

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ
по лабораторной работе №3
по дисциплине «ООП»
ТЕМА: ДОБАВЛЕНИЕ ЛОГИРОВАНИЯ

Студент гр. 9304

Преподаватель

Прокофьев М.Д.

Размочаева Н.В.

Санкт-Петербург

2020

Цель работы.

Обучение работе с классами, конструкторами на языке в C++.

Задание.

Создан набор классов, которые отслеживают игрока и элементы на поле, и выводят/сохраняют информацию об их изменениях.

Обязательные требования:

- Реализована возможность записи логов в терминал и/или файл
- Взаимодействие с файлом реализовано по идиоме RAII
- Перегружен оператор вывода в поток для всех классов, которые должны быть логированы

Дополнительные требования:

- Классы, которые отслеживают элементы, реализованы через паттерн Наблюдатель
- Разделение интерфейса и реализации класса логирования через паттерн Мост

Основные теоретические положения.

Класс - это пользовательский тип данных.

Конструктор копирования используется для инициализации класса путем создания копии необходимого объекта.

Оператор присваивания копированием (или «копирующее присваивание») используется для копирования одного класса в другой (существующий) класс.

Конструкторы перемещения принимают ссылку на значение объекта класса и используются для реализации передачи владения ресурсами объекта параметра.

Метод в объектно-ориентированном программировании — это функция или процедура, принадлежащая какому-то классу или объекту. Как и процедура в процедурном программировании, метод состоит из некоторого количества

операторов для выполнения какого-то действия и имеет набор входных аргументов.

Интерфейс — программная/синтаксическая структура, определяющая отношение между объектами, которые разделяют определённое множество и не связаны никак иначе. При проектировании классов, разработка интерфейса тождественна разработке спецификации (множества методов, которые должен реализовывать каждый класс, использующий интерфейс).

Перегрузка операторов позволяет определить действия, которые будет выполнять оператор. Перегрузка подразумевает создание функции, название которой содержит слово `operator` и символ перегружаемого оператора. Функция оператора может быть определена как член класса, либо вне класса.

Лог (log) - это специальный журнал, в котором хранится информация о состоянии работы приложения (программы). Под таким журналом можно понимать и записи в обычный текстовый файл.

Выполнение работы.

