

# 武汉大学计算机学院 实验报告

## 目标系统任务书

专 业 名 称：软件工程

课 程 名 称：高级软件工程

指 导 教 师：何璐璐

团 队 成 员：丛兴柏 卢思源 肖宇航

# 目 录

1 目标项目名称 .....	3
2 基本情况介绍 .....	4
3 目标和要求 .....	5
3.1 分析测试并修复漏洞，提高代码质量 .....	5
3.2 维护和完善已有功能 .....	5
3.3 新功能开发 .....	5
4 项目背景 .....	7
4.1 行业背景和现状的调研 .....	7
4.1.1 背景 .....	7
4.1.2 行业现状 .....	7
4.2 行业未来发展趋势 .....	8
5 特征和影响 .....	8
5.1 系统功能 .....	8
5.2 系统技术 .....	9
5.3 系统运行环境 .....	10
5.4 系统使用方法 .....	10
5.5 应用场景 .....	11
5.6 应用场景实例 .....	12
5.6.1 主界面 .....	12
5.6.2 设置 .....	15
5.6.3 直播来源 .....	18
5.6.4 滤镜 .....	19
5.7 系统优缺点 .....	22
5.8 创新点 .....	23
5.8.1 现有系统创意点 .....	23
5.8.2 新功能创意点 .....	24
6 人员分工 .....	24
6.1 项目目标任务书人员分工 .....	24
6.2 项目具体实现 .....	24
7 验收建议 .....	24
7.1 验收环境 .....	24
7.2 验收方法 .....	24
7.3 验收流程 .....	25
7.3.1 确认设计目标是否被完成 .....	25
7.3.2 系统测试验收 .....	25
7.4 验收内容 .....	26
7.4.1 界面验收建议 .....	26
7.4.2 功能验收建议 .....	26
7.4.3 性能验收建议 .....	26
7.4.4 代码质量验收建议 .....	28
7.4.5 安全验收建议 .....	28
8 参考文献 .....	30

# 1 目标项目名称

目标系统的名称是 OBS-Studio，该系统是一个直播推流与视频录制软件。

## 2 基本情况介绍

OBS-Studio 是一款免费且开源的视频录制和直播软件，它可以方便地录制游戏、屏幕、网络摄像头或其他来源，图 2.1 是 OBS 的软件图标。该软件支持向各类流行平台进行直播，例如 Twitch，YouTube，Facebook，以及国内的虎牙、斗鱼等。该软件的源代码在 github 上的链接是：

<https://github.com/obsproject/obs-studio>



图 2.1 OBS 图标

经过代码分析工具的统计（如 2.2 所示），OBS-Studio 拥有 10,000 行左右的代码量，代码量适中，在 github 上星标数达到 50.7K（如 2.3 所示），关注与引用量巨大，同时 OBS 工作室也是当前最流行的直播与视频录制软件，综上，OBS-Studio 合适作为目标系统的原型。

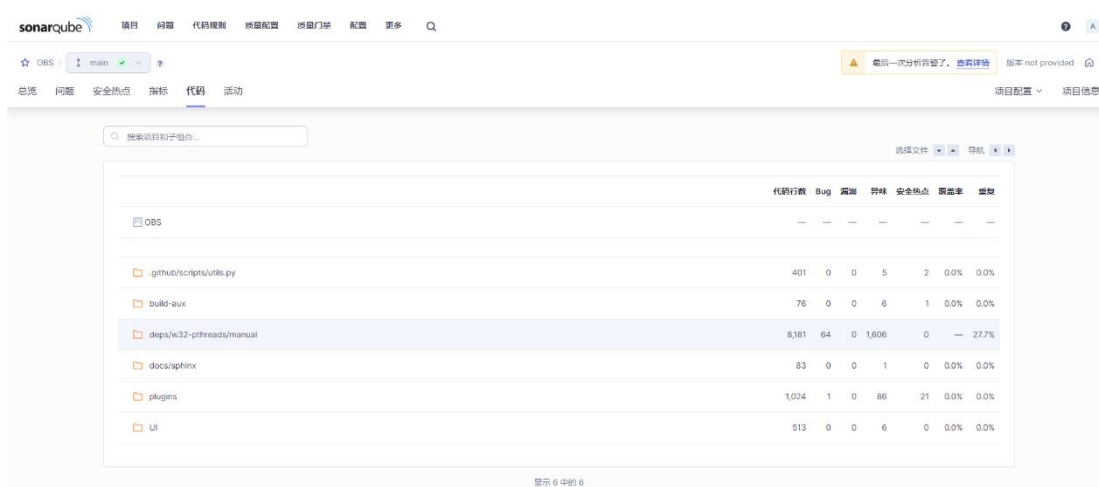


图 2.2 OBS-Studio 源码代码行数统计

Readme  
GPL-2.0 license  
Activity  
50.7k stars  
1.4k watching  
7.2k forks  
Report repository

图 2.3 OBS 的 github 仓库的关注度

## 3 目标和要求

### 3.1 分析测试并修复漏洞，提高代码质量

使用 SonarQube 自动代码审查工具对 OBS-Studio 源代码进行分析，检测代码中的错误、漏洞以及异常代码，如图 3 所示，并分别修改，以提高代码的质量和程序的可靠性。

