

# 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, лежащие в основе сетевого приложения.

Сетевые элементы, относящиеся к организации функционирования инфраструктуры многопользовательских приложений:

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

Сетевые элементы, относящиеся к взаимодействию пользователей

1. Изменение состояний объектов;
2. Запрос настроек виртуального пространства с сервера на клиент;
3. Создание виджетов с информацией о пользователях и виртуальном пространстве;
4. Создание динамических виджетов интерфейса;
5. Механизм восстановления исходного состояния пользователя.

# Постановка задачи

- **Цель работы:** разработать шаблон многопользовательского приложения 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):

- ☐ Хуже производительность;
- ☐ Серьёзно осложняет отладку проекта из-за ухудшения читаемости кода.



- ☐ Отсутствует открытый исходный код движка;
- ☐ Разработка на C# и JavaScript;
- ☐ 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++

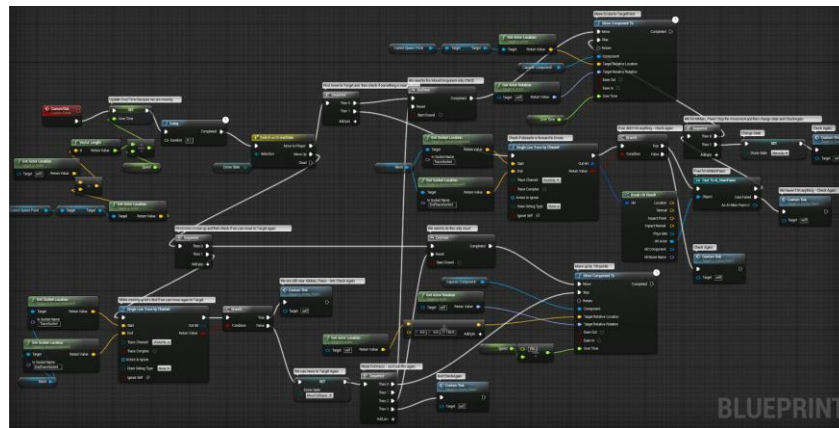


Рис. 2 — Разработка проекта на Blueprints

# Сетевое программирование — репликации

Репликация — механизм синхронизации виртуального пространства между пользователями

