

# Это вам не игрушки: портирование кросс- платформенной C++ игры на iOS

Андрей Филипенков

# HEROES III OF MIGHT AND MAGIC® Complete



# План

Начало: зависимости

Архитектура VCMI и iOS

Адаптация CMake проекта под Xcode

Android: зависимости в Conan

Поиграем 😊

# Начало: зависимости

# Наивная попытка сборки

```
cmake -S . -B ../build \  
-D CMAKE_SYSTEM_NAME=iOS
```

Could NOT find Boost (missing:  
Boost\_INCLUDE\_DIR  
program\_options) (Required is at  
least version "1.73.0").

# Зависимости проекта

## **Сервер**

Boost

## **Клиент**

- SDL + Image, Mixer, TTF

- FFmpeg

## **Лаунчер**

Qt (widgets)

## **ИИ**

Intel oneTBB

# Зависимости проекта

<https://github.com/vcmi/vcmi-ios-deps>

Зависимости  
получше:  
менеджер пакетов





**vcpkg**

**Conan**

**CMake  
FetchContent**

# Conan the Frogarian



# My Dudes

## Build profile

```
[settings]
```

```
os=Macos
```

```
arch=x86_64
```

```
compiler=apple-clang
```

```
compiler.version=14
```

```
compiler.libcxx=libc++
```

```
build_type=Release
```

## Host profile

11

```
[settings]
```

```
os=iOS
```

```
os.sdk=iphonios
```

```
os.version=12.0
```

```
arch=armv8
```

```
compiler=apple-clang
```

```
compiler.version=14
```

```
compiler.libcxx=libc++
```

```
build_type=Release
```

# Справочные материалы Conan

## Документация

- [v2 \(текущая\)](#)
- [v1](#)

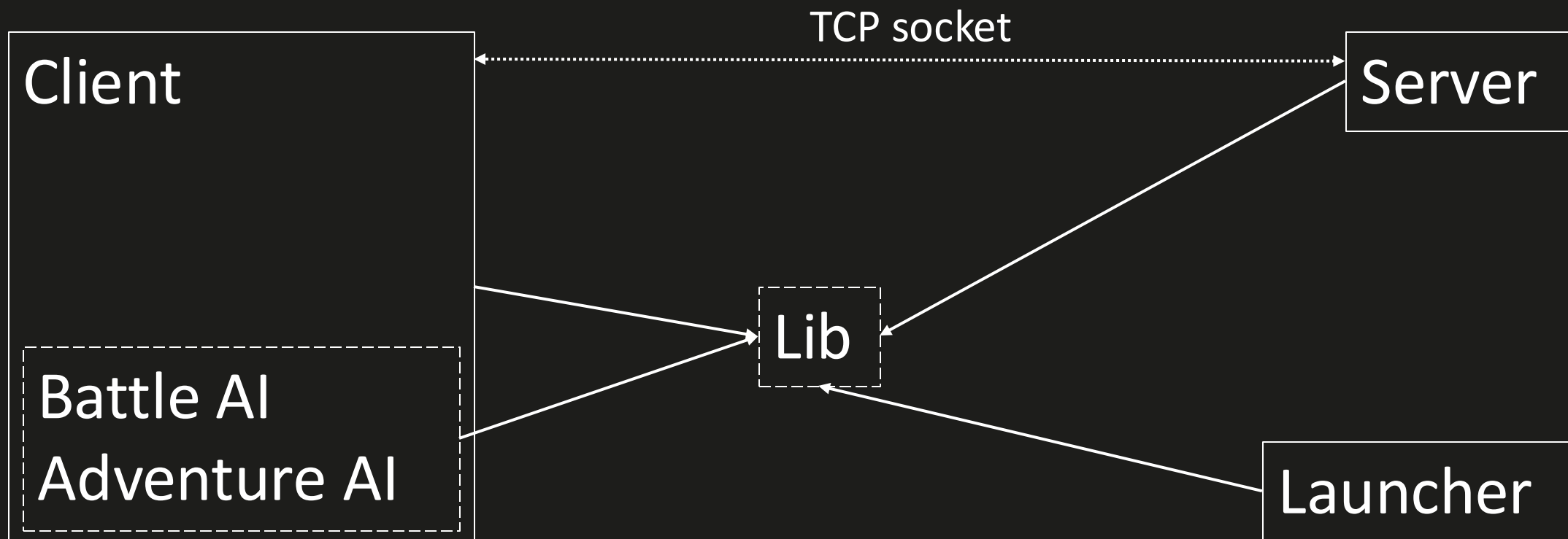
## Пулл реквесты в CCI

[Для «рецептов» под macOS / iOS / Android.](#)

