



# РАЗРАБОТКА ИГР НА C++

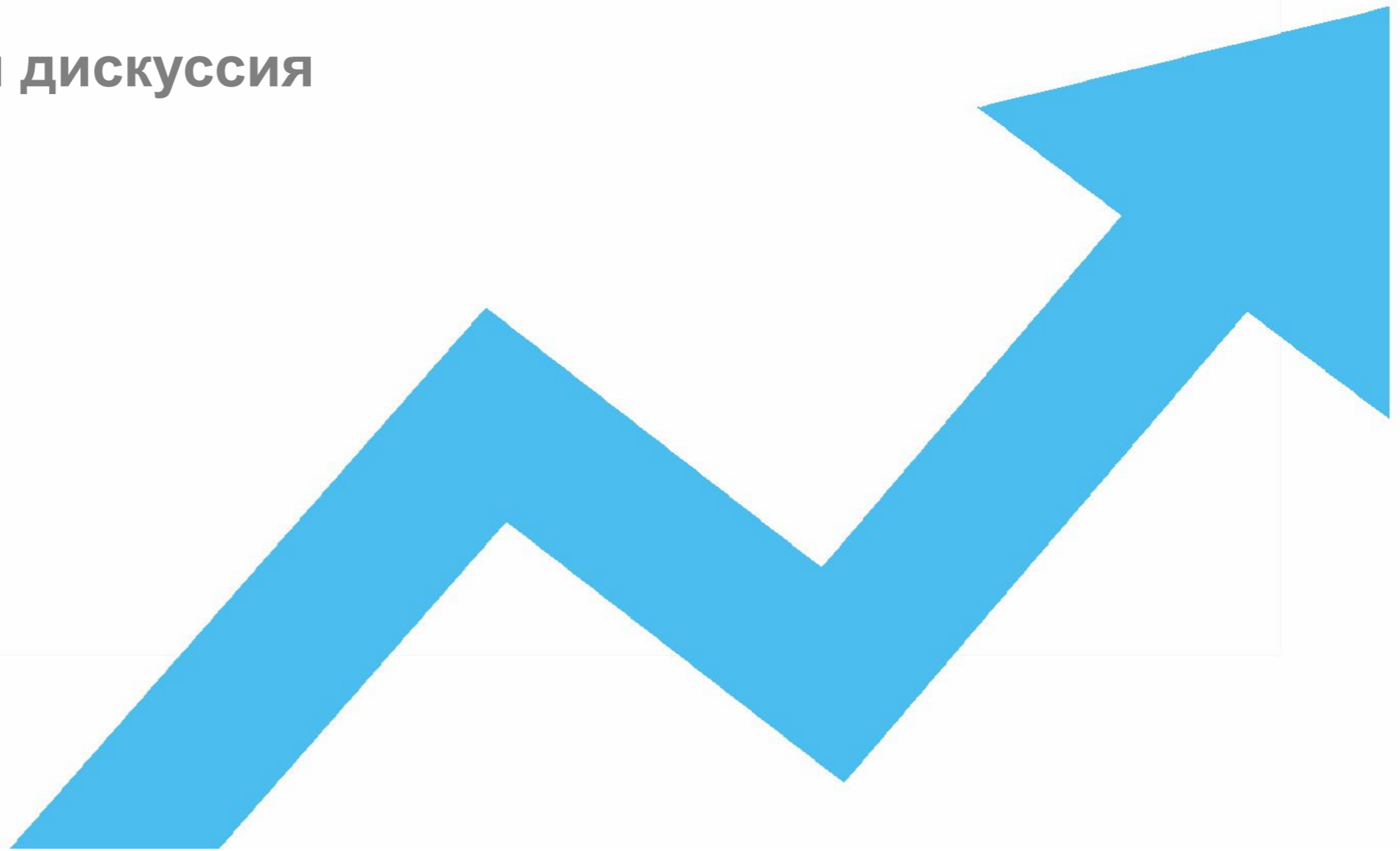
# ПРИЕМЫ И ПРАКТИКИ

Роман Кузнецов,  
технический продюсер  
Alawar


[r.kuznetsov@alawar.com](mailto:r.kuznetsov@alawar.com)

