

# Разработка и интеграция сетевых компонентов в шаблон многопользовательской игры на Unreal Engine 4

Студент: Боженко Дмитрий

Научный руководитель: Витюков Ф. А.



# Введение

Сетевые компоненты — это элементы сетевой игры.

Сетевые элементы, относящиеся  
к взаимодействию игроков

1. Возрождение;
2. Изменение состояния объекта (внутриигровые очки, уровень здоровья);
3. Запрос настроек игрового мира с сервера на клиент;
4. Создание виджетов с информацией об игроках и игровом мире;
5. Создание динамических виджетов интерфейса.

Сетевые элементы, относящиеся  
к возможностям многопользовательских игр:

1. Создание онлайн сессий;
2. Авторизация;
3. Создание лобби;
4. Создание таблиц лидеров;
5. Создание системы подбора игроков;

# Постановка задачи

- **Цель работы:** на примере шаблона многопользовательской игры реализовать графическое трехмерное приложение с сетевым взаимодействием пользователей по LAN и сети Интернет.
- **Поставленные задачи:**
  - Изучить принципы работы сетевого программирования;
  - Изучить принципы создания интерактивных виджетов;
  - Изучить принципы работы с программными интерфейсами встроенной подсистемы;
  - Реализовать автоматическую систему начисления очков и таблицу лидеров на уровне приложения;
  - Реализовать возможность создания лобби трех видов для вариативного использования шаблона.

# Обзор существующих решений



C++:

- ✓ Позволяет достичь максимальную производительность работы приложения;
- ✓ Делает проект удобочитаемым при разработке;
- ✓ Доступен открытый исходный код.

Визуальное программирование  
(Blueprints):

- ❑ Уменьшает скорость работы приложения;
- ❑ Ухудшает удобочитаемость проекта.



- ❑ Имеет лучшую поддержку при разработке только под мобильные устройства и VR;
- ❑ Отсутствует открытый исходный код движка;
- ❑ Отсутствует обширная и гибкая онлайн подсистема.

# Актуальность

Проект разработан на C++ с активным использованием объектно-ориентированного программирования (ООП).

```
#pragma once
#include ...

class UTextBlock;
class UButton;
// ...
}

UCLASS()
// ! derived blueprint class
class LABA_API UStatusControl : public UserWidget, public FTickableGameObject
{
    GENERATED_BODY()

protected:
    virtual bool Initialize() override;
    virtual void Tick(float DeltaTime) override;
    virtual bool IsTickable() const override;
    virtual TStatId GetStatId() const override;

public:
    UFUNCTION()
    void OnStartButtonMatchedClicked();

    UFUNCTION()
    void OnExitButtonClicked();
};
```

