

C++期末项目文档

指导老师：赵钦佩

组员：刘永轩 陈开昕 洪欣鹏 夏宇宁

一、实现的基本功能

- 1、支持地图绘制、建筑物绘制、兵种绘制等
- 2、(1) 键盘：
上、下、左、右：控制游戏视角的移动
ESC：弹出菜单，包含继续游戏和退出游戏两个选项
(2) 鼠标：可以对己方目标坦克和士兵等进行控制，支持拖选
- 3、建筑物种类：基地、兵营、车厂、电厂、矿源、资源站
- 4、步兵种类：大兵，警犬
- 5、战车种类：坦克
- 6、支持局域网联机对战，且支持 1-4 人在同一地图游戏
- 7、使用混合空间合成了不同状态下的动画效果

二、实现的可选功能

- 1、支持两张地图：沙漠、城市
- 2、支持更多兵种：狙击兵、工兵
- 3、支持更多战车：导弹车
- 4、队员独立制作时实现了 (1) 房间列表、(2) 国家选择、(3) 聊天系统，在聊天系统中实现了一个玩家和服务器中所有玩家聊天的功能。唯一存在的问题是，在将聊天模块和选择国家功能合并到游戏中时出现了不兼容，由于时间原因暂时没能完全解决，这个问题将在之后进一步研究解决

三、加分项

- 1、界面精致，地图美观，游戏各菜单间跳转流畅，游戏体验好
- 2、实现各战车、建筑物、兵种的血条显示功能，可以直观读取目标物体当下的血量
- 3、兵种、建筑物、战车等被选中后在地面上会出现一个亮环，显式地表示被选中的物体
- 4、实现了自动寻路系统，工兵会自动在矿源和矿厂之间来回采集资源
- 5、实现了小地图的功能
- 6、网络延迟低，联机运行流畅

四、游戏玩法

1. 进入游戏后可以设置自己的用户名，并选择进行单人游戏或多人游戏
2. 在多人游戏中选择创建游戏后，将以当前游戏为 Server 创建会话房间，其他玩家选择搜索房间能搜寻当前局域网内的会话，搜寻到后点击加入游戏即可加入，多人游戏最多允许 4 人在同一张地图对战
3. 进入单人游戏或创建多人游戏时可进行地图选择，在沙漠和城市两张地图中任选一张
4. 进入游戏后会得到一个初始化的 Worker（类似星际争霸中的 MCV，用于建造各种建筑）
5. 选中（点击或拖动框选）Worker 右侧会出现建造的选项，左键点击选择建筑，并在开阔地上左键点击以建造相应的建筑
6. 建议：首先建造基地，基地可以生产更多的 Worker；注意左上角的资源余量和电力余量，如果不足将无法建造或生产
7. 建造资源站（Resource Station）后，可派遣 Worker 到附近的资源点（Resource Source，雷达样

建筑) 收集资源, Worker 在接受指令之后会自动在资源点和资源站之间往返收集并返回资源

8. 电厂可以提供更多电力, 兵营可以建造三种兵种(警犬、大兵、狙击兵), 车厂可以建造两种战车(坦克、导弹车)
9. 玩家被打败的条件为基地(Base) 被摧毁

五、选择 UE4 引擎的原因

1. UE4 是游戏制作领域及其优秀的 3D 游戏引擎, 为开发者提供了强大的编辑器, 完整的学习开发文档
2. 虚幻引擎的 Blueprints 蓝图脚本可以用来制作顶层逻辑, 提高编写效率, 减少编码工作
3. AI 是 RTS 游戏中的重要组成部分, UE4 引擎提供了强大的“行为树”(Behavior Tree), 基于行为树可以设计出更为智能的 AI
4. UE4 提供了完整的适用于多人游戏的网络库, 仅使用蓝图就可以完成网络功能的搭建

六、UE4 的多人游戏架构

1. GameMode 对象仅存在于服务器上
2. GameState 存在于服务器和客户端上, 因此服务器可以在 GameState 上使用复制的变量, 以确保所有客户端得到最新的游戏数据
3. 存在于服务器的 PlayerController 适用于所有连接到游戏的玩家。而客户端上只存在其下属玩家的 PlayerController。也就是说, 用 PlayerController 存储那些适合所有已连接客户端的属性复本, 并不是最好的做法。这时, 您应当使用 PlayerState
4. PlayerState 无论是存在于服务器还是客户端, 均适用于所有与游戏连接的玩家。这个类可用于所有客户端(而不只是所属客户端)所需的属性(如单个玩家的当前分数)复本
5. Pawns (包括 Characters) 也存在于服务器和所有客户端之上, 而且也可以包含复制的变量和事件。至于对特定变量或事件使用 PlayerState 还是 Pawn, 则取决于具体情况

七、主要类的设计

1. RTSGameMode 定义了游戏模式、规则等信息
2. RTSGameState 定义了游戏的状态, 存在于服务器与客户端上, 用于保存当前游戏的的状态
3. RTSPlayerController 定义了游戏的操作控制, 包括视角移动、选择、以及下达指令等
4. RTSCharacter 从 UE4 的基础类 Character 继承而来、定义了最基本的 RTS 单位、包括建筑、士兵、车辆全部由此类衍生
5. RTSCharacterAIController 根据定义好的行为树, 接受移动、攻击、采矿等指令并作出相应反应, 也包括在一定范围内自动攻击等
6. 各 Component 类, 将尽可能多的功能或属性分出于 Component 类, 作为组件通过蓝图附加在基础的类上, 赋予基础类资源、攻击、生命值、建造、生产等功能和属性

八、游戏场景及模型的设计理念

考虑到不合理的架构可能会导致 UE4 的效能下降, 为尽可能减少资源消耗、GPU 占用, 以获得更流畅的游戏体验, 我们选用了低面数的卡通风格化模型, 减少渲染消耗, 游戏风格偏向卡通