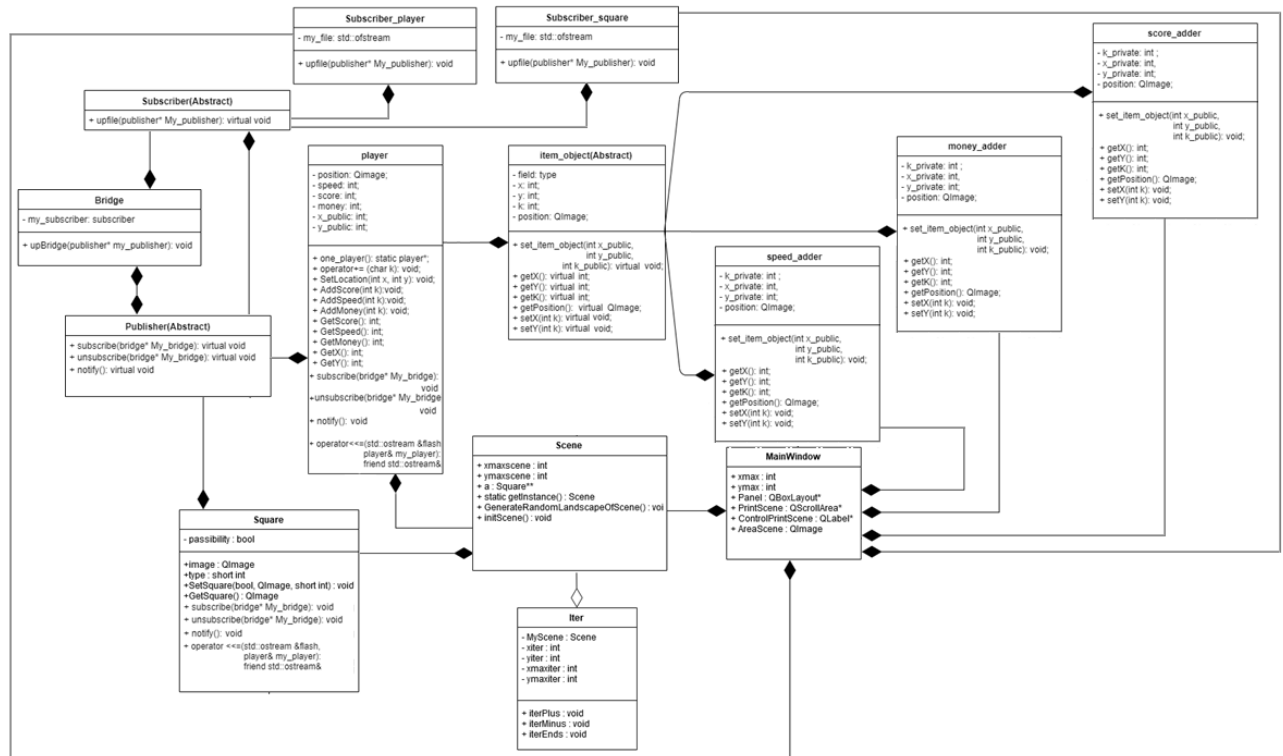


Рисунок 1 – UML-схема

Элементы для логирования были выбраны следующие: `player` - для показателя перемещения, `square` - для вывода того, какой предмет был получен игроком (`player`) при перемещении (естественно, если предмет был взят)

Был создан абстрактный класс `publisher`, который содержит в себе такие виртуальные функции, как `subscribe()`, `unsubscribe()` и `notify()`, где первая служит для занесения в список элементов (в данном случае `subscribers`), которые будут отвечать за вывод в файл, вторая, соответственно для удаления таких элементов и `notify()`, с помощью которой происходит вызов всех элементов производить вывод в файл. Т. к. класс `publisher` абстрактный, его методы были реализованы в классах `player` и `square` (соответственно: игрок и клетка). И естественно, эти два класса являются потомками `publisher`. От того, в каком классе происходит вызов соответствующих методов, зависит сам вывод, который поступит в файл `log.txt`. Кроме того был создан виртуальный класс `subscriber` (в данном случае - наблюдатель). Именно его роль заключается в том, чтобы записывать вывод в

файл. Т. к. класс абстрактный, для него было создано два других класса: `subscriber_player` и `subscriber_square`, которые, соответственно, имеют тесную связь с `player` и `square`. В данных классах реализованы методы записи файл. Где сам метод записи представляет из себя перегруженный оператор `<<`. Этот оператор реализован в `player` и `square`. Также, сделан класс `bridge` (мост), который связывает `subscriber` и `publisher`. В конструкторе моста вызывается метод записи из `subscriber` с аргументом `publisher`, т. к. `publisher` по логике программы является источником информации для `subscriber`.

Тестирование.

1) Запуск программы:

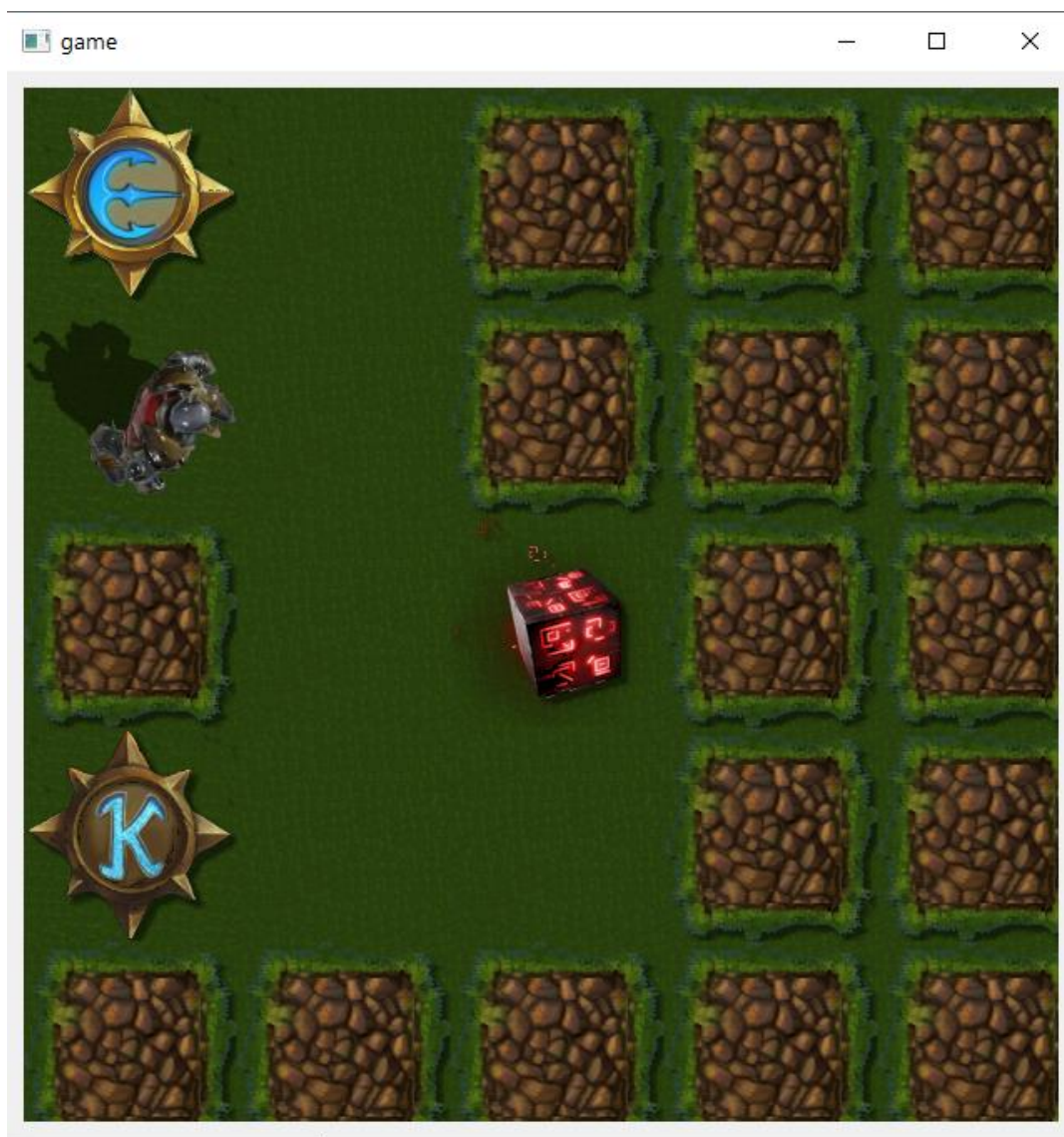


Рисунок 2 – Запуск

2) Игрок сделал два шага: вправо и вниз:

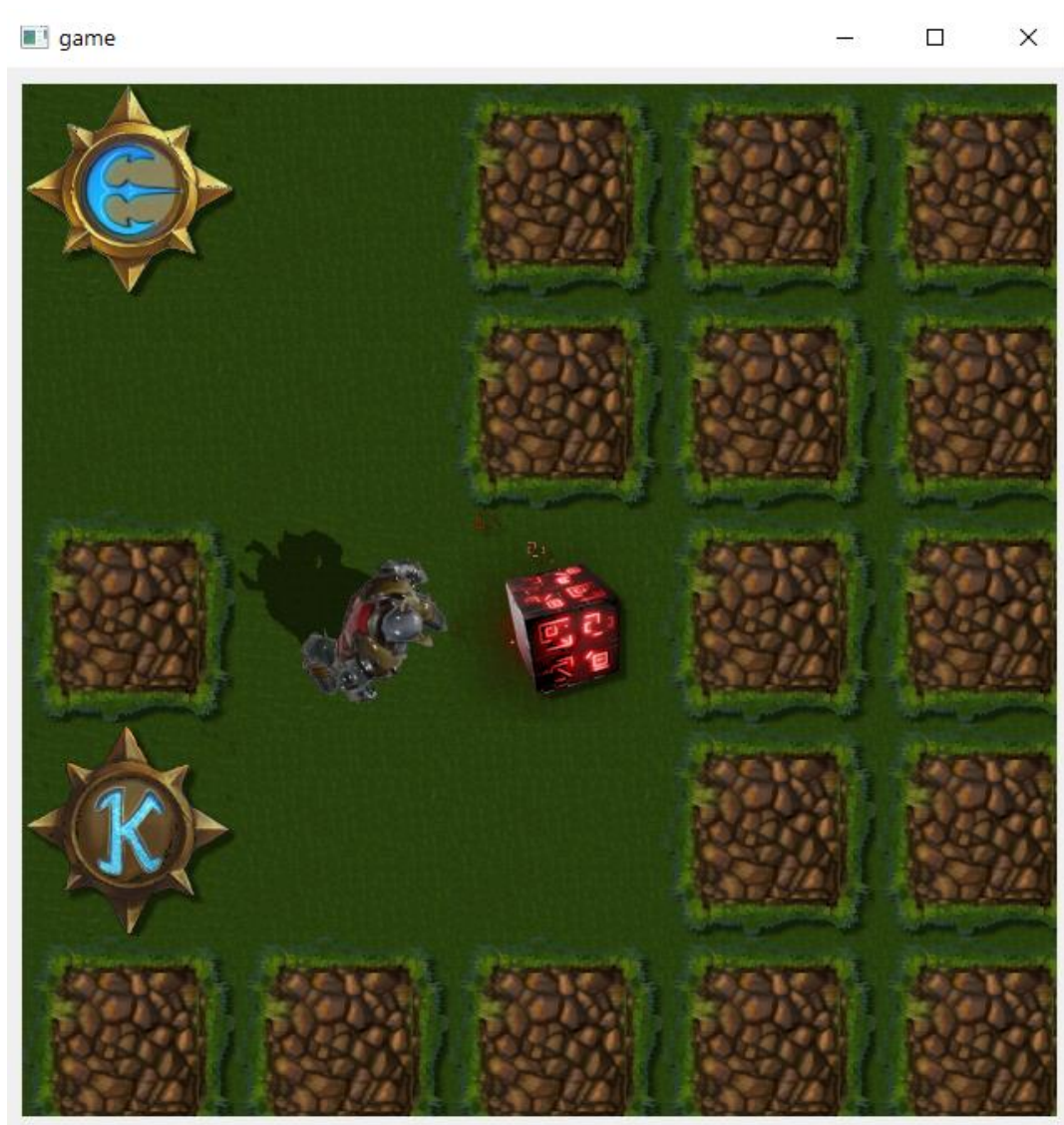


Рисунок 3 – Игрок сделал передвижения

3) Игрок сделал шаг вправо и подобрал бонус (в данном случае – на скорость):