Рис. 1 — Фрагмент кода создания виджета на C++

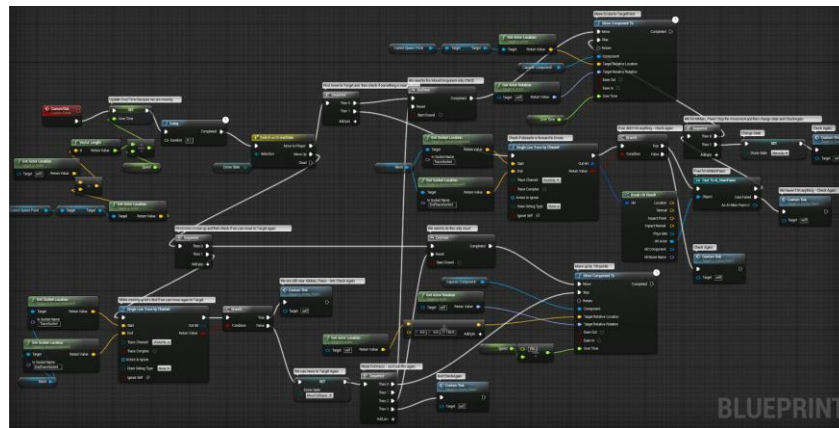


Рис. 2 — Разработка проекта на Blueprints

# Сетевое программирование — репликации

Репликация — механизм синхронизации игрового мира между игроками

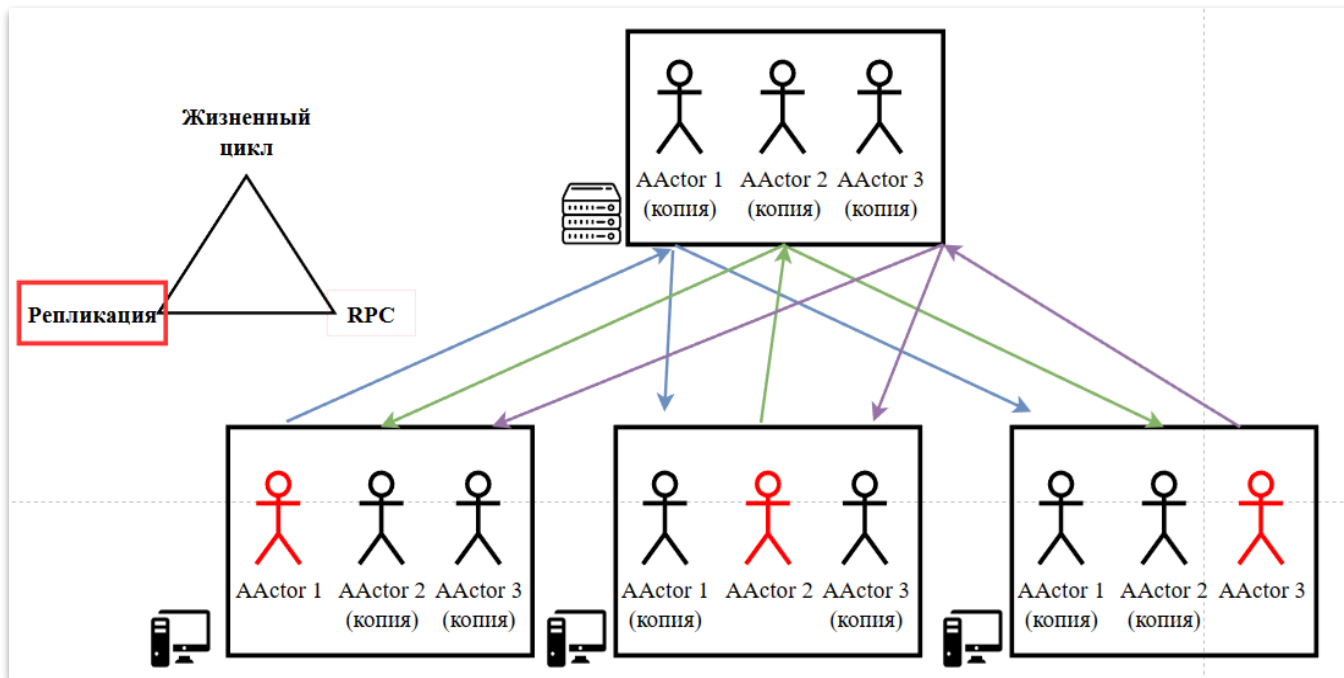


Рис. 3 — Схема репликации в сетевых приложениях

# Сетевое программирование — RPC

Remote Procedure Call (RPC) — вызов функции в другом адресном пространстве

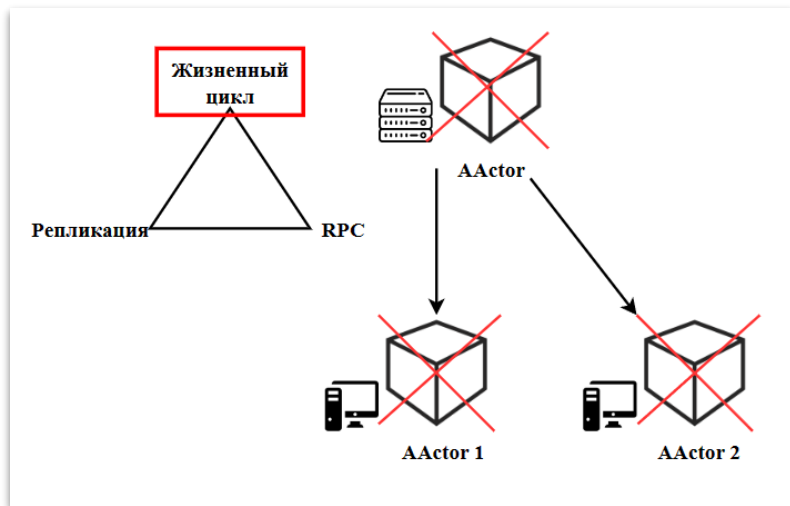