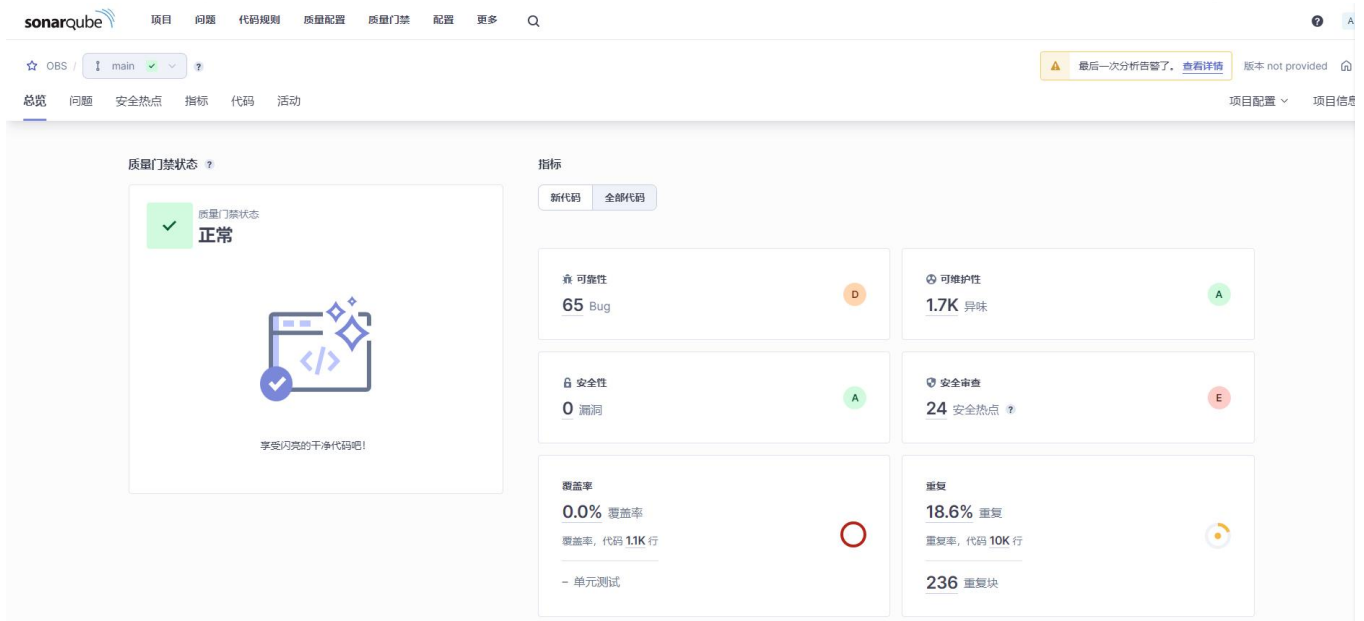


图 3 OBS-Studio 源代码分析结果

可以看到 OBS-Studio 系统的质量门禁状态是正常，说明其内建质量达到标准。具体指标而言，SonarQube 给出的评估等级是：可靠性 D，可维护性 A，安全性 A 和安全审查 E。代码存在着一定的 Bug，可以借助 SonarQube 进行修改。

### 3.2 维护和完善已有功能

团队将继续维护和完善 OBS-Studio 以下几个方面的具体功能：

1. 视频录制功能：检查编码器的稳定性和与不同硬件的兼容性，如确保在新版的显卡上 NVENC 编码器能正常工作；增加新的录制格式或编码选项。例如，当新的视频编码标准发布时（如 AV1），考虑集成对其的支持。

2. 音频混音器：定期测试以确保多个音频来源能够无缝混合，并在各种操作系统上都能正常工作；增加更多的音频效果或混音选项。例如，引入一个新的音效或混响选项，使声音更富有空间感。

### 3.3 新功能开发

在后续的开发过程中，考虑 OBS-Studio 将在原有基础上继续添加可能实现的新功能：

1. 实时 AI 助手：利用 AI 分析直播的内容，自动调整相机焦点、切换场景或提供内容建议；实时语音转文字功能，为直播自动提供字幕。

2. 集成的视频编辑工具：在录制后直接在 OBS 中进行基本的视频编辑，如剪切、添加文字、转场效果等，允许用户直接从 OBS 上传编辑后的视频到 YouTube 或其他平台。

3. 集成的教程和建议系统：对于初学者，提供内置的指导和教程；根据用户的配置和直播内容，提供优化和设置建议。

## 4 项目背景

### 4.1 行业背景和现状的调研

#### 4.1.1 背景

随着在线视频内容的需求不断增加，媒体流媒体行业正在迅速增长。许多公司、内容创作者和个人都希望能够轻松地进行直播和流媒体传输。这导致了对直播推流软件的需求激增。游戏直播已经成为在线娱乐的主要形式之一。像 Twitch 这样的平台允许玩家和内容创作者分享他们的游戏体验。这推动了对直播工具的需求，包括直播推流软件；社交媒体平台如 YouTube、Facebook 和 Instagram 等也推动了对直播工具的需求。内容创作者使用这些工具来与他们的观众进行互动，分享日常生活或创作内容。

#### 4.1.2 行业现状

OBS Studio 是开源和免费的，这使得很多个人和小型创作者更容易入门。其开放性也意味着社区可以及时为其提供补丁和新增功能。由于其开源性质，很多开发者创建了各种插件来增强 OBS 的功能。然而，尽管 OBS Studio 非常流行，但市场上还有其他推流软件，如 Streamlabs OBS（基于 OBS，但具有额外的集成和功能）、XSplit 和 vMix 等。

随着 Twitch、YouTube Live、Facebook Live 等平台的兴起，直播已成为一种受欢迎的内容形式，推动了推流软件的需求。而随着游戏和其他内容的质量不断提高，直播者和创作者需要更强大的硬件和软件来满足 1080p、4K 甚至更高分辨率的直播需求。一些直播解决方案开始使用云技术来提供更稳定的流、增强的编码功能或分布式的直播功能。随着直播变得更加流行，新的软件工具和解决方案致力于使直播变得更简单，以吸引更多的非技术型用户。

尽管直播推流软件种类众多，但这类软件仍存在一些问题：

1. 复杂性：尽管一些推流软件试图保持用户友好，但配置和优化设置仍然可能对初学者来说很复杂。
2. 硬件要求：高清直播或游戏直播需要强大的计算和图形处理能力，这可能会导致对于低端系统来说过于消耗资源。
3. 编码问题：选择合适的编码设置可能会很复杂，错误的设置可能导致视频质量下降或者直播中断。
4. 与其他软件的兼容性：有时，推流软件可能与某些游戏或应用程序不兼容，导致性能问题或崩溃。
5. 插件的质量和兼容性：虽然插件可以增强软件功能，但第三方开发的插件可