# ПЛАН ЛЕКЦИИ

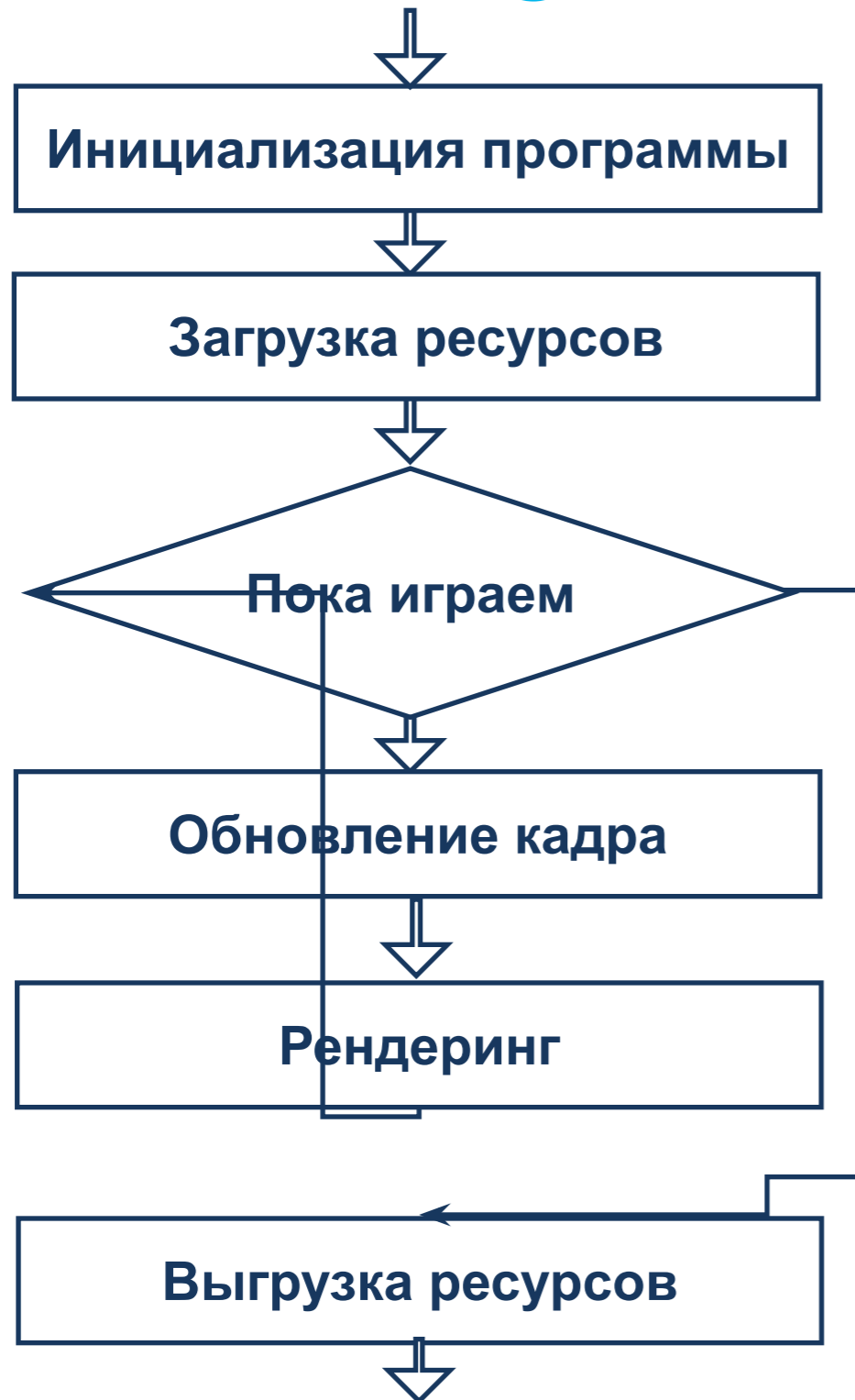
- Программная организация игр
- Особенности C++ при разработке игр
- Ваши вопросы и дискуссия



# ИГРА – ЭТО ПРОГРАММА

- 1) Процесс
  - 2) Одна или несколько нитей исполнения (threads)
  - 3) Язык программирования
    - Native-язык (C/C++)
    - Managed-язык (C#, Java)
    - Скриптовый язык (Lua, Perl, Python)
- 
- A decorative graphic consisting of two overlapping diamond shapes (squares rotated 45 degrees) in a light blue color. The top diamond has a single white circle in its center, while the bottom diamond has three white circles arranged vertically in its center.

