В дополнение к диаграмме классов проекта, которая расположена в файле ProjectClassDiagramm.cd решения, дополнительно привожу краткое описание назначения каждого из использованного класса. В рамках работы использовано 2 паттерна: Strategy и MVC. Первый паттерн позволяет реализовать возможность изменения логики игры или отрисовки, второй –взаимодействие UI и игровой логики. Также в стандартном варианте исполнения игровой логики открытие пустых клеток до клеток с бомбами-соседями идет согласно рекурсивному алгоритму заливки с просмотром окрестности фон Неймана первого порядка. Ниже приведены краткие данные о классах, используемых в решении.

Классы с данными о ячейках

Класс LogicalCell – класс, используемый для хранения информации о ячейке игрового поля, которая используется в логике.

Класс LayoutCell – класс, используемый для хранения информации о отрисованной ячейке игрового поля, которая используется в CustomControll.

Классы игровой логики

Класс GameField – класс, используемый для хранения информации о игровом поле (массив логических ячеек) и методов его обработки. Методы обработки задаются через интерфейс IStrategy и могут быть переопределены при перегрузке метода Define Strategy в наследнике.

Интерфейс IGameMechanics – Интерфейс, определяющий методы, которые должны реализовать классы игровой логики.

Класс StandartMechanic – класс, реализующий стандартные методы игровой логики.

Класс-посредник

Интерфейс IController – интерфейс, определяющий основные методы, которые должен содержать реализуемый контроллер.

Класс Controller – класс-посредник между игровой логикой и компонентом отрисовки.

Класс для обмена данными между игровой логикой и компонентом отрисовки.

Класс OpenResult – класс, возвращаемый игровой логикой в ответ на запрос обработки игрового действия компонентом отрисовки. Содержит в себе информацию о ситуации поля после игрового действия (победа, поражение, ничего), о измененном виде игрового поля, дополнительную информацию.

Классы компонента отрисовки.

Класс CustomControll – класс, реализующий UI-компонент для работы с пользователем. Обрабатывает действия пользователя и рисует поля в качестве ответа на них. Включает в себя класс с дополнительной информацией и информацией для отрисовки, класс-отрисовщик и класс – контроллер. Отрисовщик может быть изменен посредством переопределения функции DefinePainer в наследнике.

Класс CustomControllViewInfo – класс, хранящий дополнительную информацию о компоненте (число бомб, размер поля и т.п.), информацию для отрисовки (массив клеток, рисуемых компонентом и прочее). Экземпляр класса используется отрисовщиком в качестве источника информации для отрисовки.

Интерфейс ICustomControllPainter – интерфейс, определяющий основные методы, которые должен содержать реализуемый класс - отрисовщик.

Класс StandartCustomControllPainter – класс, реализующий стандартную отрисовку игрового поля.

Класс Positions – класс, применяемый для хранения информации о позиции на UI-компоненте.