能导致稳定性问题或与其他插件发生冲突。

6. 音频同步：在某些情况下，音频和视频之间可能会出现同步问题，导致延迟或错位。

## 4.2 行业未来发展趋势

直播和推流软件行业仍在迅速发展，未来的趋势受到技术进步、消费者需求和市场变化的影响：

1. 更简单的用户体验：为了吸引更多的非专业用户，推流软件可能会变得更加用户友好，减少设置的复杂性。

2. 云计算的增长：随着云计算的普及，更多的直播功能，如编码、存储和分发，可能会转移到云端，从而减少本地资源的消耗。

3. 增强的交互性：随着技术的进步，直播可能会变得更加交互性，包括虚拟现实、增强现实和其他形式的观众参与。

4. 更高的质量和分辨率：随着技术的发展，4K、8K 甚至更高分辨率的直播将成为可能，并且流行起来。

5. AI 和自动化：人工智能技术可能会用于直播中，例如自动切换镜头、实时内容分析、自动标记和内容推荐。

6. 更强大的分析工具：为了帮助直播者更好地理解他们的观众，推流软件可能会提供更详细和深入的分析工具。

7. 隐私和安全的重视：随着隐私和安全问题成为热门话题，推流软件可能会加强相关功能，以保护创作者和观众的数据。

8. 集成的社交功能：为了增强社区和观众的互动，推流软件可能会更加深入地集成社交功能和工具。

9. 更丰富的内容创作工具：除了基本的推流功能，未来的软件可能会提供更多的创作工具，如图形编辑、动画和其他视觉效果。

## 5 特征和影响

### 5.1 系统功能

OBS-Studio 是一款免费且开源的视频录制和直播软件，主要拥有以下功能：

高性能实时视频/音频捕获和混合。创建由多个源组成的场景，包括窗口捕获、图像、文本、浏览器窗口、网络摄像头、捕获卡等

设置无限数量的场景，可以通过自定义过渡无缝切换。

直观的混音器，具有噪声门、噪声抑制和增益等每源滤波器。通过 VST 插件支



持进行完全控制。

功能强大且易于使用的配置选项。添加新源，复制现有源并调整其属性。

简化的设置面板可以访问各种配置选项，以调整直播或录制的各个方面。

模块化的“停靠”UI 允许重新排列布局。可以将每个单独的 Dock 弹出到其自己的窗口。

## 5.2 系统技术

1. 图形 API: Direct3D: 在 Windows 上, OBS 使用 Direct3D 11 进行视频渲染。OpenGL: OBS 在 macOS 和 Linux 上使用 OpenGL。

2. 音频处理: libspeexdsp: 用于声音的回声消除和其他音频处理功能。

3. 视频编码/解码: FFmpeg: 一个非常强大的库, 用于处理音频和视频数据。OBS 使用 FFmpeg 进行各种编码和格式转换操作。x264: 一个流行的开源 H.264 编码器。NVENC 和 AMD AMF: 这两个是 NVIDIA 和 AMD 的硬件编码接口, OBS 支持它们以允许硬件加速编码。

4. 媒体源和设备捕捉: libVLC: OBS 可以利用 VLC 的功能来播放各种媒体格式。DirectShow: 在 Windows 上用于设备捕捉。

5. 窗口和界面: Qt: OBS 的用户界面是使用 Qt 框架构建的, 它是一个跨平台的 C++ 图形用户界面框架。

6. 浏览器集成: CEF (Chromium Embedded Framework): OBS 使用它来集成浏览器功能, 允许用户将 web 内容嵌入到他们的流或录像中。

7. 通信和流: WebSockets: 通过 OBS-WebSocket 插件, OBS 可以支持 WebSockets, 允许远程控制和管理。RTMP (Real-Time Messaging Protocol): OBS 使用 RTMP 进行直播推流。

8. 其他库和技术: Jansson: 一个 C 语言的 JSON 库, 用于数据解析和生成。mbedtls: 用于加密和 SSL 功能。zlib: 用于压缩。

## 5.3 系统运行环境

OBS 系统推荐运行在 Windows10、MacOS 以及 Linux 操作系统上。下面是运行 OBS 工作室所要求的系统软硬件支持：

Windows:

DirectX 10.1 compatible GPU

Windows 10, or Windows 11

macOS:

Intel or Apple Silicon CPU

OpenGL 3.3-compatible GPU

macOS 11 ("Big Sur") or later

Linux/Unix:

OpenGL 3.3-compatible GPU

X window system or Waylan

## 5.4 系统使用方法

1. 通过直接下载安装官网提供的安装包并执行安装，下载链接如下：

<https://obsproject.com/download>

2. 使用官方文档中给出的指令安装：

以 Windows10 系统为例：

可以使用三个不同的包管理工具安装

winget:

```
winget install -e --id OBSProject.OBSStudio
```

Chocolatey:

```
choco install obs-studio
```

Scoop:

```
scoop bucket add extras scoop install obs-studio
```

3. 通过开源的源码构建，流程如下：

先决条件：

- Windows 10 1909+ (或 Windows 11)
- Visual Studio 2022 (至少是社区版)
- Windows 10 开发工具包 (最低 10.0.20348.0)

- C++ ATL 用于最新的 v143 构建工具（x86 和 x64）
- Git for Windows
- CMake 3.24 或更高版本

配置构建项目：

- 克隆包含子模块的存储库：

```
git clone --recursive https://github.com/obsproject/obs-studio.git
```

- 将当前目录设置为 obs-studio
- 检查可用的 CMake 预设：cmake --list-presets
- 选择预设：windows-x64cmake --preset windows-x64

可用和支持的体系结构包括：x64

不再支持 OBS 工作室的 x86（32 位）版本

任何其他 CMake 变量都可以照常提供，如有必要，也可以覆盖预设设置的变量  
编译 obs-studio：

- 在生成的生成目录中打开 Visual Studio 解决方案文件（build\_x64\obs-studio.sln）
- 选择要构建的构建配置（Debug、MinSizeRel、Release、RelWithDebInfo）
- 按<Control>+<Shift>+<B>以生成解决方案（生成 ->生成解决方案）或者，按 <F5>生成并运行它（调试 -> 开始调试）

