

Cocos2d-x游戏引擎

概述

内存管理机制

更新调度机制

重要数据结构

cocos2d::Ref类

cocos2d::Layer类

cocos2d::Sprite类

cocos2d::TMXTiledMap类

cocos2d::DrawNode类

cocos2d::LabelBMFont类

战场场景

战场场景是FudanCraft游戏的核心部分，也是和用户交互的主要界面。

战场场景需求分析

作为一个交互性、娱乐性、竞技性较强的RTS游戏，FudanCraft的战场场景需要实现以下一些几项主要功能：

- 将战场地图和战场地图上地图单位的实时运动情况展现给用户；
- 允许用户通过鼠标移动或键盘按键移动视角，观察整个地图，并灵活切换视角，观察基地等关键位置；
- 允许用户通过鼠标的点击、拖框等操作完成选中单位、移动单位和下达攻击命令等游戏操作；
- 提供菜单，允许用户通过点击菜单创建单位；
- 显示金钱增减、单位被攻击、摧毁和最后的胜利提示等重要的游戏消息；
- 播放动感的背景音乐，给用户提供良好的视听体验

游戏场景的具体操作方法和交互模式在用户手册中有详细描述。

战场场景相关的数据结构

战场场景的逻辑主要在 `class BattleScene` 类中实现，与战场场景相关的类还有 `class ControlPanel` 等。

BattleScene类

BattleScene类是战场场景的主要部分，包含战场地图、鼠标操作、金钱显示、游戏消息显示等功能部件，BattleScene类实现了这些功能部件的初始化和调度。

BattleScene直接继承自 cocos2d::Layer，是一个完整的cocos2d图层，便于初始化、更新和场景切换。BattleScene类中重要的属性如下表所示：

变量名	类型	结构	种类	初值
player_id	int	普通变量	private	0
socket_server	SocketServer*	指针	private	nullptr
socket_client	SocketClient*	指针	private	nullptr
battle_map	cocos2d::TMXTiledMap*	指针	private	nullptr
grid_map	GridMap*	指针	private	nullptr
unit_manger	UnitManager*	指针	private	nullptr
control_panel_	ControlPanel*	指针	private	nullptr
mouse_rect	MouseRect*	指针	private	nullptr
money	Money*	指针	private	nullptr
notice	Notice*	指针	private	nullptr
msg_set	GameMessageSet	类对象	public	未初始化

Battle类的重要方法如下：

- static BattleScene* create(SocketClient* _socket_client, SocketServer* _socket_server); 按照cocos2d内存管理方式创建BattleScene对象，返回指向新对象的指针
- virtual bool init(SocketClient* _socket_client, SocketServer* _socket_server); 初始化战场场景和内部的各个控件
- void win(); 通知战场场景游戏胜利结束，显示消息并禁止后续操作
- void lose(); 通知战场场景游戏以失败结束，显示消息并禁止后续操作
- void update(float f) override; 战场场景的更新函数
- bool onTouchBegan(cocos2d::Touch*, cocos2d::Event*)override; 触摸开始事件回调函数
- void onTouchMoved(cocos2d::Touch*, cocos2d::Event*)override; 触摸移动事件回调函数
- void onTouchEnded(cocos2d::Touch*, cocos2d::Event*)override; 触摸结束事件回调函数
- void onKeyPressed(cocos2d::EventKeyboard::KeyCode, cocos2d::Event *) override; 按键事件回调函数
- void scrollMap(); 地图滚动函数
- void focusOnBase(); 将视角集中到当前用户的基地
- void destroyReward(int destroyed_type); 获取击毁地方单位的奖励
- void menuBackCallback(cocos2d::Ref* pSender);

MouseRect类

MouseRect类是用户通过触摸移动或按住拖动鼠标的操作，在地图上拖出出矩形这一功能的抽象。MouseRect需要实现在地图移动和视角变化时灵活变化形状的功能。

MouseRect继承自 `cocos2d::DrawNode`，`cocos2d::DrawNode` 是cocos2d引擎提供的基础绘图类，可以灵活地画出空心、实心矩形多边形等基础形状。MouseRect的主要属性有：

变量名	类型	结构	种类	初值
touch_start	cocos2d::Point	类对象	public	未初始化
touch_end	cocos2d::Point	类对象	public	未初始化
start	cocos2d::Point	类对象	public	未初始化
end	cocos2d::Point	类对象	public	未初始化

MouseRect的重要方法有：

- `void update(float f) override;` 鼠标矩形的更新函数

Money类

Money类用于存储和管理用户当前持有的金钱，并在战场场景中更新和显示金钱的数值。

Money类继承自 `cocos2d::LabelBMFont`，便于金钱数值的显示和快速更新。

Money类的重要属性如下：

变量名	类型	结构	种类	初值
money	int	普通变量	private	0
timer	int	普通变量	private	0
inc_prd	int	普通变量	private	宏MONEY_INC_PERIOD
inc_amt	int	普通变量	private	宏MONEY_INC_AMOUNT

Money类的重要方法如下：

- `void update(float f) override;` 金钱自动更新函数，使金钱每隔一段时间增加一定数值
- `bool init() override;` 金钱初始化函数，初始化金钱为某一固定数值，读取BM字体集，并进行显示
- `void updateMoneyDisplay();` 更新金钱的显示
- `bool checkMoney(int cost) const;` 检查金钱是否足够
- `void spendMoney(int cost);` 花掉数值等于cost的金钱，即金钱数量减少cost
- `void increaseMoney(int amount);` 金钱数量增加amount

Notice类

Notice类用于显示重要的游戏提示。Notice类实现了游戏提示的更新、定时消失和固定显示。

Notice类继承自 `cocos2d::LabelBMFont`，便于游戏提示的显示和快速更新。

Notice类的重要属性如下：

变量名	类型	结构	种类	初值
ntc_life	int	普通变量	private	0
timer	int	普通变量	private	0

Notice类的重要方法如下：

- `void update(float f) override;` 游戏提示的更新函数，实现游戏提示的定时消失
- `void displayNotice(std::string ntc, int _ntc_life);` 显示提示函数，显示字符串ntc对应的游戏提示，定时_ntc_life后消失
- `void displayNotice(std::string ntc);` 显示提示函数的重载，显示字符串ntc对应的游戏提示，固定显示不消失
- `bool init() override;` 游戏提示初始化，初始提示为"Welcome to FudanCraft!"，读取BM字体集，并进行显示

ControlPanel类

战场场景相关的重要算法

网络模块与战场场景的初始化

视角移动与地图滚动的实现

鼠标选框的实现

地图

地图是RTS游戏的关键部分，单位的创建、移动都与地图密切相关。