#### 为NPC添加AI

使用前，你需要导入SheepAI和SheepAIHandler的Java类

代码如下

var SheepAIHandler = Java.type("saltsheep.ssl.api.SheepAIHandler");

var SheepAI = Java.type("saltsheep.ssl.api.SheepAI");

创建AI的四种形式：

new SheepAI(entity,shouldRun,run);

new SheepAI(entity,shouldRun,run,endedRun);

new SheepAI(entity,shouldRun,run,canStopDefault);

new SheepAI(entity,shouldRun,run,endedRun,canStopDefault);

其中，

entity应传入NPCAPI实体（IEntity）

shouldRun应传入一个函数，参数为NPCAPI实体，返回值boolean，例子如下：

function myShouldRun(entity){

    return true;

}

run应传入一个函数，参数为NPCAPI实体，不返回值，例子如下：

（return表示提前结束函数）

function myRun(entity){

    return ;

}

endedRun应传入一个函数，参数为NPCAPI实体，不返回值，例子如下：

（return表示提前结束函数）

function myEndedRun(entity){

    return ;

}

canStopDefault应传入一个布尔值。

再讲讲五个参数依次的含义：

entity，指AI的所属实体

shouldRun，SSL每一刻都会检查AI是否能够运行，通过执行shouldRun，如果其返回true则会运行该AI，这一部分是和MC同一线程运行，你可以在此做一些预准备工作。

Run，AI的逻辑实现部分，这一部分将会挂在额外线程上运行，所以任何逻辑操作都必须符合run代码规范，这会在下文提到。

endedRun，在代码结束后运行，即使AI被停止也可以运行

canStopDefault，AI在开始运行时，默认的canStop时，canStop我们会在后文介绍。

在创建AI之后，你应用SheepAIHandler为实体注册AI，以下是SheepAIHandler的几个API：

SheepAIHandler.addAI(entity,group,ai)

SheepAIHandler.stopAll(entity)

SheepAIHandler.clearAI(entity)

entity应传入NPCAPI实体（IEntity）

group应传入一个字符串

ai应传入一个SheepAI实例

group是什么？为什么要group？

对于AI，SheepAI在实现上采用了分组管理，机理是：

同一group的AI只能有一个运行，如果有正在运行的AI，这个group内的其他AI在该运行AI结束前不会被触发。

不同group的AI可以并行运行。

stopAll是什么？

stopAll，它可以停止该实体所有canStop为true的AI

clearAI是什么？

clearAI可以停止该实体所有canStop为true的AI，并且将该实体已添加的AI清空，我强烈建议在实体init阶段加上一句clearAI，在clearAI后添加AI

#### AI的使用方法

我希望你使用的API，有且仅有以下几个（将以实例代码的形式展现）：

function init(e){

    SheepAIHandler.clearAI(e.npc);

    ai = new SheepAI(e.npc,aiShouldRun,aiRun,function(npc){

        npc.say("代码结束执行！endedRun!");

    },false);

    SheepAIHandler.addAI(e.npc,"示例AI组",ai);

}

function died(e){

    SheepAIHandler.clearAI(e.npc);

}

function interact(e){

    ai.stopThread();

}

var ai;

var aiShouldRun = function(npc){

    return true;

};

var aiRun = function(npc){

    ai.setCanStop(true);

    ai.work(function(){

        npc.say("这是运行部分！");

    });

    ai.waitTick(10);//\*这将会使得AI暂停10游戏刻

    ai.work(function(){

        npc.say("暂停之后……");

    });

};

希望你使用的API共四个，work,waitTick,setCanStop,stopThread

以下是API功能讲解：

work(Runnable)

运行一段代码

waitTick(int)

暂停AI指定刻数的时间

setCanStop(bool)

设置AI是否可以被停止，如果AI无法被停止（canStop为false），那么AI会继续运行下去。

额外特性：

创建传入的canStopDefault值，决定了AI开始运行时的canStop值

如果canStop为true，那么当实体死亡时，会自动停止AI

stopThread()

如果AI可以被停止，那么会直接停止AI剩下的工作，而endedRun会立即执行。

## 代码规范!

这非常重要！请务必注意！

①你不应该在run内，work外进行任何的游戏逻辑操作！但你可以通过在work外定义变量，在work内对变量赋值缓存来变相实现逻辑，但如果你这么做了，请不要在同一刻内调用这个变量，因为AI的实现是waitTick结束->运行->添加work->遇到waitTick->处理work，work的运行将会比同一刻内的run内容慢！

②run内，work外，可以使用三个API，work,waitTick,setCanStop

③work内不应该调用work,waitTick,setCanStop

④你不应该在aiA的run内调用aiA.stopThread()，也即是不可以让AI自己停止自己来实现提前结束代码，但你可以在aiB的run内，以及任何其他地方这么做！

⑤不要在ai自己的run以外的地方调用work,waitTick,setCanStop！也即只有aiA自己可以调用work,waitTick,setCanStop，像是aiB以及任何其他地方都不可以这么做！

⑥在waitTick之间，有且仅能有一个work

如果你不遵守以上规范，你的代码会有概率出现问题，这难以解决。

目前可以确定的，不遵守规范的可能后果：

①在run内work外进行游戏逻辑操作，导致代码出现了执行顺序/数据异常之类的问题

## 完整例子

var SheepAIHandler = Java.type("saltsheep.ssl.api.SheepAIHandler");

var SheepAI = Java.type("saltsheep.ssl.api.SheepAI");

var ai = null;

//\*trigger的设计目的在于，通过interact触发AI

var trigger = false;

function init(e){

    SheepAIHandler.clearAI(e.npc);

    ai = new SheepAI(e.npc,

        function(npc){

            if(trigger){

                trigger = false;

                npc.say("shouldRun:"+npc.world.totalTime);

                return true;

            }

            return false;

        },

        function(npc){

            ai.work(function(){

                npc.say("run1:"+npc.world.totalTime);

            });

            ai.waitTick(100);

            ai.work(function(){

                npc.say("run2:"+npc.world.totalTime);

            });

            ai.waitTick(20);

        },

        function(npc){

            npc.say("endedRun")

        });

    SheepAIHandler.addAI(e.npc,"group",ai);

}

function interact(e){

    //\*即使没在运行也可以stop

    ai.stopThread();

    e.npc.say(e.npc.world.totalTime);

    //\*只在空手时触发AI

    if(e.player.getMainhandItem().isEmpty())

        trigger = true;

}