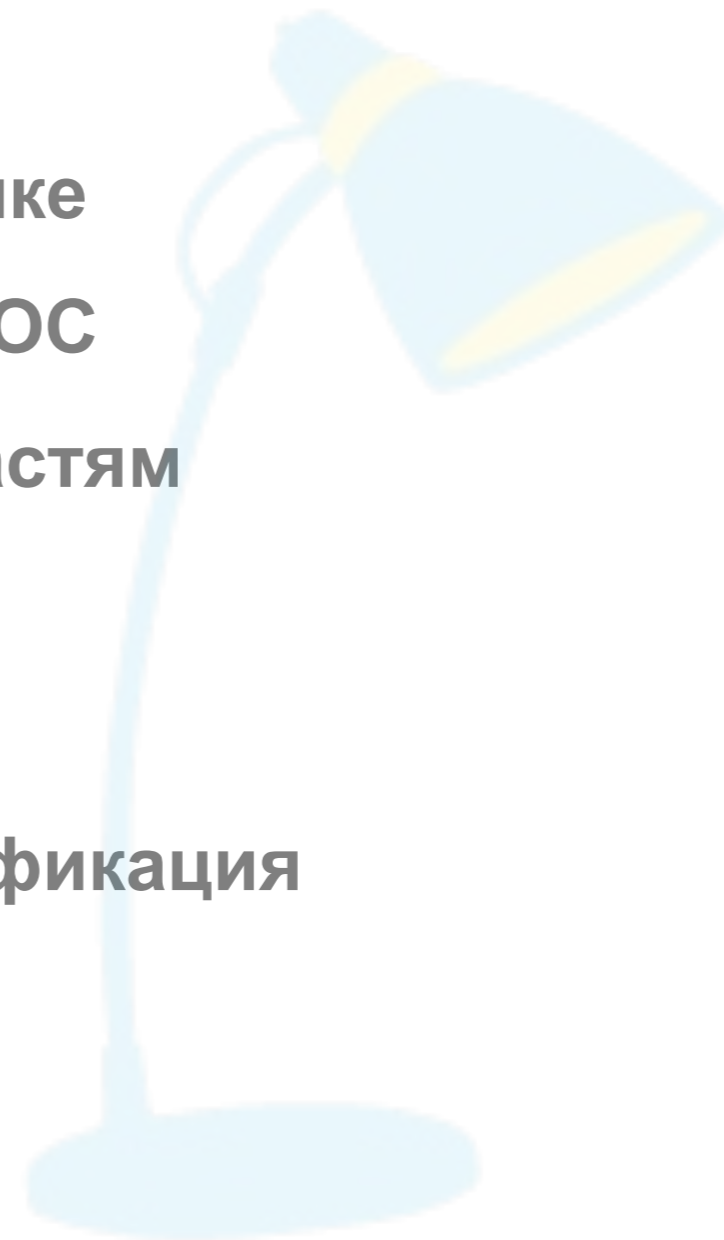
# ОРГАНИЗАЦИЯ ИГРЫ



- 1) Инициализация программы (создание окна, инициализация платформозависимых систем)
- 2) Загрузка ресурсов (текстуры, модели, звуки, шейдеры)
- 3) Обновление кадра (логика игры)
- 4) Рендеринг (отображение на экране)
- 5) Выгрузка ресурсов

## **C++ VS C#/JAVA/OBJS**

- **Нативный код или управляемый?**
- **Наличие библиотек на нужном языке**
- **Доступ к низкоуровневым частям ОС**
- **Доступ к платформозависимым частям ОС**
- **Скорость исполнения**
- **Наличие специалистов и их квалификация**



