеллект компьютера:

Искусственный интеллект должен учитывать сброшенные карты.

Искусственный интеллект должен уметь оценивать ценность карты и выбирать оптимальный ход. Например, не ходить в начале игры с козыря, крыть картой, которой нет у противника, беречь парные карты, крыть наименьшей картой.

enum Suit

{

HEARTS,

SPADES,

CLUBS,

DIAMONDS

}

struct Card

{

int \_cardValue;

bool \_isTrump;

Suit \_cardSuite;

}

// Колода карт

struct CardDeck

{

Card[] \_cards; // 36 or 50 cards

int \_countCardsLeft;

// Methods

void GiveAwayCard(Player pl);

void Shuffle();

}

struct Player

{

Card[] \_cards;

bool \_isPlaying; // Ходит ли игрок?

void Turn();

}

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)