**Тестовое задание.**

1. Проект создан на базе стандартного шаблона Win32 приложения, предоставляемого студией.
2. Компилятор msvc2015r2, сборка приложения с флагом /MT, исключающим зависимость от студийных dll.
3. Код доступен на гите: <https://github.com/goldstar2154/Cells>

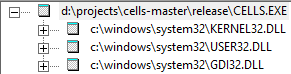
<https://github.com/goldstar2154/Cells/releases/tag/v1.0>

1. Примерное время на выполнение ~12h.

**Структура.**

1. Cells.exe – исполняемый файл, 345 600 байт.

Зависимости:



1. config.ini – файл с настройками, 191 байт.

Содержимое:

|  |  |
| --- | --- |
| [Animation] | Секция параметров, отвечающих за работу рендера |
| Delay=10 | Определяет скорость работы анимации, меньше – быстрее.  Не рекомендуется использовать значение 0.  По умолчанию = 10. |
| [Border] | Секция параметров, отвечающая за параметры линий окантовки |
| Width=2 | Ширина линии окантовки клеток.  Если равно 0, окантовки не будет.  По умолчанию = 2. |
| [Cell] | Секция параметров, отвечающая за параметры клетки |
| Width | Ширина клетки в пикселах.  По умолчанию = 25. |
| Height | Высота клетки в пикселах.  По умолчанию = 25. |
| InactiveCellFill | Цвет заливки неактивной клетки.  По умолчанию = 190, 190, 190 |
| InactiveCellBorder | Цвет заливки окантовки неактивной клетки.  По умолчанию = 120, 120, 120 |
| ActiveCellFill | Цвет заливки активной клетки.  По умолчанию = 255, 255, 255 |
| ActiveCellBorder | Цвет заливки окантовки активной клетки.  По умолчанию = 0, 0, 0 |

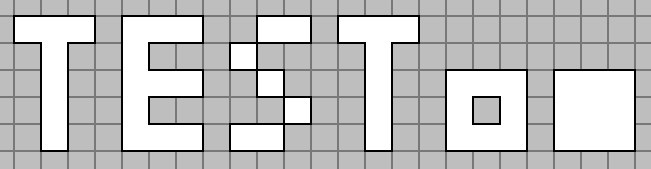
В случае отсутствия файла приложение использует значения по умолчанию.

**Функционал**

1. Согласно заданию
   1. В коде используется 1 лямбда и 2 шаблона.
   2. Из стороннего функционала используются только:

<thread>, <stack>, <mutex>.

1. Скрин



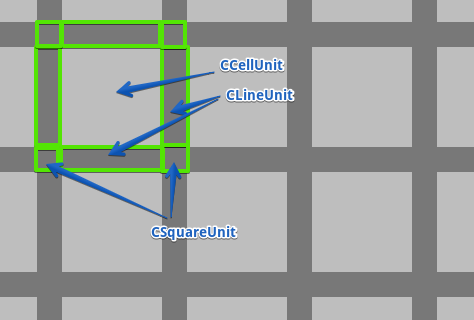
1. Основные классы:
   1. CSettingsManager

Осуществляет работу с файлом конфигурации. Предоставляет данные конфигурации другим классам. Синглтон.

* 1. CBaseUnit, CCellUnit, CLineUnit, CSquareUnit

Представляют собой графические примитивы для отрисовки.

Каждый из классов реализует свой метод render(…), в котором осуществляет свою отрисовку в зависимости от состояния (активен/неактивен, идет анимация или нет).



* 1. CRenderEngine

Представляет собой примитивный однопоточный движок для отрисовки юнитов.

Отрисовка производится непосредственно на массив пикселей, через вызов метода render() для требуемых объектов.

На каждом такте отрисовки движок копирует изменения в device context окна приложения.

При большом количестве юнитов (например: размерами 1х1 пикс) движок проседает по производительности и начинает кушать много памяти за счет большого количества юнитов.

Также просадки по производительности наблюдаются при большом количестве одновременных операций анимации.

1. Общие замечания

Я старался писать максимально простой код, однако работа с массивами реализована довольно спорным способом. Было бы прозрачней для понимания реализовать их в виде двумерных массивов и оправдания тут нет)

Изначально отрисовка планировалась с помощью примитивов gdi, однако код получился еще более громоздким, поэтому было решено рисовать прямо в памяти, что дало некоторый прирост производительности.

Код не проверялся профайлерами и может быть не оптимален.