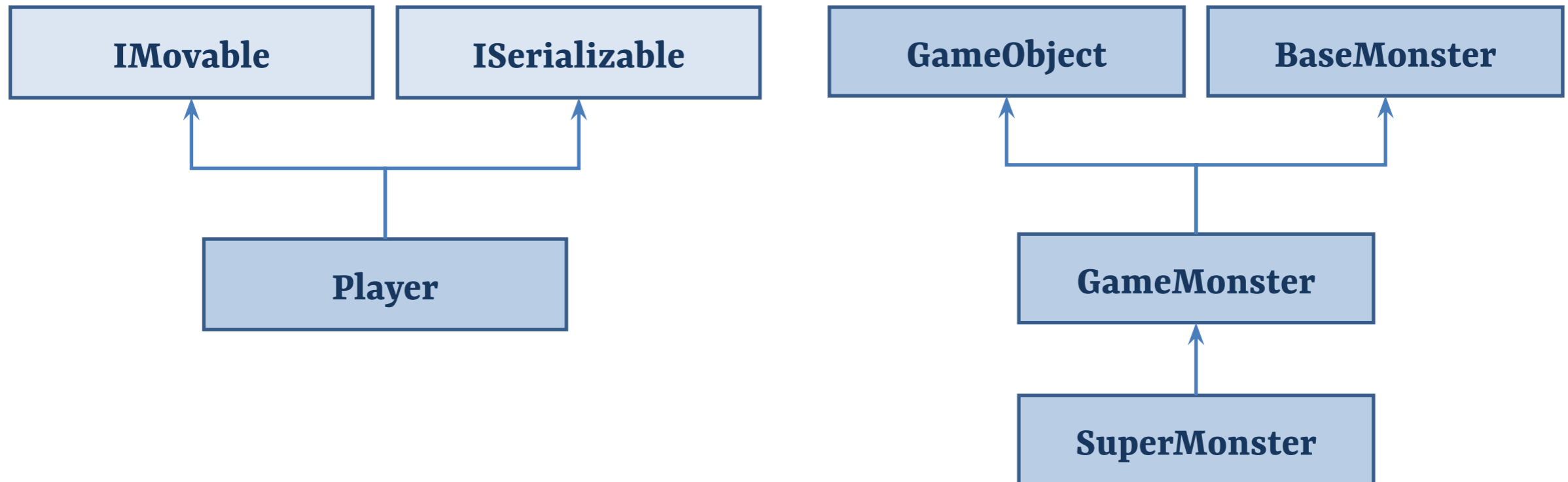
## ОСОБЕННОСТИ C++

- **Наследование**
- **Шаблоны**
- **Выделение памяти**
- **Обработка ошибок**
- **Стандартная библиотека**
- **Паттерны проектирования**