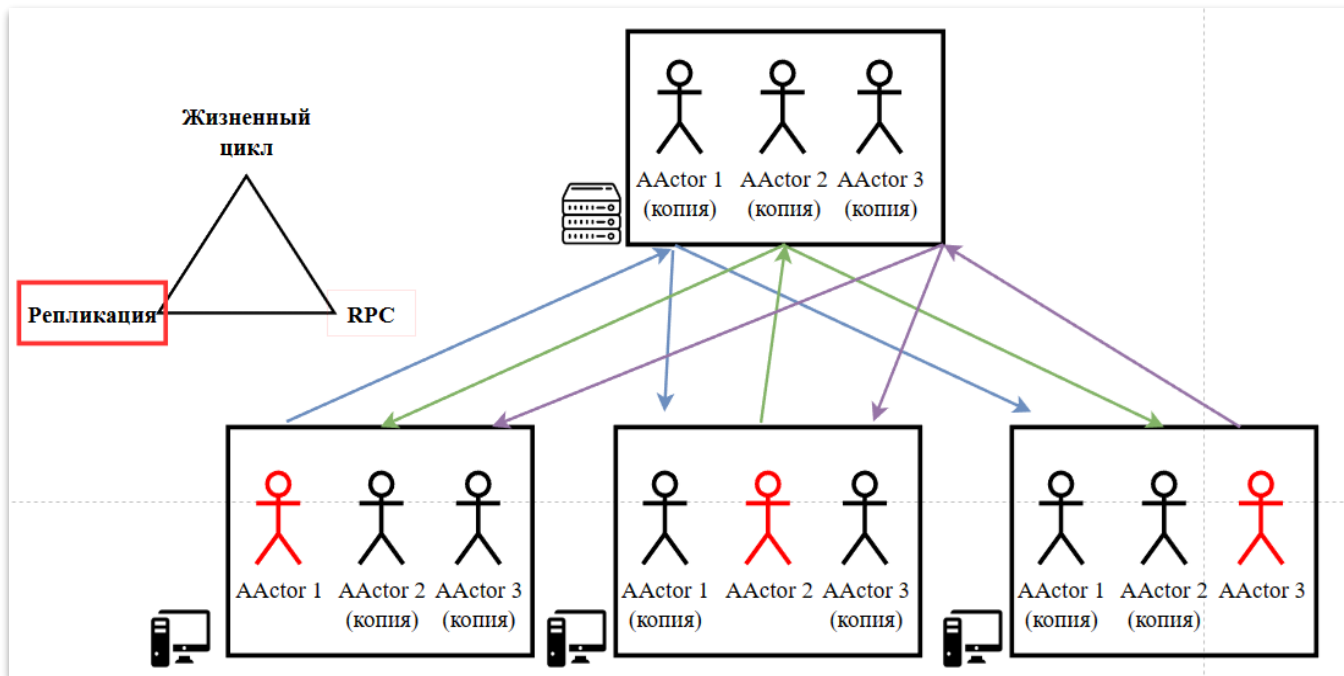


Рис. 3 — Схема репликации в сетевых приложениях

# Сетевое программирование — RPC

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

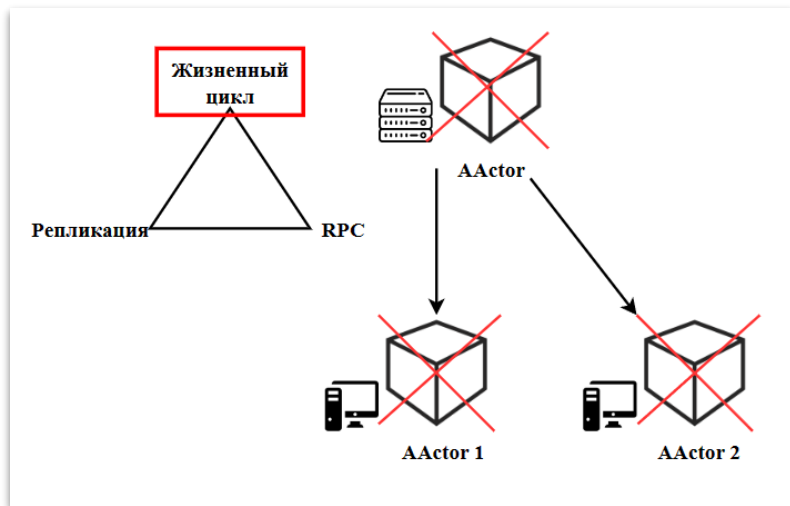


Рис. 4 — Схема жизни объектов в сетевых приложениях

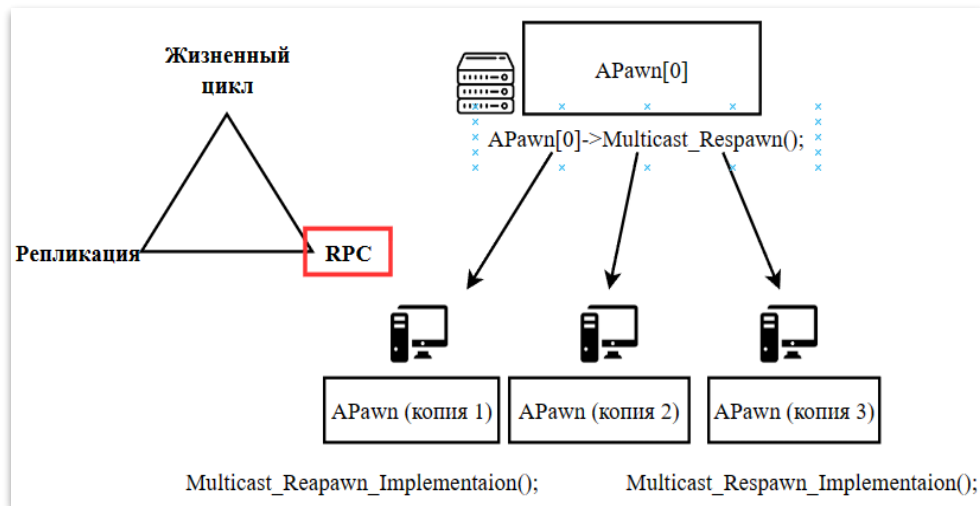


Рис. 5 — Схема вызова RPC

# Сетевые подключения

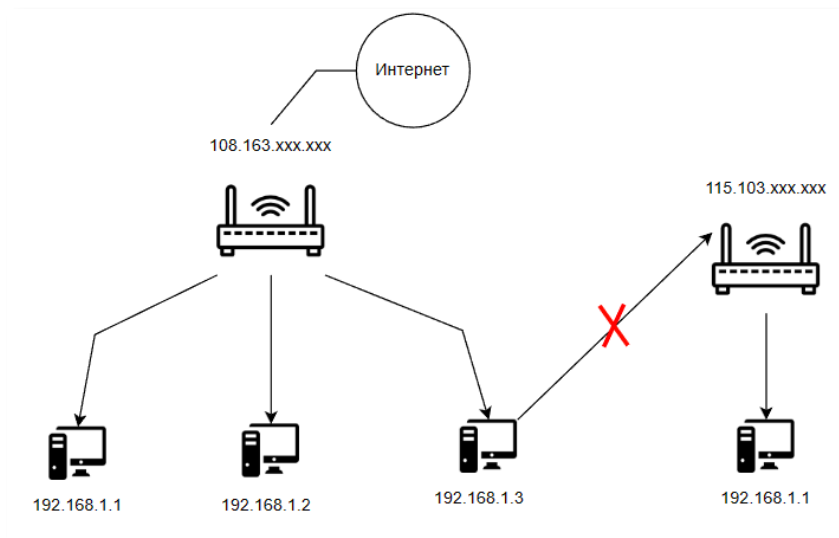


Рис. 6 — Подключение пользователей по LAN

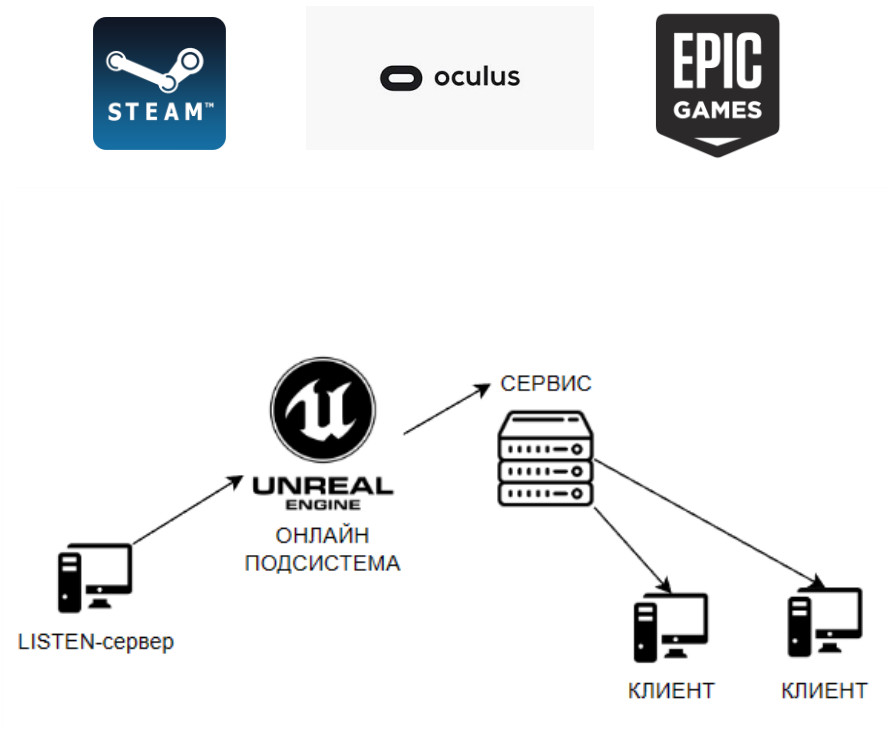


Рис. 7 — Схема подключений через онлайн подсистему



# Решение проблемы сетевых задержек



Рис. 8 — Таймер до начала игрового процесса

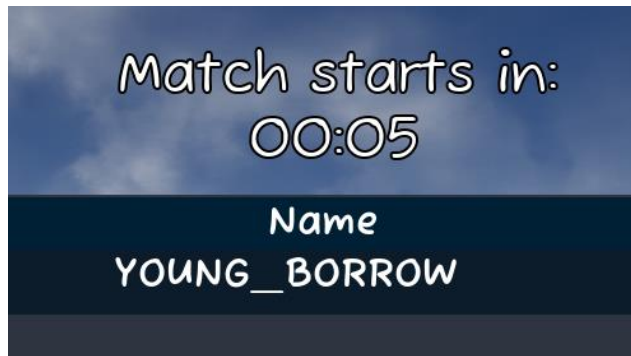


Рис. 9 — Таймер на уровне лобби

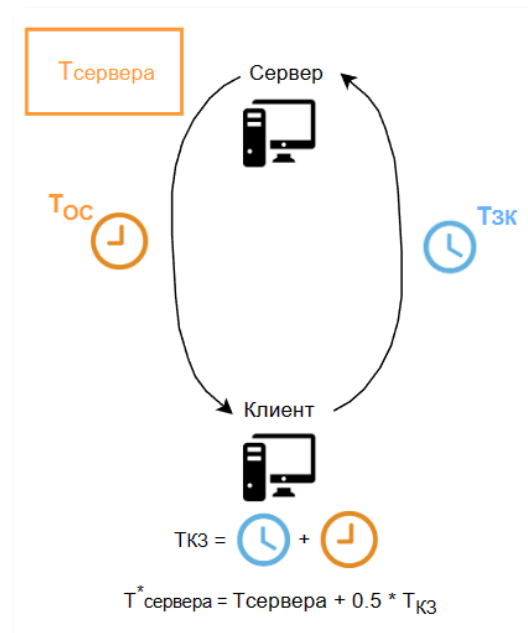


Рис. 10 — Схема круговой задержки

