

## ----- Community issues / PRs -----

》》》》概述:

## ----- Finishing the Pull&Requests -----

》》》》概述:

## ----- Final Cleaning before C# feature -----

》》》》概述:

1:00~10:00 SDK脚本更改  
10:00~15:27 单击实体显示轮廓 (合并代码&演示)  
15:27~19:23 闲聊  
19:24~23:02 Hazel 3D 展示和代码分析  
23:03~28:23 Vulkan 和 OpenGL 的区别与分析  
28:24~34:07 闲聊  
34:08~36:34 设置线宽以及成果分析  
36:35~37:29 & 38:28~54:54 & 1:00:17~1:02:12 自定义工作文件目录 -> Application specification (代码编写&演示&提交)  
1:02:13~1:04:20 闲聊  
1:04:21~1:15:05 1.丢弃 alpha 因子以便透过上层实体的缝隙 (空白透明处) 选择下一层实体 2.纹理序列化  
1:15:05~1:17:30 补全缺失参数  
1:17:31~1:22:45 Ctrl + s 快捷键  
1:22:46~1:26:11 合并分支&闲聊  
1:26:12~1:29:33 讨论  
1:29:35~1:36:00 思考 Offset 的维护代码  
1:36:01~1:47:00 尝试编写并分析上述代码  
1:47:01~最后 关于游戏引擎的讨论与计划

## ----- C# Scripting -----

》》》》概述:

0:00~5:10 闲聊+先导 (可跳过)  
5:10~ 15:12 安装 Mono&及其注意事项 + Dotpeek  
15:12~20:50 工作流程分析  
20:50~22:10 克隆 Mono & 切换到安全的公开版本  
22:11~24:04 讨论使用子模版管理和二进制文件管理的不同之处 (可跳过)  
24:04~27:54 构建 Mono 解决方案  
27:54~32:05 讨论Rust + 以及其他问题 (可跳过)  
32:06~33:07  
33:08~35:09 提交上次的纹理序列化代码  
35:10~35:48 创建新分支  
35:38~40:28 寻找包含库目录并索引 mono.h 文件 (找挺久) 好像是因为没有正确构建 mono 项目 (可跳过)  
40:28~43:59 等待构建 + 找到 include 目录 + 为 Hazel 创建依赖项 Mono(包括许可证、README 文件)

44:00~52:00 链接 Mono (premake脚本) + 讨论 Mono 与项目的构架

52:00~55:00 回答问题 (可跳过)

55:00~1:02:22 创建 Scripting 项目文件 (基础代码框架)

1:02:22~1:14:05 编写代码

1:14:05~ 1:25:40 讲故事 (可跳过)

1:25:41~1:46:5 编写C#连接代码 + 调整 Premake 文件

1:46:06~1:57:55 修复 premake 中未链接的库

1:57:56~ 1:59:35 修复为文件构架问题

1:59:40~2:04:01 简单测试、演示

2:04:02~2:14:10 创建对象 + 调用构造函数

2:14:10~2:17:25 调用函数

2:17:25~2:24:16 调用有参数的函数

2:24:16~2:30:19 编写 ShutDown() 函数 (随后几十秒又做更改)

2:30:19~2:44:55 修改代码以修复程序的崩溃 (没有思考出来结果, 留下了一个待办注释 :-( )

2:44:55~2:46:46 在 Release 模式下进行 “生成” 操作

2:46:46~2:52:42 提交更改 (并修改 .gitignore 文件) (可跳过)

2:52:42~ Outro

》》》》为什么既要下载 mono 的二进制文件, 又要克隆 Mono 的仓库?

》》》》如果直接在 premake 文件中编写: `Library["WinSock"] = "ws2.32.lib"`, 这个库文件应该位于哪里? 是从哪个地方索引过来的?