## Проект VSMI

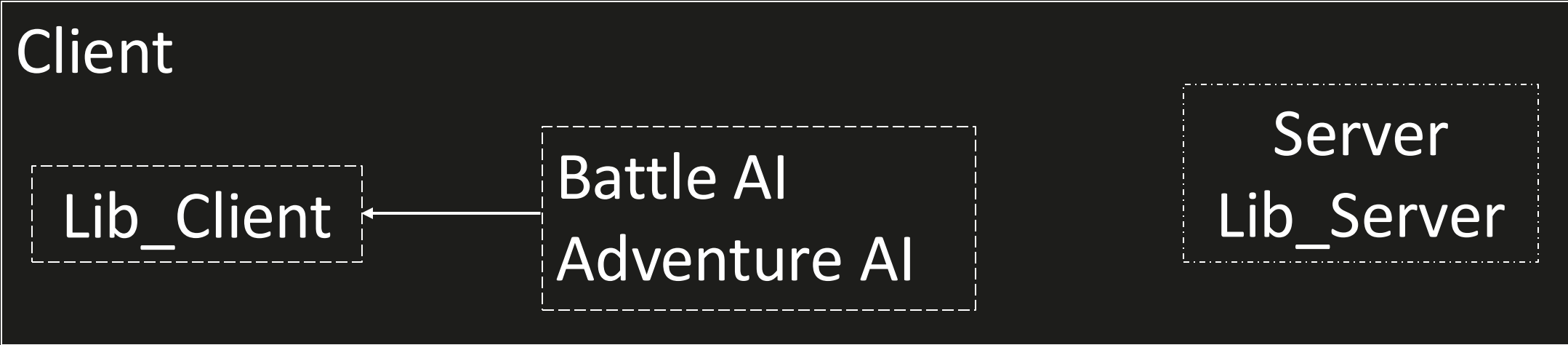
- [пулл реквест для macOS](#)
- [пулл реквест для iOS](#)
- [наша документация](#)
- [conanfile.py](#)

# Архитектура VSM и iOS



GAME

Start  
Server





# C++ реализация

```
Lib_Client  
namespace Client {  
    Lib  
}
```

```
Lib_Server  
namespace Server {  
    Lib  
}
```

```
Использование  
using namespace *
```

## Объявление макросов

```
#ifndef NAMESPACE
# define NAMESPACE_BEGIN namespace NAMESPACE {
# define NAMESPACE_END }
# define USING_NAMESPACE using namespace
NAMESPACE;
# define WRAP_NAMESPACE(x) NAMESPACE::x
#else
# define NAMESPACE_BEGIN
# define NAMESPACE_END
# define USING_NAMESPACE
# define WRAP_NAMESPACE(x) x
#endif
```

## Использование в общей библиотеке

18

```
lib/CBonusTypeHandler.h

@@ -14,6 +14,8 @@
#include "IHandlerBase.h"
#include "HeroBonus.h"

+ VCMILIB_NAMESPACE_BEGIN
+
class JsonNode;

@@ -94,6 +96,8 @@ class DLL_LINKAGE CBonusTypeHandler : public IBonusType
    std::vector<CBonusType> bonusTypes; //index = BonusTypeID
};

+ VCMILIB_NAMESPACE_END
+
#ifndef INSTANTIATE_CBonusTypeHandler_HERE
- extern template class std::vector<CBonusType>;
+ extern template class std::vector<VCMILIB_WRAP_NAMESPACE(CBonusType)>;
#endif
```

# iOS platform support #776

Edit

&lt;&gt; Code ▾

Merged nullkiller merged 129 commits into `vcmi:develop` from `kambala-decapitator:ios-PR` on Sep 24, 2022

Conversation 60

Commits 129

Checks 0

Files changed 627

+4,611 -872 

## Commits

129

## Files changed

627-420=207

## Diff

4611-1925=2686

872-209=663

## Длительность

Около 9 месяцев

<https://github.com/vcmi/vcmi/pull/776>

# Адаптация CMake проекта под Xcode

```
if(APPLE)
  if(CMAKE_SYSTEM_NAME STREQUAL "Darwin")
    set(APPLE_MACOS 1)
  else()
    set(APPLE_IOS 1)
  endif()
endif()
```

**Использование**

```
if(APPLE_IOS)
  ...
endif()
```

**CMake 3.14+**

```
if(IOS)
  ...
endif()
```

```
set(CMAKE_XCODE_ATTRIBUTE_CLANG_ENABLE_OBJC_ARC YES)
set(CMAKE_XCODE_ATTRIBUTE_DEBUG_INFORMATION_FORMAT[variant=Debug] dwarf)
set(CMAKE_XCODE_ATTRIBUTE_ENABLE_BITCODE NO)
set(CMAKE_XCODE_ATTRIBUTE_ENABLE_NS_ASSERTIONS NO)
set(CMAKE_XCODE_ATTRIBUTE_ENABLE_NS_ASSERTIONS[variant=Debug] YES)
set(CMAKE_XCODE_ATTRIBUTE_GCC_WARN_64_TO_32_BIT_CONVERSION NO)
set(CMAKE_XCODE_ATTRIBUTE_MARKETING_VERSION ${PROJECT_VERSION})
set(CMAKE_XCODE_ATTRIBUTE_ONLY_ACTIVE_ARCH NO)
set(CMAKE_XCODE_ATTRIBUTE_ONLY_ACTIVE_ARCH[variant=Debug] YES)
set(CMAKE_XCODE_ATTRIBUTE_TARGETED_DEVICE_FAMILY "1,2")
```

```
set(CMAKE_XCODE_ATTRIBUTE_DEBUG_INFORMATION_FORMAT[variant=Debug] dwarf)
```