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

$T_{зк}$  — время задержки запроса клиента

$T_{ос}$  — время задержки ответа сервера

$T_{кз}$  — время кольцевой задержки

# Онлайн сессии

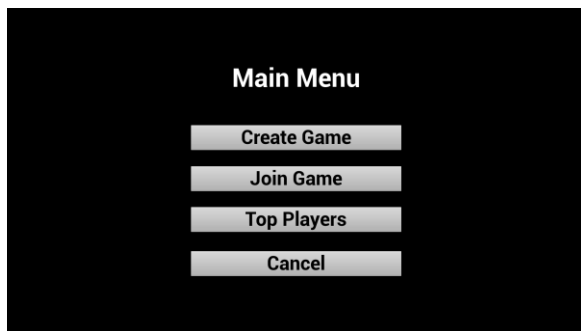


Рис. 11 — Меню для управления сессиями

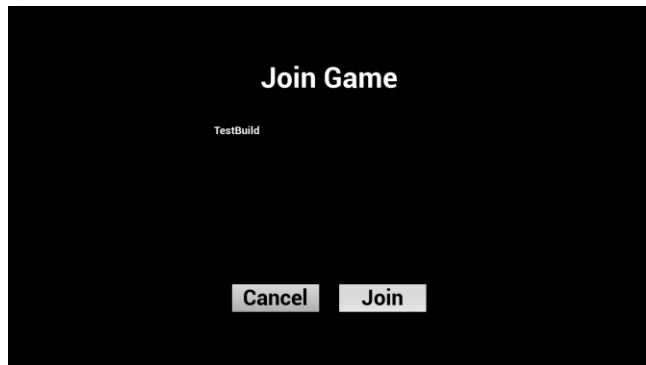


Рис. 13 — Подключение к существующей сессии

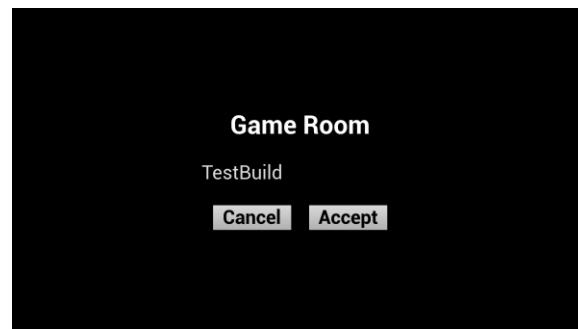


Рис. 12. — Создание онлайн сессии

```
void UAB4GameInstance::CreateSession()
{
    if (!SessionPtr.IsValid()) return;

    FOnlineSessionSettings SessionSettings;

    if ((FOnlineSubsystem::Get()->GetSubsystemName()) == "NULL" || BIsLANGame)
    {
        SessionSettings.bIsLANMatch = true;
        SessionSettings.bUsesPresence = false;
        SessionSettings.bUseLobbiesIfAvailable = false;
    }
    else
    {
