Классы:

**Lose -** создает форму при поражении \ победе. Используются БД, стандартные диалоги, QLabel, QIcon, QPushButton, uic.

Функции:

1. \_\_init\_\_ - выводит лейбл с большим шрифтом и картинкой поздравления о победе \ оповещения о поражении. Выводит в правом верхнем углу чемпиона режима, если таковой имеется.
2. defnazad1 - возвращает пользователя назад к выбору режима. Имеет в названии цифру 1 т,к повторяет действие функции defnazad, находящейся в другом классе
3. restart1 - используя глобальные переменные flag и val, которые дают информацию о выбранном ранее режиме, перезапускает игру. Глобальная переменная a сделана такой, чтобы форма не сворачивалась.
4. dob - работа с БД. Если пользователь зарегистрирован в БД, то просит ввести пароль после нажатия на кнопку «Добавить». Если ник свободен, просит придумать пароль и заносит информацию в БД. Вторая БД хранит в себе информацию: имя, время, уровень.
5. NewUser - выводит стандартный диалог после нажатия на кнопку «Добавить», если пользователя нет в БД, просит придумать пароль
6. UppUser - выводит стандартный диалог после нажатия на кнопку «Добавить», если пользователь есть в БД. Просит ввести пароль, чтобы обновить данные

**FAQInfo -** создаёт форму при нажатии на кнопку «FAQ» в форме выбора режима. Считывает информацию с файла FAQ.txt и выводит его. Содержит информацию об игре.

Функции:

1. \_\_init\_\_ (initUI) - Чтение информации с файла .txt и вывод её на экран.

**Modes -** создаёт форму выбора режима. Используются QRadioButton, QSlider, QPushButton, QLabel.

Функции:

1. \_\_init\_\_(initUI) - Коннект кнопок, границы QSlider’а, шаг QSlider’а
2. changeValue - работа со слайдером. Вывод значения слайдера на лейбл
3. Start - Создание формы игры в зависимости от того, использовался слайдер или нет или какую QRadioButton выбрал пользователь.
4. defFAQ - создает объект класса FAQInfo, выводит информацию об игре

**MyButton -** Переработанный класс кнопки для того, чтобы можно было ставить и убирать флажки нажатием ПКМ. За основу был взят код с <https://ru.stackoverflow.com/questions/748835>

Функции:

1. \_\_init\_\_ создает список со значениями 0 (флажок не поставлен). Используется для просмотра информации о том, установлен ли флажок
2. mousePressEvent - обработка ПКМ. Если координата кнопки в списке имеет значение 0, меняет значение на 1 и устанавливает флажок. Иначе убирает флажок и меняет значение на 0.

**Sapper -** класс игры. С помощью цикла for и глобальной переменной leng расставляет кнопки, присваивает им индивидуальное имя объекта(NameObject) в формате ‘n’ + coord\_y + ‘x’ + coord\_x и запускает отсчёт времени.

Функции:

1. click - при клике в первый раз расставляет бомбы, в последующие разы вызывает функцию xod
2. xod - функция хода игры. Подсчитывает кол-во бомб рядом и выводит на кнопку, если бомб рядом нет, красит в зелёный цвет. Также уменьшает кол-во свободных клеток на 1 (левый нижний угол). При поражении \ кол-ве оставшихся свободных клеток = 0 открывает форму Lose. Открытые кнопки блокируются
3. Returne - возвращает пользователя к выбору режима
4. Restart - начинает игру заново с теми же параметрами
5. Deftimer - отсчёт времени путём использования библиотеки threading

В будущем планирую доделать проект: Сделать так, чтобы соседние зелёные кнопки рядом с уже открытой зелёной открывались сами. Также планируется добавить возможность выбора плотности бомб. Сейчас стоит плотность 7. (кол-во клеток // 7)