或者，也可以在命令行上构建项目：

- 确保当前目录设置为源代码目录（如果您从上面的“配置构建项目”继续，您已经在那里）obs-studio

- 运行 cmake --build --preset windows-x64
- 通过以下方式运行应

用./rundir/<Debug|Release|RelWithDebInfo|MinSizeRel>/bin/64bit/obs64.exe

## 5.5 应用场景

OBS-Studio 是一款免费、开源的跨平台直播流媒体软件，具有广泛的应用场景，适用于不同领域的用户，包括以下几个方面：

1. 游戏直播： OBS Studio 最常见的用途之一是游戏直播。游戏玩家可以使用 OBS Studio 捕获他们的游戏画面和音频，并将其实时传输到流媒体平台，如 Twitch、YouTube Gaming 和 Facebook Gaming，与观众互动。

2. 教育和在线培训：教育工作者和在线培训师可以使用 OBS Studio 创建高质量的在线教育内容。他们可以捕获自己的摄像头、屏幕、演示文稿等，并将这些内容流式传输给学生或观众。

3. 娱乐和创意内容创作：创意性内容创作者可以使用 OBS Studio 制作各种内容，如 Podcast、YouTube 视频、娱乐节目等。他们可以添加特效、实时图层叠加以及音频处理，以创造有趣的内容。

4. 活动直播：OBS Studio 广泛用于直播各种活动，包括音乐会、体育赛事、会议和婚礼。它可以帮助将这些事件实时传送到在线观众，无论观众身在何处。

5. 音乐演出和演出活动：音乐家、演员和表演者可以使用 OBS Studio 将他们的音乐会、演出和表演实时传输给全球观众。这有助于扩大他们的观众群体和增加曝光度。

6. 社交媒体直播：用户可以将 OBS Studio 与社交媒体平台（如 Facebook 和 YouTube）集成，以实时分享自己的生活、活动和见解。

## 5.6 应用场景实例

### 5.6.1 主界面



OBS 界面可以分为菜单栏、预览区、主按钮区和运行状态四个区块。日常使用中，只需要接触主按钮区，包括场景、来源、混音器、转场特效和控制按钮。

- 场景



场景（Scenes）是指直播中不同的场面。不同的场景可以用于不同的领域或场合，例如自习室场景、街拍场景、个人访问、卖货直播。

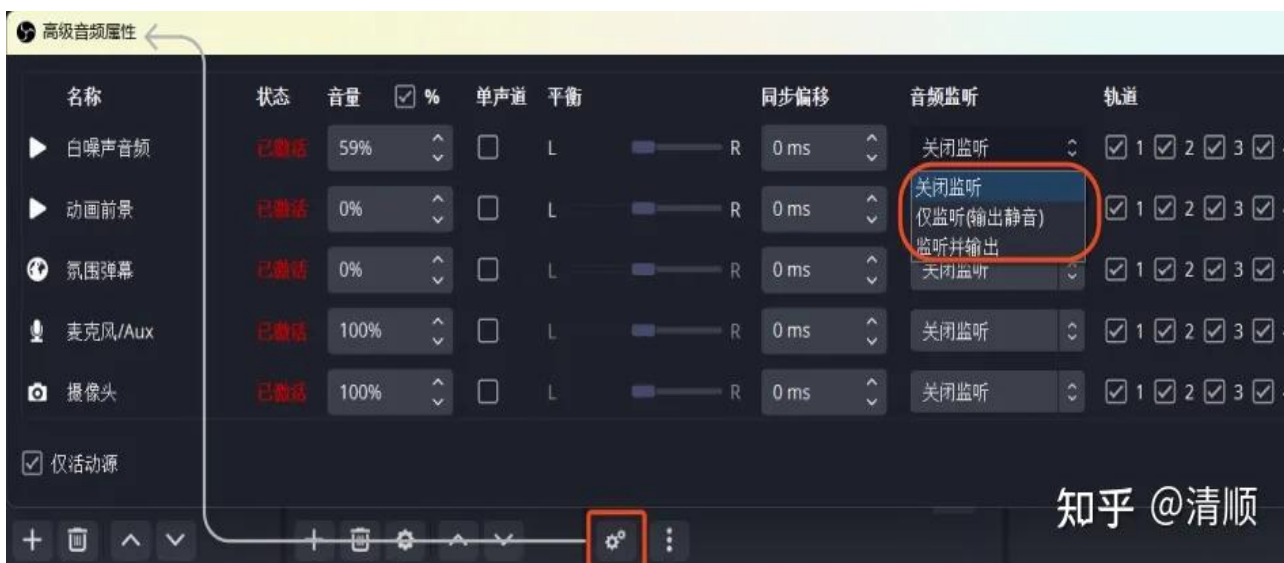
场景有四个按钮，从左到右依次为：添加场景，删除场景，向上移动场景，向下移动场景。

- 来源



来源（Source）是指场景中的素材，比如摄像头画面、图像、音频等。来源区有 7 个按钮，上方两个为源属性和滤镜，下方 5 个按钮依次为：添加来源，删除来源，源属性，向上移动源，向下移动源。