        SessionSettings.bIsLANMatch = false;
        SessionSettings.bUsesPresence = true;
        SessionSettings.bUseLobbiesIfAvailable = true;
    }

    SessionSettings.bShouldAdvertise = true;
    SessionSettings.numPublicConnections = 16;
    SessionSettings.bAllowJoinInProgress = true;
    SessionSettings.bUseLobbiesVoiceChatIfAvailable = true;

    FOnlineSessionSetting setting;
    setting.Data = F::ServerName.ToString();
    setting.AdvertisementType = EOnlineDataAdvertisementType::ViaOnlineServiceAndPing;

    SessionSettings.Set(ServerNameKey, setting);
    SessionSettings.Set(SEARCH_KEYWORDS, FString(16, "Test session"), EOnlineDataAdvertisementType::ViaOnlineService);

    const FUniqueNetIdPtr NetID = GetFirstGamePlayer()->GetPreferredUniqueNetId().GetUniqueNetId();
    GetFirstGamePlayer()->SetCachedUniqueNetId(NetID);

    SessionPtr->CreateSessionCompleteDelegates.AddObject(this, &UAB4GameInstance::OnCreateSessionComplete);
    SessionPtr->CreateSession(NetID, SessionNameConst, SessionSettings);
}
```

Рис. 14 — Создание сессии через C++

# 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. 17 — Внутриигровой виджет с информацией о завершившемся матче

Рис. 15 - 16 — Таблица лидеров на уровне приложения

# Разработка отдельных уровней



Рис. 18 — Главное меню

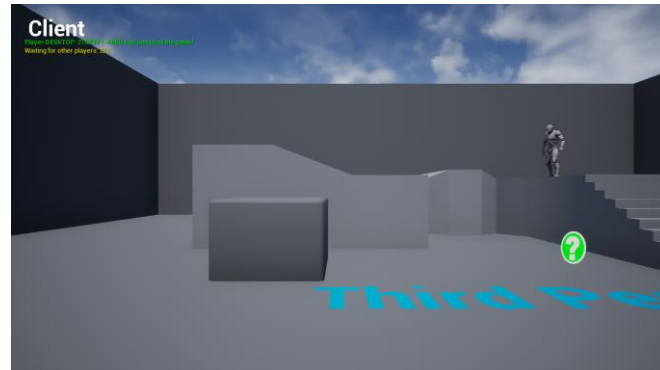


Рис. 19 — Уровень лобби с взаимодействием с виртуальным пространством

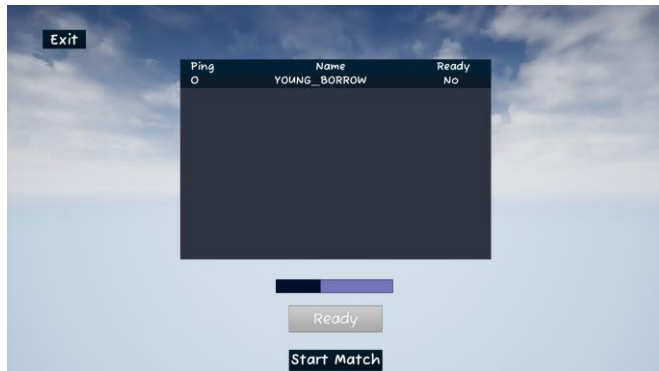


Рис. 20 — Уровень лобби с виджетами

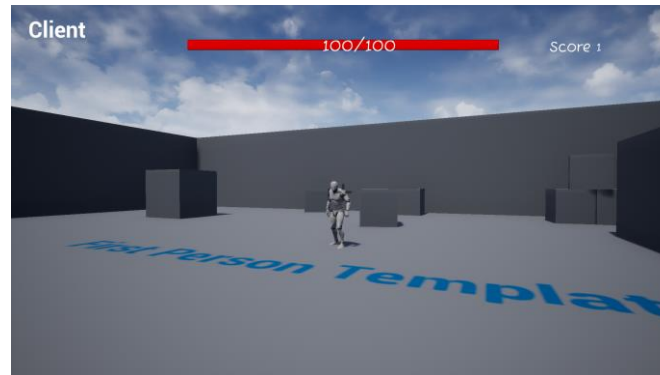


Рис.21 — Основной уровень

# 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.