**Пояснительная записка к проекту**

В начале файла **main.py** я импортирую нужные мне библиотеки (классы из PyQt, NumPy) и алгоритм для просчета следующего поколения для игры “Жизнь” Джона Конвея.

Cоздаю класс Main, наследующегося от класса QMainWindow, и в init’e наследую все его свойства, загружаю UI из уже созданного мной **main\_ui.ui** в Qt Designer, вызываю функцию initUI(). В ней я задаю иконку приложения с помощью QIcon, задаю стандартные переменные и массив, которые понадобятся далее, чтобы не вызвалась ошибка из-за отсутствия этих переменных в этих функциях.

**Генерация сетки.**

Cвязываю поля для ввода ширины **le\_width** и высоты **le\_height** сетки с обработчиком ошибок ввода для невозможности создания сетки с некорректно введенными данными. Привязываю событие нажатия кнопки **gen\_btn** к функции paint. В ней я проверяю не пустые ли поля с ширинойи высотой пустыми, если же они пустые, то в поле **err\_lbl\_gen** выводится соответствующий текст, и кнопка выключается, тоже самое происходит в обработчике ошибок ввода. Но если же поля для ввода оказались не пустыми, то я проверяю не генерировалась ли сетка прежде или является ли сейчас сетка пустой, если что то из этого нет, то создаю массив сетки, задаю значения масштаба и расположения сетки, включаю остальную часть **menu\_box**, т.к. в самом начале он был выключен, перерисовываю сетку, если же сетка была сгенирирована ранее или сетка была не пустая и если в QMessageBox **m\_box** пользователь согласился перерисовать сетку, то производятся те же действия, что и в первом случае, за исключением того что если была включена автоматическая смена поколений, то она выключалась. В **paint\_event** создается объект QPainter и с помощью его в функции **draw** рисуется сетка. Сначала задается цвет кисточки, и если был нажат соответсвующий флажок **cb\_showgrid** то рисуется сетка, затем идет отрисовка живых клеток.

**Основная часть.**

Событие смены значения флажка **cb\_showgrid**, который показывает/скрывает сетку, с функцией перерисовки. Связываю событие нажатия кнопки **clear\_btn** с функцией **clear\_grid**, которая очищает сетку.

**Автоматическая смена поколений**

Связываю событие нажатия кнопки **next\_step\_btn** с функцией **next\_step**, в которой обновляем сетку в соответсвии с правилами игры “Жизнь” Джона Конвея с помощью функции из импортируемого файла.

Выключаю кнопку **stop\_btn**, и связываю событие нажатия ее и кнопки **start\_btn** с функцией **animation**, также связываю событие изменения значения слайдера **speed\_slider** с этой же функцией. В это функции если отправителем является кнопка старта, то создаем таймер, который каждый свой тик будет обрабатывать следующее поколение, выключаем кнопки старта и следующего поколения, включаем кнопку остановки, если отправитель слайдер, то изменяем время каждого тика в соответствии со значением слайдера, если же отправитель кнопка остановки, то останавливаем таймер выключаем кнопки старта и следующего поколения, включаем кнопку остановки, включаем кнопки старта и следующего поколения, выключаем кнопку остановки.

**Рандомизация сетки**

Cвязываю поле для ввода процента появления живой клетки **le\_rnd** с обработчиком ошибок ввода **rnd\_err**. Связываю событие нажатия кнопки **rnd\_btn** с функцией **rnd\_grid**, в которой если **le\_rnd** не пустой, то если сетка пустая или пользователь согласился в **m\_box**, то сетка рандомизируется,если же **le\_rnd** оказался пустой, то производятся действия аналогичные с ситуацией с **le\_width** и **le\_height**.

**HeatMap**

Выключаю поле для ввода **le\_aliveHeat** и кнопку **heat\_btn** до того, как соответсвующий флажок **cb\_heat** не будет нажат. Связываю событие изменения текста в le\_aliveHeat с обработчиком ошибок ввода. Привязываю событие нажатия кнопки с функцией **heat**, в которой если число было введено корректно, то изменяется соответствующая переменная **heatAlive**.

**Горячие клавиши**

Привязываю соответсвующие поля с классом **hotkey**.

**График**

Создаю переменную **generation**, которая хранит в себе номер текущего поколения. Создаю массив **population**, который хранит в себе число живых клеток каждого поколения. На основе этой переменной и массива создаю график **graph**.

**Обработка мыши и клавиатуры**

**Колесо мыши.** Если курсор находится не в пределах **menu\_box**, то **curr\_grid\_pos** и **scale** изменяются в соответствии с поворотом колесика мыши. В результате сетка масштабируется в соответствии нахождения курсора.

**Нажатие кнопок мыши.** Если была нажата правая кнопка мыши, то **rb\_x** и **rb\_y**, которые понадобятся позже, если же была нажата левая кнопка мыши, то высчитывается на какой клетке сетки находился курсор и на этом месте появляется или убирается живая клетка.

**Движение мыши.** Если в данный момент зажата правая кнопка мыши, то изменяется положение сетки с помощью предыдущего положения курсора, записанного в **rb\_x** **и rb\_y**.

**Клавиатура.** Если нажата клавиша **Home**, то сетка центрируется, если **F11**, то сетка центрируется и программа переводится в режим полного экрана или наоборот выходит из него если она был в нем.