23

The screenshot shows the Xcode interface for the 'test-cpp.xcodeproj' project. The 'Build Settings' tab is active, and the 'Debug Information Format' setting is expanded. The 'Debug' configuration is selected, and the value is set to 'DWARF'. The 'Release' configuration is set to 'DWARF with dSYM File'. The 'Levels' dropdown is set to 'dwarf'.

Setting	test-cpp
Debug Information Format	<Multiple values> ⇅
Debug	DWARF ⇅
Release	DWARF with dSYM File ⇅

The Quick Help panel provides details for the 'DEBUG\_INFORMATION\_FORMAT' setting. It includes a summary, declaration, value type, and discussion.

**Quick Help**

**Summary**  
The type of debug information to produce.

**Declaration**  
DEBUG\_INFORMATION\_FORMAT

**Value Type**  
Enumeration (String)

**Discussion**

- *DWARF*: Object files and linked products will use DWARF as the debug information format. [dwarf]
- *DWARF with dSYM File*: Object files and linked products will use DWARF as the debug information format, and Xcode will also produce a dSYM file containing the debug information from the individual object files (except that a dSYM file is not needed and will not be created for static library or object file products). [dwarf-with-dsym]

# Настройки Xcode в CMake

## Уровень проекта

```
set(CMAKE_XCODE_ATTRIBUTE_<SETTING> <VALUE>)
```

## Уровень таргета

```
set_target_properties(  
<TARGET> PROPERTIES  
XCODE_ATTRIBUTE_<SETTING>  
<VALUE>)
```

## Синтаксис xcconfig

[The Unofficial Guide to xcconfig files](#)



# Важные настройки для Apple платформ

**CMAKE\_MACOSX\_RPATH**

```
set(CMAKE_MACOSX_RPATH 1)
```

**CMAKE\_OSX\_DEPLOYMENT\_TARGET**

```
set(CMAKE_OSX_DEPLOYMENT_TARGET 12.0)
```

**CMAKE\_OSX\_ARCHITECTURES**

```
cmake ... -DCMAKE_OSX_ARCHITECTURES  
="armv7;arm64"
```

# Формирование пакета приложения .app

Стандартный механизм

cmake --install ...

+

add\_custom\_command

(либо add\_custom\_target)

```
add_custom_command(TARGET vcmiclient POST_BUILD
  COMMAND ${CMAKE_COMMAND} --install "${CMAKE_BINARY_DIR}" \
  --component "${CMAKE_INSTALL_DEFAULT_COMPONENT_NAME}" \
  --config "$<CONFIG>" \
  --prefix "$<TARGET_BUNDLE_CONTENT_DIR:vcmiclient>"
)
```

# Подпись кода

**Таргет(ы) приложения**

Включить

CODE\_SIGNING\_ALLOWED

**Остальные таргеты**

Выключить (поведение по умолчанию), подписываются вручную

# Формирование ipa файла

Стандартный механизм  
CPack

```
set(CPACK_GENERATOR ZIP)  
set(CPACK_ARCHIVE_OUTPUT_EXTENSION ipa)
```

```
set(CPACK_INCLUDE_TOPLEVEL_DIRECTORY OFF)  
set(CPACK_INSTALL_CMAKE_PROJECTS  
"${CMAKE_CURRENT_BINARY_DIR};${CMAKE_PROJECT_NAME};app;/")
```

```
install(TARGETS vcmiclient DESTINATION Payload COMPONENT app)
```

# Зависимости для Android: Conan

```
[settings]  
os=iOS  
os.sdk=iphonesos  
os.version=12.0  
arch=armv8  
compiler=apple-clang  
compiler.version=14  
compiler.libcxx=libc++  
build_type=Release
```

```
[settings]  
os=Android  
os.api_level=21  
  
arch=armv8  
compiler=clang  
compiler.version=14  
compiler.libcxx=c++_shared  
build_type=Release
```

# Демонстрация

[https://vk.com/video1645302\\_456240183](https://vk.com/video1645302_456240183)

# Итоги

- Узнали как собирать C++ зависимости проекта под мобильную ОС
- Научились собирать зависимости через Conan вместо своих скриптов
- Узнали как адаптировать CMake проект под iOS
- Можем играть в Героев 3 на мобильных устройствах 😊



# VCMi GitHub

iOS issues:

- [Главная репа](#)
- [Мой форк](#)



[Android issues](#)

<https://github.com/vcmi/vcmi>

# Спасибо за внимание!

Андрей Филипенков

Senior iOS dev

@kambala\_decapitator

kaspersky