Рис. 4 — Схема жизни объектов в сетевых приложениях

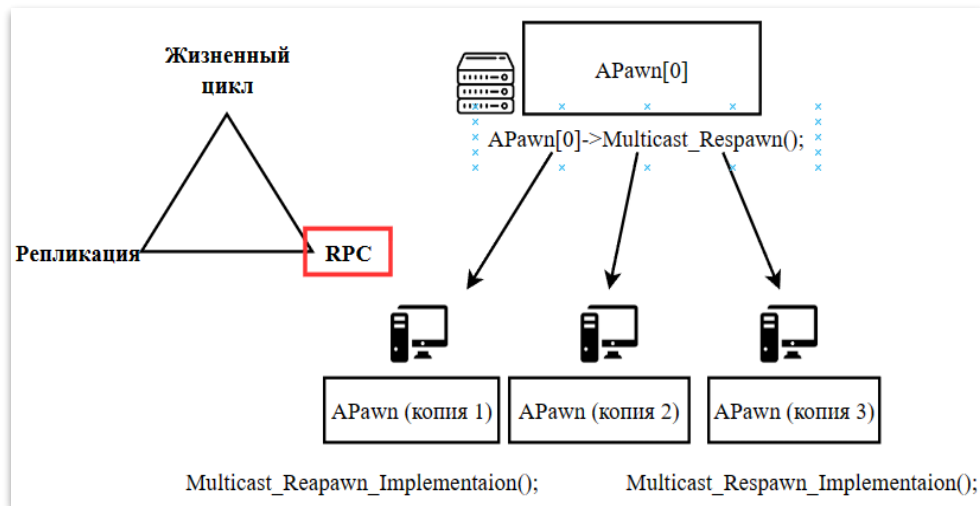


Рис. 5 — Схема вызова RPC

# Сетевые подключения

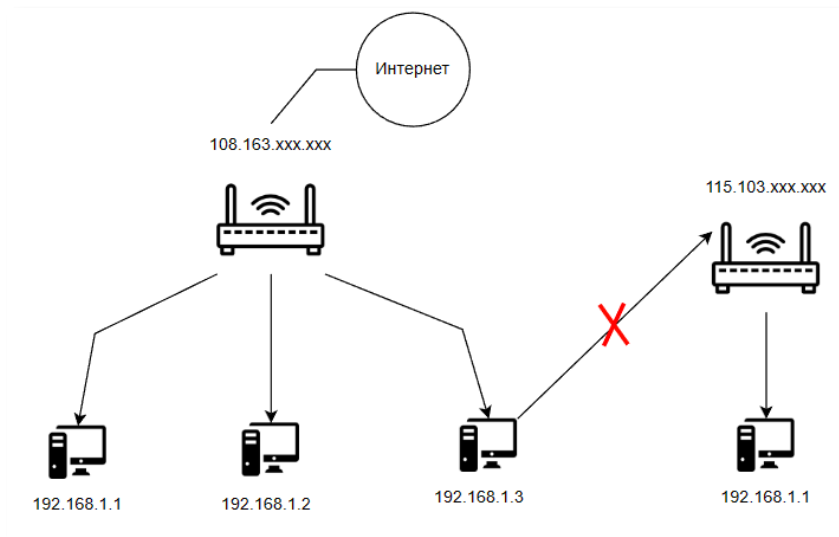


Рис. 6 — Подключение пользователей по LAN

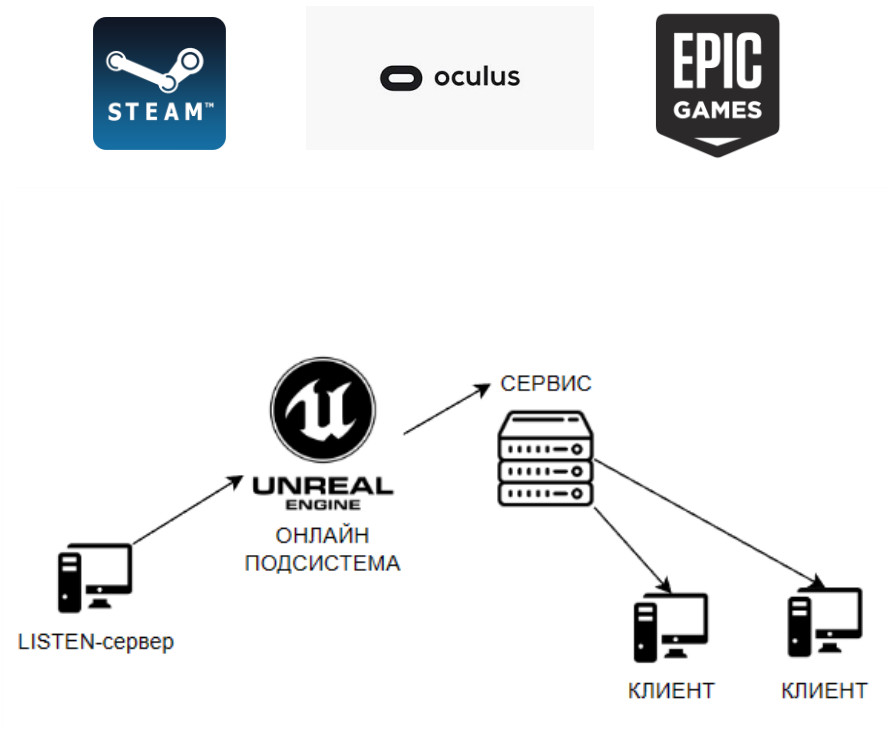


Рис. 7 — Схема подключений через онлайн подсистему



# Решение проблемы сетевых задержек



Рис. 8 — Таймер до начала игрового процесса

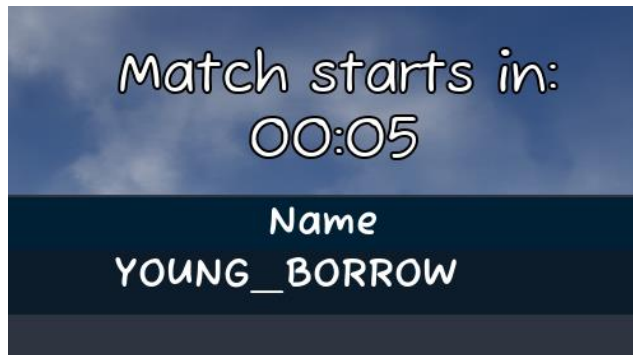


Рис. 9 — Таймер на уровне лобби

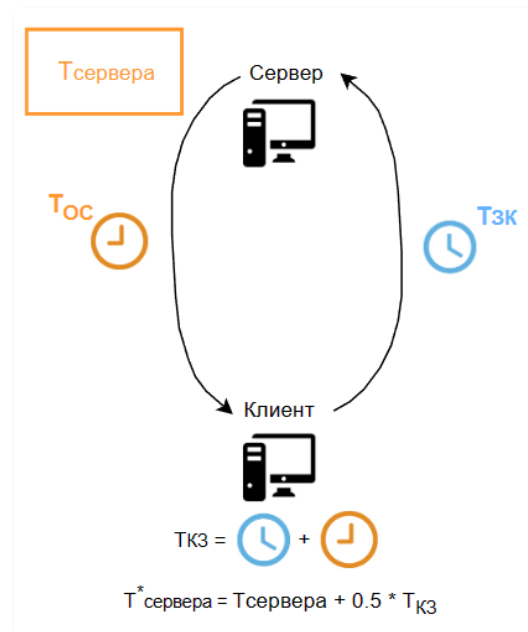


Рис. 10 — Схема круговой задержки

$T_{сервера}$  — время сервера на момент принятия запроса

$T_{зк}$  — время задержки запроса клиента

$T_{ос}$  — время задержки ответа сервера

$T_{КЗ}$  — время кольцевой задержки

# Разработка таблицы лидеров на уровне приложения

Back

Players leagues

Recruits	Guardians	Crusaders	Legends
-2000 — -1000	-999 — 0	1 — 1000	1001 — 2000

Top Guardians players

Rank	Name	Score	Global Rank
1	Iffor	-25	2
2	ver33dd	-25	3
3	Skallains	-25	4
4	Alta Veritus	-25	5