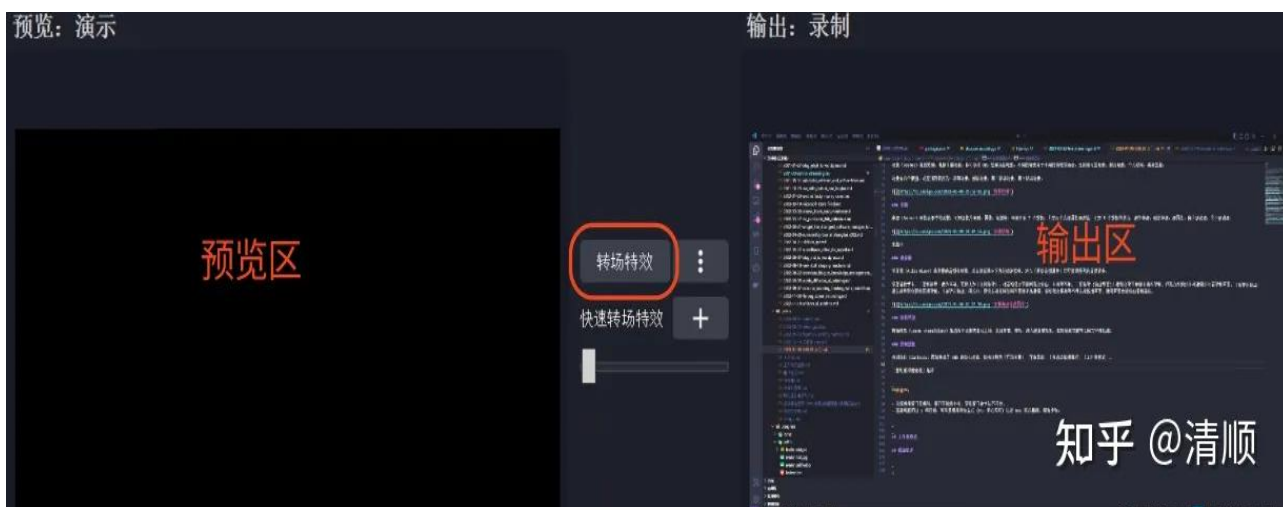
## • 混音器



混音器（Audio Mixer）是直播的音频控制器。点击混音器左下角的齿轮按钮，进入「高级音频属性」，可管理所有的音频设备。

混音器设置中，「音频监听」最为重要。默认为「关闭监听」，指音频仅对直播间观众播放，主播听不到。「仅监听 (输出静音)」表示观众听不到，但主播能听到，观众可通过主播话筒的收音听到声音。「监听并输出」表示主播和观众都能听到。「监听并输出」模式中，建议主播使用耳麦，而不是话筒，否则观众容易听不清主播说话的声音，话筒声音会被输出音频盖住。

## • 转场特效



转场特效（Scene Transitions）指的是在两个场景或素材之间，采用渐变、移动、滑入滑出等技巧，实现场景或情节之间平滑过渡的效果。

- **控制按钮**

控制按钮（Controls）区域集成了 OBS 的核心功能，按钮分别为「开始直播」「开始录制」「启动虚拟摄像机」「工作室模式」。

「启动虚拟摄像机」是将 OBS 场景作为摄像头画面对外输出，适合将画面输出到不支持 OBS 推流的平台或软件。例如，抖音直播要求粉丝在一万以上才能使用 OBS 推流，而启用虚拟摄像头后，将直播摄像头设为「OBS-Camera」即可达到类似推流效果。

「工作室模式」下，显示区域会分为两块：左侧为预览区，右侧为输出区。点击中间按钮「转场特效」，即可将左侧预览区输出到直播画面。

### 5.6.2 设置

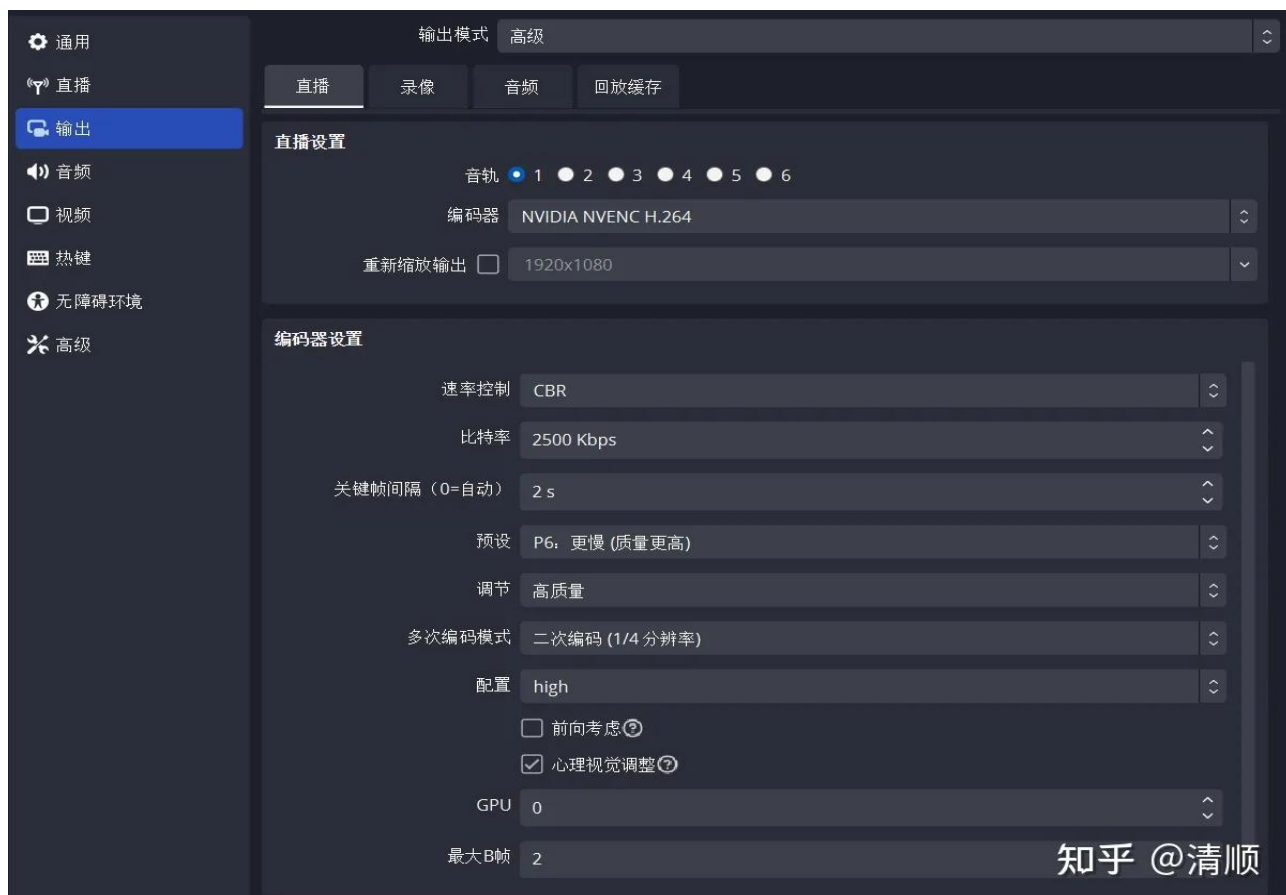
在使用 OBS 直播前，请先点击右上角菜单栏中的「文件」>「设置」，调整 OBS 的设置。



