

开发流程

1. fork源仓库到自己名下
2. clone自己fork的仓库到本地
3. 开发功能, commit
4. 从源仓库pull一次(如果有冲突, 需要先解决)
5. push到自己仓库
6. web上发起merge request

BattleCore

1. 工程结构
 1. BattleCommon
 2. BattleInterface
 3. BattleLogic
 4. BattleView
2. 生成代码
 1. BattleLogic.properties与BattleView.properties, 设置用于生成的Context类型
 2. 执行脚本GenerateCodes.bat生成代码
3. 编译(需要Make环境)
 1. 开发过程, 自我调试时打Debug版本: Make debug deploy
 2. 自己测试完成后打Release版本: Make release deploy
4. Assemblies与调试
 1. 选中Assets/Bundles/Assemblies目录, Ctrl+Shift+M生成MDB(每次更改都需要重新生成)
 2. IDE连结Unity, 开始Debug
 - 注意BattleDef.lua中, EnableLogicThread需要先设置为false

编辑器

1. 插件Odin
2. 编辑器设计思路
 1. Editor
 2. Runtime
3. 编译与拷贝
 1. IDE 上Build Release 版本(需要先切换到Release模式, Build-> Change Solution Configuration)
 2. 手动或者通过菜单Tools/CopyDll复制到游戏工程中, 手动复制到BattleCore的Libraries文件夹中
4. 注意:
 - SkillEditor.dll以及LevelEditor.dll(两个Runtime版本)不能引用UnityEngine, 需要在校验服跑