Back

Players leagues

Recruits	Guardians	Crusaders	Legends
-2000 — -1000	-999 — 0	1 — 1000	1001 — 2000

Top Crusaders players

Rank	Name	Score	Global Rank
1	YOUNG_BORROW	50	1



Рис. 13 — Внутриигровой виджет с информацией о завершившемся матче

Рис. 11 - 12 — Таблица лидеров на уровне приложения

## Разработка отдельных уровней

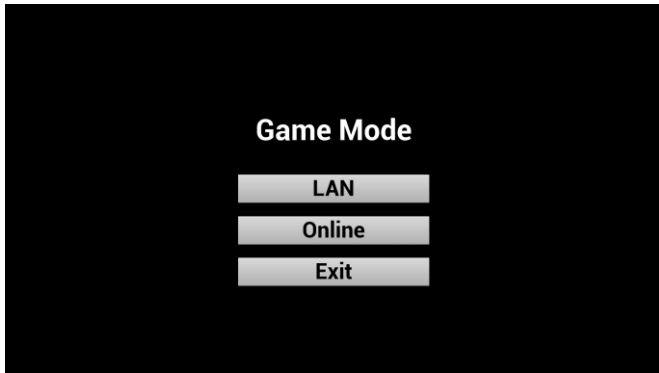


Рис. 14 — Главное меню

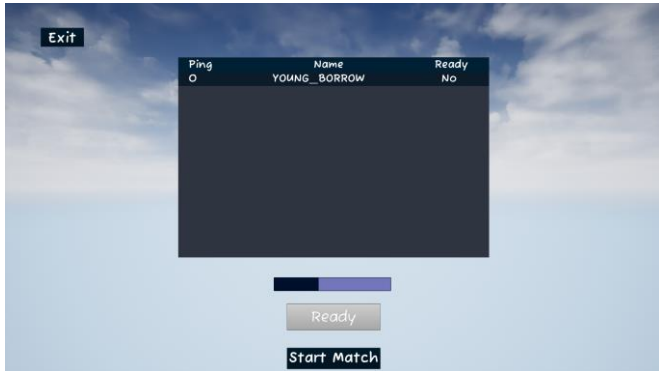


Рис. 16 — Уровень лобби с виджетами

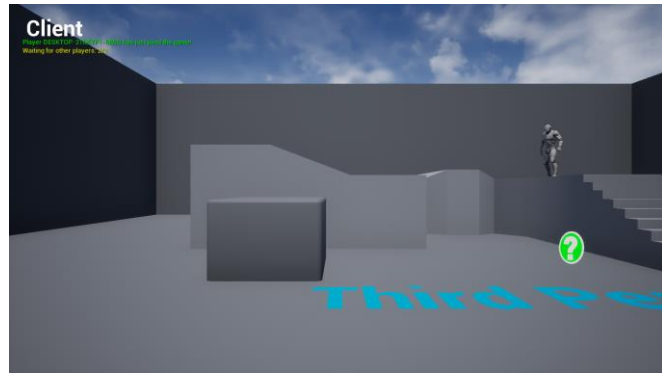


Рис. 15 — Уровень лобби с взаимодействием с игровым миром

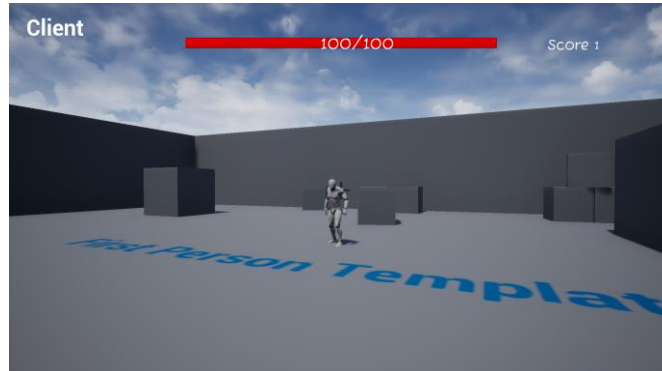


Рис.17 — Основной уровень

# Дальнейшая разработка дипломного проекта

В рамках дальнейшей разработки, чтобы продукт имел презентабельный вид и мог быть добавлен на одну из площадок игр, необходимо:

1. Произвести работу над дизайном;
2. Решить проблему со всеми возможными межсетевыми задержками и проблемами, внедрить клиент-серверное предсказание;
3. Реализовать систему создания матчей и подбора игроков;
4. Реализовать внедрение искусственного интеллекта.

# Заключение

В результате работы были выполнены следующие задачи:

- Изучены принципы сетевого программирования в Unreal Engine 4;
- Изучены принципы создания интерактивных виджетов;
- Получены практические навыки работы с онлайн подсистемой в Unreal Engine 4;
- Реализована автоматическая система начисления очков и таблица лидеров на уровне приложения;
- Реализованы три вида лобби для вариативного использования шаблона.