荆甲RTS框架

背景介绍:

为了更好地合作统一开发，需要制定一个共同开发协议，对开发过程中的代码进行标准化和规范，让框架更加通用。

# 玩家界面协议:

电脑端:鼠标与键盘操作，鼠标左键进行框选，右键点击地面移动，点击敌人判断攻击。左键点击建筑物(自方建筑)，弹出信息面板。

移动端:手势与按钮操作，点击建筑或者物体显示详细信息面板，拖动则移动视野，两指缩放。 点击框选按钮进入框选状态，再次拖动，则框选物体，被框选物体进入选中状态。如果此时为选中状态时，点击地面进行移动，如果点击敌人则判断攻击，并取消选中状态。

资源存放协议:

整个项目下会有不同的文件夹，请归类存放，以便整理。

①

Jinjia\_Scripts:此文件夹里面存放脚本。里面的脚本通常存放的是通用脚本，例如Jinjia\_Game\_Manager等等，如果是模型的控制器Controler 脚本，则和模型放在单独文件夹下。

②

Jinjia\_Resources:此文件夹下放置模型文件夹，每个模型文件夹命名为:

Jinjia\_xxx\_model，xxx代表模型的名称。

每个模型 文件夹内包括以下内容：模型，贴图，控制器脚本，动画控制器（Animator）。

③Jinjia\_Animations:此文件夹下放置Jinjia\_xx\_Animation,用于存放每种模型的动画。

一，介绍:

1.首先手机和电脑端可以在开头写个布尔判断，然后同一个脚本里面分别写两个平台的方法，如果当前为其中一个平台，另一个就失效。 电脑就是鼠标键盘，手机就是手势和按钮。

2.游戏分两层管理，第一层为指挥层，分别有四个管理器，Ui\_Manager,Game\_Managerw,Input\_Manager,Ai\_Manager。第一层例如GameManager统一调度指挥所有单位的宏观指令，例如框选对象，将对象存进列表等，而具体的动画，攻击效果，移动等就交给第二层级的脚本，挂在具体的单位上，并且留出接口API 。当要进行移动攻击时，由第一层级管理器调用第二层级脚本的API 或者函数来进行达到效果。

同理，UI\_Manager则是管理所有UI界面，而第二层级的脚本则具体地管理当前界面的具体操作。

二，脚本以及变量规范命名格式

1.

为了我们的框架更有辨识度，开头统一添加Jinjia。

第一层称为管理器，固定命名格式为：

Jinjia\_Game\_Manager,

Jinjia\_Ui\_Manager,

Jinjia\_Input\_Manager,

Jinjia\_Ai\_Manager

分别功能是对游戏中所有单位进行管理，对所有Ui进行管理，对用户操作输入进行管理，对电脑Ai进行管理。

2.

第二层级称为控制器(Controler)，通过被第一层级管理器调用，来执行具体功能。

控制器可根据程序员具体需要，在第一层核心管理器的框架内进行自定义。

游戏中每个模型单独放在一个文件夹，文件夹中包括单个模型，贴图，以及对应的脚本(属于第二层级，控制器Controler)

命令规范:

Jinjia\_xxx(单位的种类)\_Controler。

例如控制普通士兵的脚本，可以为:

Jinjia\_comSodier\_Controler.

例如我有很多个兵种，则comSodier 是其中一种。

三，核心脚本介绍

1. 通用脚本

①Jinjia\_UseType

此脚本外部具有序列化，可在外部选择士兵，建筑，载具的种类。作用与Tag 一样。起因是因为Tag只有一个，不够用，所以自己写了一个，几乎囊括了游戏中所有需要的判断标志。

②Jinjia\_Game\_Manager

新加功能一

补充，游戏管理器将会添加 自动注册器，新的单位不用手动拖进Jinjia\_Game\_Manager脚本文件的列表下注册。前提是:所有新加的单位都存放在 Game\_unit这个节点下，游戏管理器将会自动索引节点下所有单位进行注册。

新加功能二

Jinjia\_Game\_Manager会通过挂在单位上的脚本Jinjia\_UseType自动识别单位的标志，甄别单位上对应的脚本。进行匹配调用单位身上的第二层级控制器。

也就是说，程序员不用关心第一层级管理器的代码，只需要写自己第二层单位种类具体动画，攻击，移动的控制器就行了。

③Jinjia\_Ui\_Manager

④Jinjia\_Input\_Manager

⑤Jinjia\_Ai\_Manager

2.自定义脚本(根据不同兵种定制)

例如，Jinjia\_tank\_Controler，为控制坦克的控制器。

Jinjia\_pao\_Controler，为控制炮手士兵的控制器。

命名规范为Jinjia\_xxx\_Controler

四，API使用方法