- **直播**

点击「设置」>「直播」，将「服务」调整为自定义，服务器和推流码则按平台要求输入。

## • 输出



点击「设置」>「输出」，将「输出模式」设为高级。

「直播设置」>「编码器」中，如果显卡配置较好，尽量选硬件编码，如NVIDIA NVENC H.264。

「编码器设置」>「比特率」用来设置直播的清晰度，推荐 4000–6000 Kbps。  
注意：直播并非越清晰越好，它对网络稳定性要求较高，过高的清晰度可能导致直播卡顿。

## • 音频

点击「设置」>「音频」>「通用」，将声道设为单声道。除非你是歌唱主播，否则无需使用立体声。



- 视频



点击「设置」>「视频」，按需设置直播分辨率，通常为 1920x1080 或 1280x720。如果你是做电商直播，需要进行竖屏直播，就将分辨率反过来设置，比如 1080x1920 或 720x1280。「输出（画布）分辨率」与「基础（缩放）分辨率」建议设为相同数值，避免出现缩放问题。

「常用帧率」在日常直播中只需要 30，游戏直播建议调整到 60。

- 高级



点击「设置」>「高级」，将「进程优先级」设为高，以降低电脑卡顿的概率。然后，启用「直播延迟」和「自动重连」功能，以避免直播事故的发生。「直播延迟」启用后，观众看到的直播画面会比实时要慢，延迟时间一般为 3~20 秒。

### 5.6.3直播来源



直播来源种类繁多，可以分为四类：静态来源、音频来源、视频来源和组合来

源。

- **静态来源**

- 文本（GDI+）：添加文字，并可设置背景和移动效果。
- 色源：添加指定色彩的纯色画面素材。
- 图像：添加静态图像素材。
- 图像幻灯片放映：以幻灯片形式播放指定图像文件。
- 浏览器：以浏览器方式显示网址内容或打开本地 html 文件。

- **音频来源**

- 音频输入采集：话筒输入。
- 音频输出采集：音响输出。
- 应用程序音频捕获：获取指定应用程序的音频输出。

- **视频来源**

- 视频采集设备：捕获设备的视频画面，多用于摄像头。
- 窗口采集：捕获指定程序的窗口画面。使用窗口采集时，窗口不能最小化，否则窗口会卡住不同步。
- 游戏源：采集游戏画面。
- 显示器采集：捕获单个显示器的完整视频画面。
- 媒体源：播放本地或在线的音视频文件。

