

# Clash of Clans - C++ 联网对战游戏

License MIT C++ 14 Platform Windows | Android

一个基于 **Cocos2d-x** 引擎的《部落冲突》风格的 C++ 联网对战游戏。该项目实现了完整的客户端-服务器架构，支持实时 PVP、部落系统、部落战争、对战观战等多种游戏玩法。

[详细说明文档](#) | [快速开始](#) | [网络协议](#)

## 核心功能

### 游戏玩法

- **城镇建设系统** - 网格化地图，20+ 种建筑，升级系统
- **实时 PVP 对战** - 180 秒战斗，资源掠夺，奖杯系统
- **部落系统** - 创建部落，成员管理，权限控制
- **部落战争** - 多部落对战，24 小时持续，星数累计
- **对战观战** - 实时观看，同步操作，战斗录像
- **AI 防守** - 自动反击，目标优先级，伤害计算

### 网络架构

- **TCP Socket 通信** - 45 种消息类型，完整业务覆盖
- **匹配系统** - 基于奖杯的动态配对，等待补偿
- **多线程并发** - 单独线程/连接，Mutex 保护，1000+ 在线

## 项目结构

```
coc/
├── proj.win32/                      # Windows 工程
└── Classes/
    ├── Scenes/                         # 游戏场景
    ├── Managers/
        ├── SocketClient.h/cpp          # 网络客户端 ?
        ├── BattleManager.h/cpp         # 对战逻辑
        └── ...
    ├── Buildings/                      # 建筑系统
    └── UI/                             # 用户界面
└── Server/
    ├── Server.h                        # 服务器主类 ?
    ├── Server.cpp                     # 服务器实现
    └── ServerMain.cpp                 # 入口
└── engine/                           # Cocos2d-x 引擎
└── README.md                         # 本文件
└── 联网对战使用说明.md              # 详细文档
```

# 快速开始

## 环境要求

项目	要求
系统	Windows 7+ 或 Android 5.0+
编译器	Visual Studio 2015+ 或 NDK r21+
C++	C++14 标准
引擎	Cocos2d-x 3.17 或 4.x

## 编译和运行

### 1. 服务器

```
cd proj.win32
# 在 VS 中打开 Server.vcxproj
# Release 配置 → Build
./Server.exe
# 输出: Server started on port 8888
```

### 2. 客户端

```
cd proj.win32
# 在 VS 中打开 HelloCpp.vcxproj
# Debug 配置 → F5
```

### 3. 配置连接地址

```
// 修改客户端代码
SocketClient::getInstance()->connect("127.0.0.1", 8888); // 本地
// 或
SocketClient::getInstance()->connect("192.168.1.100", 8888); // 局域网
```