# НАСЛЕДОВАНИЕ

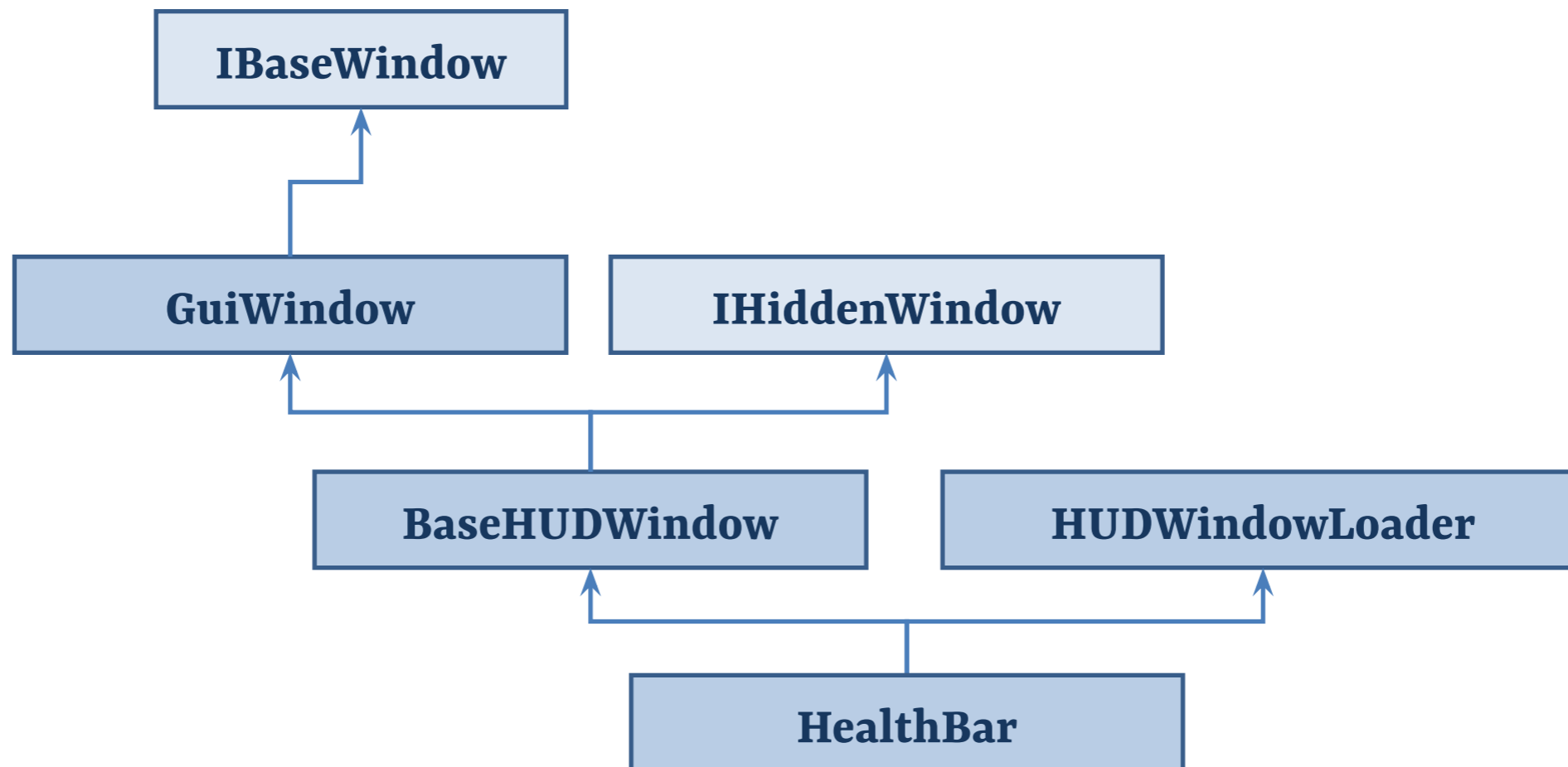
Используем интерфейсы  
при множественном наследовании



**PROFIT: не используем `dynamic_cast` и `RTTI`**

# НАСЛЕДОВАНИЕ

Не создаем лишние сущности  
и глубокие иерархии



**PROFIT: значительное упрощение кода**



# НАСЛЕДОВАНИЕ

**Тотальный полиморфизм –  
это ПЛОХО!**

*«А вдруг от класса, который я пишу, в будущем захотят сделать наследника»*

*«Напишу виртуальный деструктор на всякий случай»*

*«Я наследую все классы в игре от IObject»*

**PROFIT: экономия памяти и времени исполнения**

# ШАБЛОНЫ

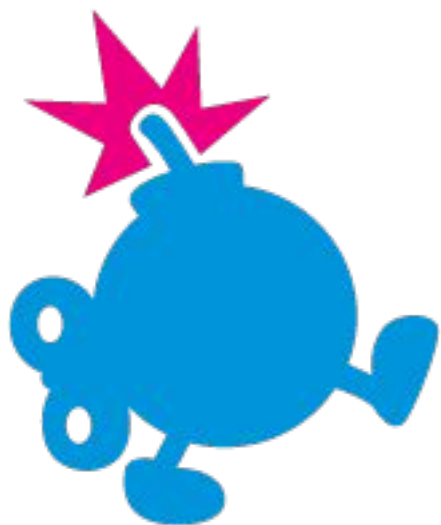
## Причины превращения кода из простого в сложный:

*«Шаблоны – это круто!»*

*«Я читал книжку Андрея Александреску»*

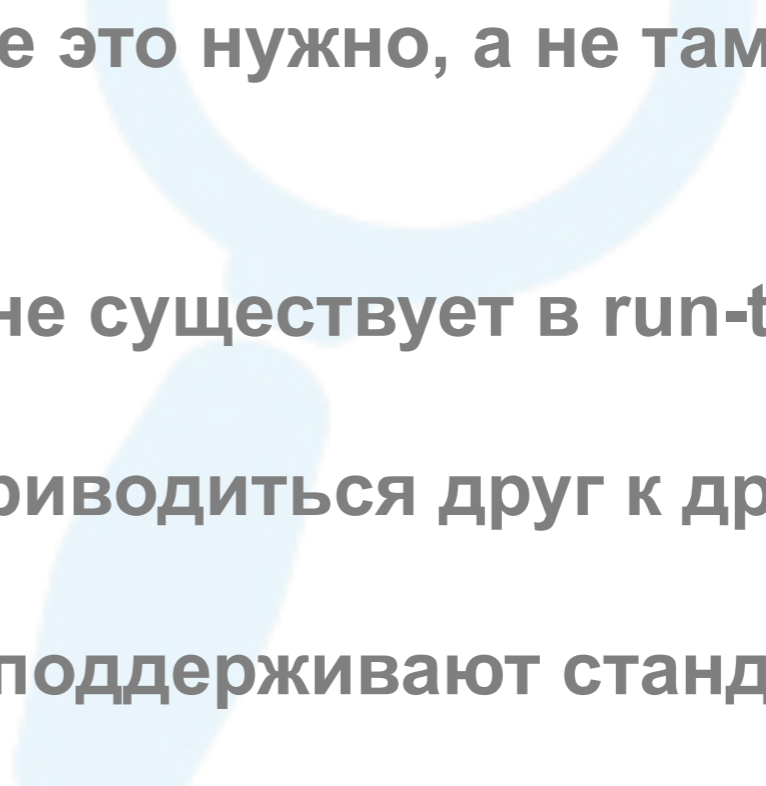
*«Я разобрался с boost»*

*«Я люблю головоломки и «абстрактный» код»*



# ШАБЛОНЫ

## Джентельменский подход к использованию:

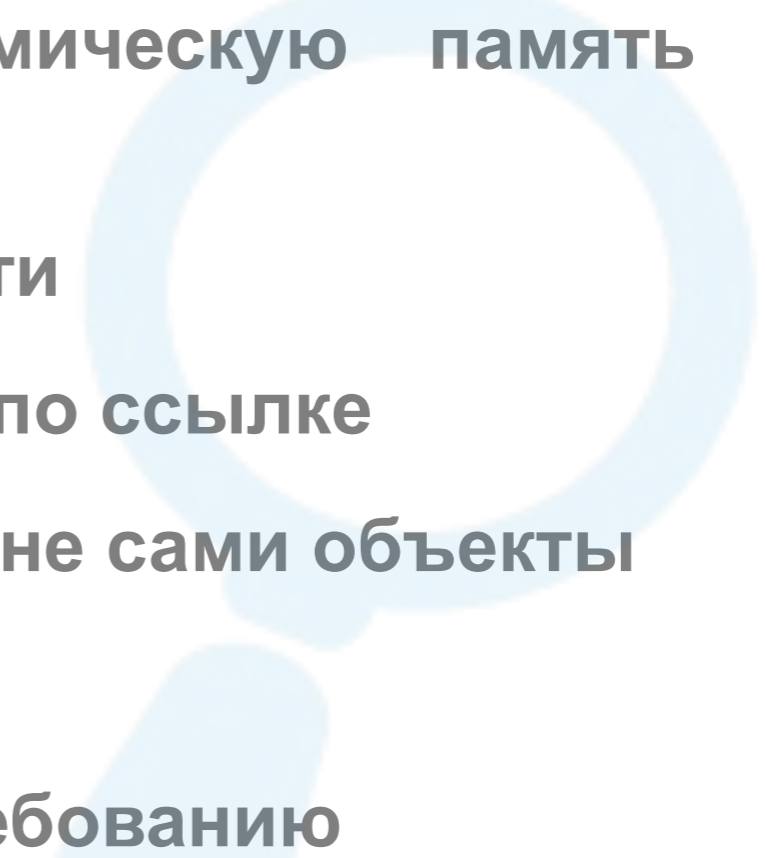
- Пишем шаблонные классы там, где это нужно, а не там, где хочется!
  - Помним, что шаблонных классов не существует в run-time
  - Помним, что типы могут неявно приводиться друг к другу
  - Помним, что не все компиляторы поддерживают стандарт C++ в отношении шаблонов
- 
- A large, light blue magnifying glass graphic is positioned on the right side of the slide, with its handle pointing downwards and its lens area overlapping the text of the list.