- **组合输出**

- 场景：将指定场景作为来源输入当前场景中。
- 来源镜像：将指定来源复制到当前场景。
- 分组：将来源置于同一分组，可批量更改可见性和锁定状态。

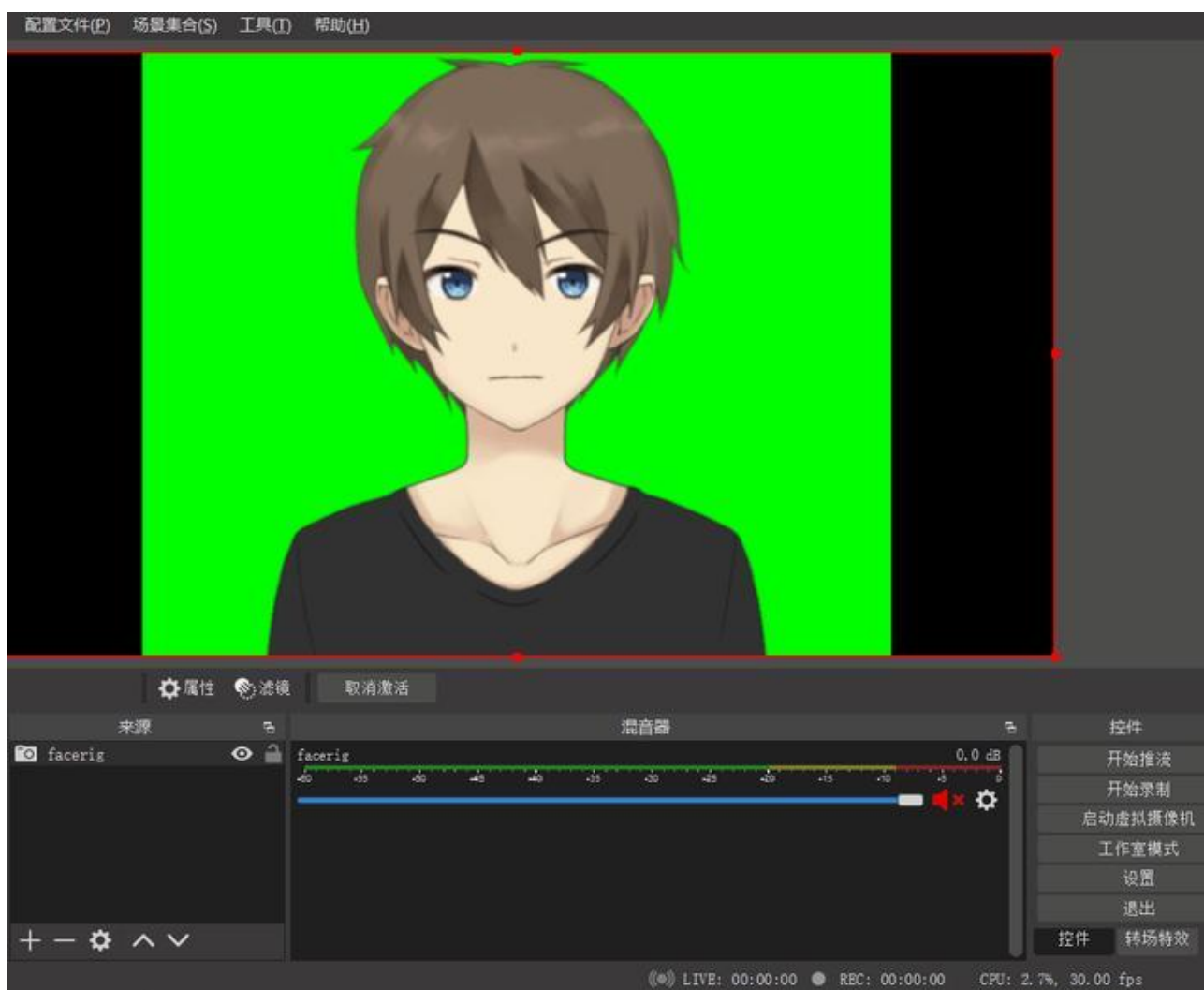
## 5.6.4 滤镜

选择「来源」>「滤镜」，可对来源素材进行裁剪、过滤、增益等多种处理。滤镜可分为效果滤镜和音视频滤镜。下文将介绍几种常用滤镜。

- **常用滤镜**

- 裁剪/填充：效果滤镜，对来源进行裁剪，剪去多余的部分。
- 色彩校正：效果滤镜，修改伽玛度、对比度、亮度、饱和度，对来源的显示效果进行调整。
- 增益：音视频滤镜，当麦克风声音过大或过小时，可以通过增益滤镜调整音量，但是不宜设置数值过大，容易造成声音失真。
- 噪音阈值：音视频滤镜，当背景有噪音时，设置噪音阈值来控制底噪，降低噪音干扰。

- **色度键（抠图）**



「色度键」属于效果滤镜，能将指定颜色变为透明，多用于绿幕实时抠图，可以在直播时把绿幕替换为其他场景。

- 模糊



「模糊」属于效果滤镜，可将指定区域模糊化以维护隐私。推荐设置：类型设为「Dual Filtering（双重过滤）」，子类型为区域，大小为 4。此设置可降低模糊滤镜占用的性能大小。要使用模糊滤镜，需要安装插件 StreamFX。

## 5.7 系统优缺点

OBS-Studio 作为一款开源的直播推流与视频录制软件，具有很多优点：

1. 开源和免费：OBS Studio 是完全免费的，没有任何隐藏费用。它是开源软件，这意味着它的代码可以被公众访问，允许社区成员为其提供贡献和改进。
2. 跨平台：OBS Studio 支持 Windows、macOS 和 Linux，这为不同的用户群体提供了方便。
3. 灵活性和自定义：OBS 允许用户创建自定义的场景和来源，配合转场、滤镜和插件等功能，为直播和录制提供了高度的灵活性。
4. 高质量的输出：OBS 提供了多种编码选项，可以达到高分辨率和高质量的直播和录制。
5. 强大的音频处理：内置的音频混音器允许对多个音频源进行深入的控制，还支持音频滤镜和效果。
6. 插件支持：由于其开放性，许多开发者为 OBS 开发了插件，为其增加了额外的功能。
7. 社区支持：作为一个受欢迎的开源项目，OBS 有一个活跃的社区，提供技术支持、教程和建议。

当然，由于实现原理与开发过程的限制，也存在一些不足：

1. 初始设置可能复杂：尽管 OBS 功能强大，但对于初学者来说，正确地配置所有的设置可能会感到有些复杂。
2. 性能需求：高质量的直播和录制可能需要强大的硬件资源，特别是当使用一些高级编码设置时。
3. 某些平台或硬件的兼容性问题：尽管 OBS 支持多个平台和大部分硬件，但仍然可能会遇到某些特定系统或硬件配置上的问题。
4. 插件的稳定性和质量：虽然存在大量的插件，但由于它们是由第三方开发者创建的，可能存在稳定性和兼容性问题。
5. 没有内置的视频编辑工具：尽管 OBS 是一个出色的录制工具，但它不提供内置的视频编辑功能，用户需要使用其他软件来编辑和后期制作。
6. 与某些应用程序的互操作性：在某些特定应用程序或场景中，捕捉窗口或特定内容可能会遇到问题。

## 5.8 创新点

### 5.8.1 现有系统创意点

1. 场景和来源结构：该功能允许用户设计复杂的布局，通过多个场景轻松切换，每个场景中都可以有多种来源，如游戏、摄像头、窗口捕捉、图像等。

2. 实时预览：在开始录制或直播之前，用户可以预览他们的设置和场景布局。

3. 自定义转场：在不同的场景之间切换时，用户不仅可以选择基本的转场效果，还可以导入自定义的转场动画。

4. 内置音频混音器：这允许用户在软件内部控制和调整音频来源的级别，并应用过滤器，如压缩、噪音抑制和增益。

5. 浏览器集成：这是一个非常强大的功能，可以用于显示网页内容、聊天窗口或基于浏览器的插件，如捐款提醒。

6. 模块化插件系统：OBS 支持各种插件，这允许开发者扩展其功能或增加新的集成。

7. 强大的编码选项：用户可以选择使用不同的编码器，并可以微调大量的设置，以优化性能或视频质量。

8. 热键支持：用户可以为几乎每一个操作设置自定义快捷键，从而提高工作流的速度和效率。

9. Studio 模式：允许用户预览下一个场景或来源布局，然后再将其推送到直播或录制中，为更复杂的生产环境提供了一个强大的工具。

10. 过滤器系统：为视频和音频来源添加各种过滤器，如色彩校正、裁剪、滚动文本等。

11. 跨平台：OBS Studio 支持 Windows、macOS 和 Linux，这确保了多种平台的用户都可以使用。

### 5.8.2新功能创意点

1. 实时 AI 助手：AI 可以通过实时分析直播者的位置、表情和动作，自动调整相机的焦点。例如，如果直播者向左移动，AI 可以自动调整焦点以保持他们在框中的位置；可以根据直播内容的情境变化（如游戏直播中的战斗或对话环节）自动选择合适的场景。

2. 集成的视频编辑工具：在录制完毕后，用户可以立即进入简易的编辑模式，进行裁剪、添加字幕或音效等操作；提供预制的文本、转场和动画模板，帮助用户快速制作高质量的视频。