### 4. 运行

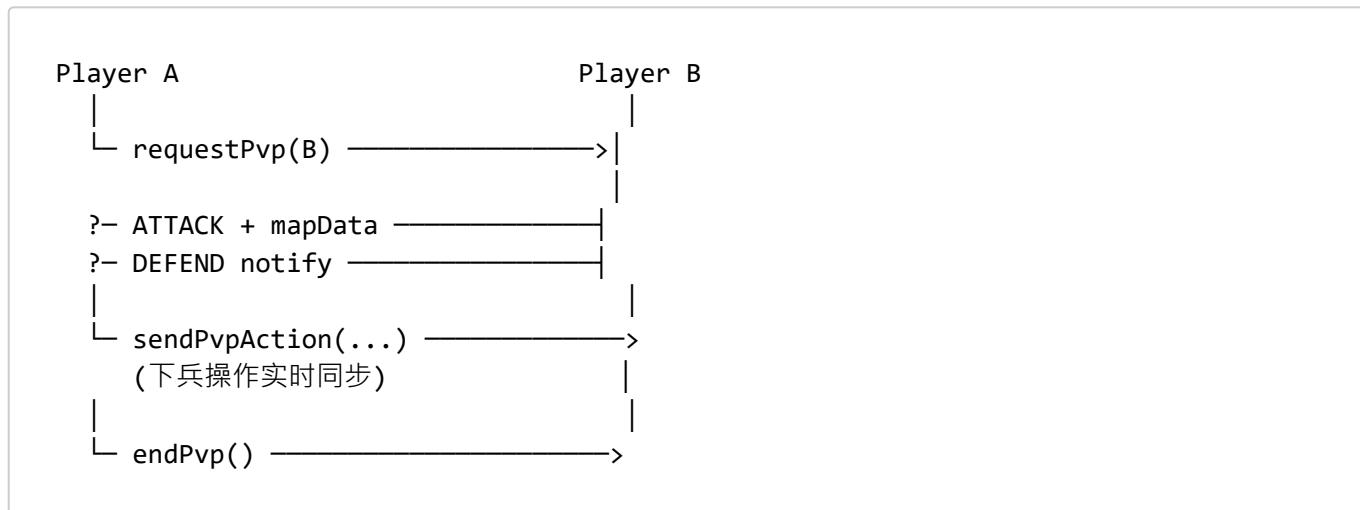
- 启动 2+ 个客户端 (不同玩家 ID)
- 登录游戏
- 发起对战

## 网络通信

## 消息类型汇总

类型	ID	说明
登录	1	玩家登录
地图操作	2-3	上传/查询地图
用户列表	5-6	获取可攻击玩家
匹配	10-12	请求/取消匹配
对战	13-15	开始/结束对战
部落	20-25	创建/加入/管理部落
部落战争	30-34	搜索/发起/结果
<b>PVP</b>	<b>40-45</b>	<b>** 实时对战系统**</b>

## PVP 流程



## 客户端 API

```
auto client = SocketClient::getInstance();
client->connect("127.0.0.1", 8888);
client->login("player_001", "MyName", 500);

// 获取玩家列表
client->requestUserList();
client->setOnUserListReceived([](const std::string& list) {
    // 显示玩家列表
});

// 发起 PVP
client->requestPvp("opponent_id");
client->setOnPvpStart([](const std::string& role,
                      const std::string& opId,
                      const std::string& mapData) {
    // 进入对战
});
```

```

});  

// 下兵操作  

client->sendPvpAction(UNIT_BARBARIAN, 100.0f, 200.0f);  

// 对战结束  

client->endPvp();  

// 在 update 中处理回调  

void update(float dt) {  

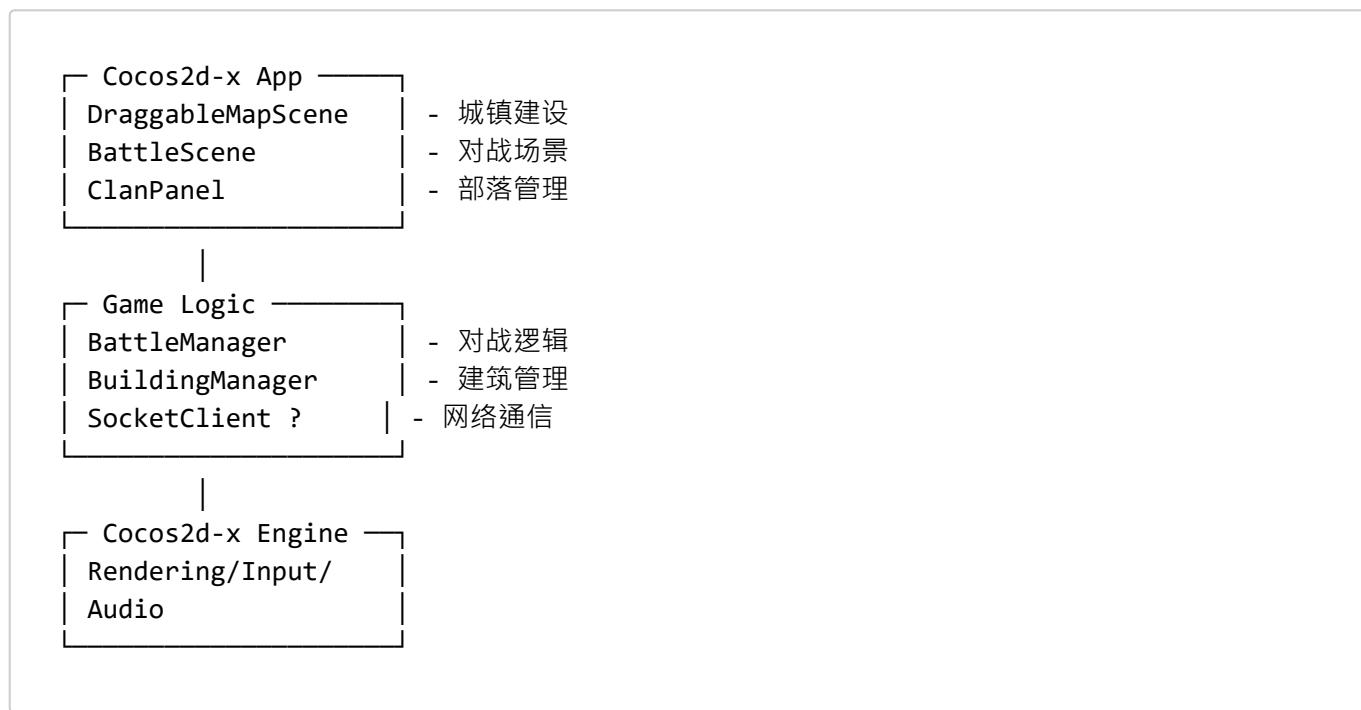
    client->processCallbacks();  

}

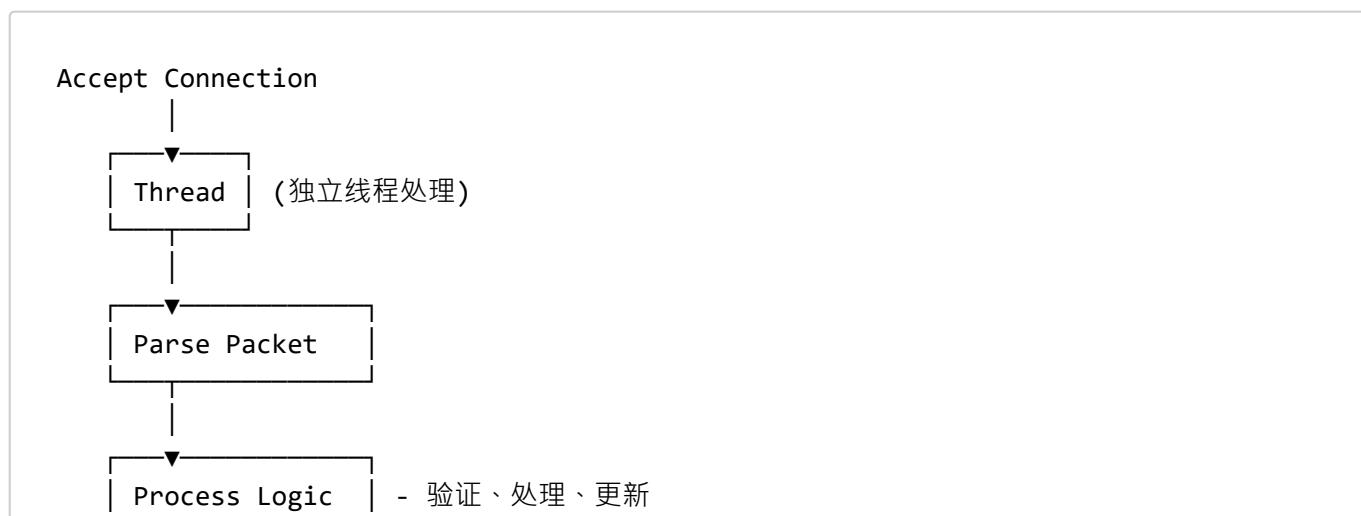
```