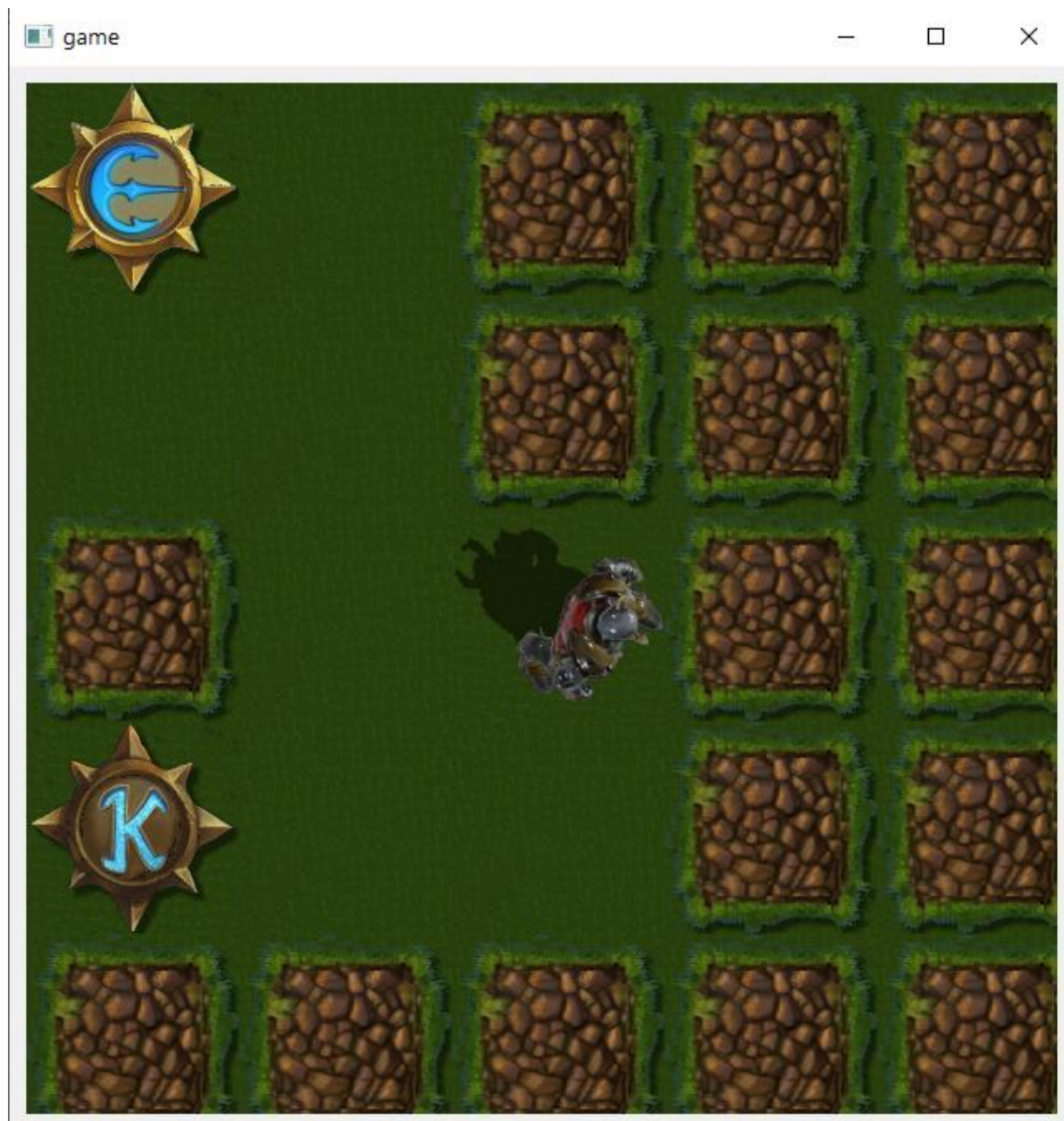


Рисунок 4 – Игрок подобрал бонус

4) Завершение игры и проверка лога, который находится в файле log.txt:

```
the player went to the point: [1,1]. time:Tue Nov 03 09:21:49 2020  
the player went to the point: [1,2]. time:Tue Nov 03 09:21:51 2020  
the player picked up the item [speed]. time:Tue Nov 03 09:23:48 2020  
the player went to the point: [2,2]. time:Tue Nov 03 09:23:48 2020
```

Рисунок 5 – Проверка лога

Лог верен, поскольку игрок действительно сначала пришел в позицию [1,1], а затем в позицию [1,2], после чего перейдя в позицию [2,2] берёт предмет.

Выводы

Были получены практические знания по использованию логирования в программировании. Созданы классы для вывода данных в лог. При работе был использован перегруженный оператор << для записи в файл. Кроме того, при написании, были использованы паттерны Наблюдатель и Мост.