3. 集成的教程和建议系统：为新用户提供简单明了的启动教程，帮助他们设置和启动第一次直播；基于用户的硬件和网络配置，为他们提供最佳的编码和推流设置建议。

## 6 人员分工

### 6.1 项目目标任务书人员分工

项目分析与基本情况介绍：肖宇航

项目背景及其特征与影响分析：丛兴柏

人员分工与验收建议、文档排版：卢思源

### 6.2 项目具体实现

实时 AI 助手：卢思源

集成的视频编辑工具：肖宇航

集成的教程和建议系统：丛兴柏

## 7 验收建议

### 7.1 验收环境

Windows 10 及更高版本

SonarQube

### 7.2 验收方法

对软件进行一系列测试，包括界面测试、功能测试、性能测试、代码质量测试、安全测试和用户体验测试，若均通过，则证明验收通过。

界面测试：对 OBS-Studio 进行 UI 界面测试，确保 OBS-Studio 的界面展示效果、按钮功能正常。



功能测试：对 OBS-Studio 进行功能测试，确保其能够完成所有预期的功能。

性能测试：测试 OBS-Studio 的性能能力，包括响应速度，系统负载等。

代码质量测试：使用 SonarQube 测试代码的编写质量，修改代码直到质量合格为止。

安全测试：测试 OBS-Studio 的安全性能，包括数据保护、信息安全方面，防止数据泄露。

用户体验测试：测试 OBS-Studio 体验功能，确保用户能够顺利地完成任务，界面是否友好，以及用户能否方便地操作、学习使用软件。

## **7.3 验收流程**

### **7.3.1 确认设计目标是否被完成**

首先确认团队关于目标系统 OBS-Studio 的要求和功能要求是否被达成。

### **7.3.2 系统测试验收**

在完成目标系统开发后，团队成员进行系统性的测试：包括界面测试，功能测试，性能测试，代码质量测试、安全测试，测试 OBS-Studio 在各种情况下是否正常运行，并且进行用户体验测试，若可以满足设计目标，若满足以上测试，则验收合格。

## 7.4 验收内容

### 7.4.1界面验收建议

对 OBS-Studio 的 UI 界面进行测试，观察 UI 界面、按钮功能是否正常

界面验收			
类别	内容	方法	标准
页面	检查每个页面是否 正常	检测每个页面能否 正常唤起，界面功能， 数据展示是否正常	页面可正常唤起， 数据展示正确
按钮	检测每个按钮的功能是否 正常	点击每个按钮，观察 对应的功能是否执行	按钮功能正常

### 7.4.2功能验收建议

检查 OBS-Studio 的功能是否满足目标系统的功能设计，并且功能不存在 Bug。

功能验收			
类别	内容	方法	标准
数据持久化功能	数据库操作	检查数据库文件是否正确 记录数据，是否可以正 确删除数据	数据库正确操作 AppModels 、CategoryModels、 WebSiteModels 数据
更新功能	OBS-Studio 可否 正常更新	检查更新	APP 可以正常更新
数据同步	OBS-Studio 多个 主机之间进行数据的	多台主机之间同步 OBS-Studio 的数	统计时间可以正 确统计、同步并合并

### 7.4.3性能验收建议

对 OBS-Studio 多次运行，记录其性能指标，对比验收标准查看是否满足验收要求。

性能验收			
类别	内容	方法	标准
UI 性能	UI 页面可流畅打开	打开 OBS-Studio，检查界面是否存在卡顿，加载时间过长的情况	界面无卡顿，加载速度快，响应速度快，平均加载时间应小于 1s
数据库性能	数据库操作流畅，响应速度快	对数据库进行大量数据插入、删除操作，检测是否存在卡顿	大量数据的插入读取不卡顿，读取半年内的时间查询时间小于 1s
数据处理性能	处理大量数据时处理时间不应过长	读取数据库中半年内的数据，并展示在页面上	半年的数据处理时间小于 2s
网络性能	网络请求速度快，网络连接稳定	请求更新，测试连接是否建立，连接是否稳定、网速是否达标	建立连接时间小于 500ms，进行一次更新时间小于 1min

7.4.4代码质量验收建议

采用 SonarQube 平台对代码质量进行分析检测，分析目标系统存在的 Bug、Code Smell 情况。若存在 Bug、Code Smell 或重复度过高等情况，依据 SonarQube 分析报告对目标系统代码再次进行修改和完善，直至代码质量完全合格为止。

代码质量验收			
类别	内容	方法	标准
Bugs	审查编写的代码是否存在 Bug	通过 IDE 和 SonarQube 综合分析代码 Bug 情况	目标系统不存在 Bug
Vulnerabilities	审查代码是否存在 Vulnerabilities	通过 SonarQube 审查代码情况，观察是否存在漏洞	目标系统不存在 Vulnerabilities
Hotspots Reviewed	审查代码的安全质量问题	通过 SonarQube 审查代码情况，观察 HotspotsReviewed	目标系统安全质量评级达到 B 及以上
Code Smells	审查目标系统代码是否存在代码异味	通过 SonarQube 审查代码情况，观察代码异味情况	目标系统不存在代码异味
Coverage	审查核心代码的测试覆盖率	通过 SonarQube 审查代码情况，观察测试覆盖率	目标系统测试覆盖核心代码，测试覆盖率 80%
Duplications	审查代码的重复度	通过 SonarQube 审查代码情况，观察是否代码重复度过高	目标系统重复度小于 3%

7.4.5安全验收建议

检查 OBS-Studio 的数据是否安全，在联网状态下检测是否发送异常数据。

安全验收			
类别	内容	方法	标准
数据库安全	检查数据库文件是否 正常	检测数据库文件，检测是否加密	数据库正常无泄漏
网络安全	进行网络通讯时检查是否有异常 信息通讯	对 OBS-Studio 网络请求时抓包检测，查看操作是否异常	网络通讯仅用于更新 APP，不涉及数据信息传输

## 8 参考文献

- [1] GB T-8567-2006 计算机软件文档编制规范