Язык: С++

1. Задача: Создать Win32 оконное приложение.

1.1. При старте программы окно разворачивается на полный экран. Поддержка изменения размера окна пользователем не требуется.

1.2. В окне должно быть нарисовано прямоугольное поле, состоящее из квадратных клеток размером 32х32 пикселя.

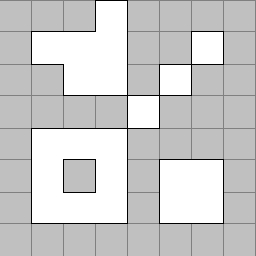
1.3. Размеры поля - на сколько хватит развёрнутого окна, вычислять из размера окна.

1.4. Фон поля - серый, с нарисованной более тёмным тоном сеткой в 1 пиксель толщиной, чтобы пользователь представлял себе, где какая клетка находится.

1.5. При клике мышкой на клетку она меняет свой статус с пустой на занятую и наоборот. У занятой клетки цвет заливки меняется на белый и появляется чёрная окантовка (тоже 1 пиксель толщиной). При этом окантовка объединяется с соседними занятыми клетками, т.е. если стоят две занятые рядом, имея общую сторону, то окантовка у них одна.

1.6. По нажатию Ctrl-Z: отменять последнее изменение поля, вплоть до начального состояния программы в момент запуска.

Пример:



2. Требования:

2.1. Не применять «трёхэтажные» шаблоны, лямбды и boost

2.2. Для работы с окном и рисованием в него можно использовать только функции WinAPI и GDI.

2.3. Все параметры в программе должны читаться из файла конфигурации с фиксированным именем: размеры клеток, цвета клеток, фона, линий сетки и окантовки, скорость анимации и прочее. Чем больше, тем лучше. Формат файла такой, чтобы пользователь смог его впервые открыв, прочитать, понять, что за что отвечает и поменять.

3. Бонусные очки:

3.1. За анимацию добавления/удаления клетки. При клике на клетку отображение клетки на экране не меняется моментально, а плавно в течении некоторого времени.

Если в процессе анимации пользователь снова кликнул на клетку, то анимация должна поменяться в обратную сторону с момента прерывания. Т.е. если анимация шла с 0% до 100% и её прервали на 75% то она должна пойти с 75% до 0%.

Какая именно анимация - на усмотрение разработчика.

Тут мы хотим увидеть работу с фоновыми событиями, которые могут быть в любой момент прерваны/отменены. (Фоновое событие не обязательно должно работать в отдельном потоке)

4. На что будет обращено внимание:

4.1. Как будет построена архитектура приложения, как поделены сущности на классы и как между ними построено взаимодействие.

4.2. Какие классы-кирпичики будут созданы для решения задачи, и как из них будет построена основная логика программы.

5. Примечание: мелкие баги допустимы.