## ? 系统架构

### 客户端层级



### 服务器处理流程





## 功能详解

### 实时 PVP 对战

**特点：**

- ? 低延迟 ( 50-100ms)
- 实时同步单位操作
- 即时 AI 反击
- 可被观战

**流程：**

1. 请求 PVP → 验证在线 → 获取地图
2. 显示地图 → 玩家下兵 → 实时转发
3. AI 自动反击 → 伤害计算 → 资源更新
4. 180 秒结束 → 计算结果 → 奖杯/资源变化

### 部落战争

**特点：**

- ? 多个部落参与
- 24 小时持续
- ? 星数累计制
- 实时状态更新

**阶段：**

1. 搜索 → 匹配相近奖杯的部落
2. 确认 → 通知两个部落成员
3. 对战 → 成员选择对手 · 发起攻击
4. 统计 → 每次攻击上报星数 · 实时更新
5. 结算 → 24h 后自动结算

### 对战观战

**功能：**

- 实时观看两名玩家对战
- 同步显示双方操作
- 支持回放

使用：

```
client->requestSpectate("player_id");
client->setOnSpectateJoin([](bool ok, const std::string& atkId,
                           const std::string& defId,
                           const std::string& mapData) {
    if (ok) {
        // 进入观战模式
    }
});
```