# ШАБЛОНЫ

## Где полезно использовать

- Математические алгоритмы и структуры данных
- Реализация паттернов проектирования
- Собственные контейнеры

# ВЫДЕЛЕНИЕ ПАМЯТИ

- 1) Стараемся не выделять динамическую память внутри игрового цикла
  - 2) Используем предвыделение памяти
  - 3) Используем передачу аргументов по ссылке
  - 4) Храним в коллекциях указатели, а не сами объекты
  - 5) Move-семантика в C++11
  - 6) Загрузка/выгрузка ресурсов по требованию
- 
- A large, light blue magnifying glass graphic is positioned on the right side of the slide, with its handle pointing downwards and its lens area overlapping the right side of the list items.

# ОБРАБОТКА ОШИБОК

- 1) Коды ошибок вместо исключений!
- 2) Используем логирование
- 3) Используем ASSERT'ы
- 4) Собираем статистику



# СТАНДАРТНАЯ БИБЛИОТЕКА

Три «кита»: `vector`, `list` и `map`

Классы `string` и `wstring`

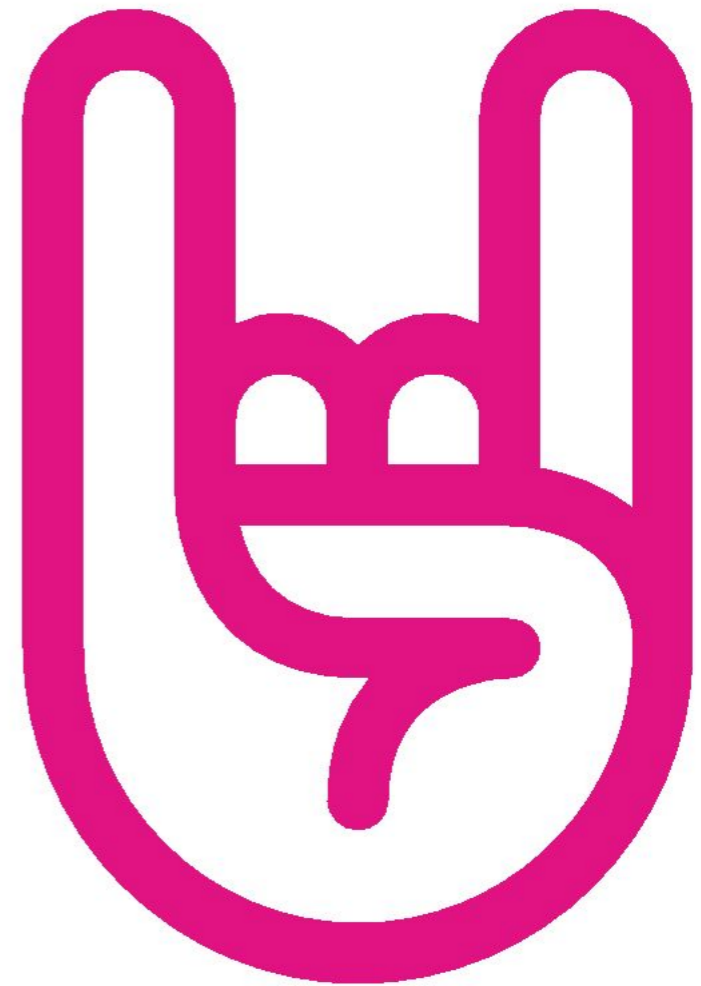
В стандарте C++11:

- Умные указатели  
(`shared_ptr`, `weak_ptr` и т.д.)
- Потoki и атомарные операции



# ПАТТЕРНЫ ПРОЕКТИРОВАНИЯ

- 1) Синглтон
- 2) Абстрактная фабрика
- 3) Фасад
- 4) Наблюдатель
- 5) Итератор





## ВЫВОДЫ

**1**

**С++ успешно используется и будет продолжать использоваться при разработке игр**

**2**

**В С++ есть особенности, которые необходимо учитывать при разработке**

**3**

**Программисты С++ востребованы в индустрии компьютерных игр**

Роман Кузнецов, технический продюсер

r.kuznetsov@alawar.com



alawar

# Спасибо за внимание! Ваши вопросы

Больше о нашей работе и  
решениях на  
<http://habrahabr.ru/company/alawar>

