## 改进 PlantEater

//改进：设置初始方向，避免食草者只在一块区域活动，最终饿死

提高初始能量值，防止死的太快

function SmartPlantEater() {

this.energy = 30;

this.direction = "e";

}

//改进：提高食草者繁殖的能量标准，降低食草者增长速度。

SmartPlantEater.prototype.act = function(view) {

var space = view.find(" ");

if (this.energy > 90 && space)//当动物能量大于90且周围有空格，繁殖

return {type: "reproduce", direction: space};

//原来做法：当发现植物时，就随机挑一个吃

// var plant = view.find("\*");

// if (plant)

// return {type: "eat", direction: plant};

//改进：当只有一个植物时，不吃，降低植物消失速度

var plants = view.findAll("\*");

if (plants.length > 1)，

return {type: "eat", direction: randomElement(plants)};

if (view.look(this.direction) != " " && space)//如果当前方向不是空格且周围有空格，移动到空格

this.direction = space;

return {type: "move", direction: this.direction};

};

## 实现食肉动物 Tiger

function Tiger() {

this.energy = 100;

//初始方向为w，与食草者反方向，增加随机性

this.direction = "w";

//跟踪最近6次turn中，每一次，tiger周围食草者的数量

this.preySeen = [];

}

Tiger.prototype.act = function(view) {

//每一次turn，食草者数量的平均数量

var seenPerTurn = this.preySeen.reduce(function(a, b) {

return a + b;

}, 0) / this.preySeen.length;

var prey = view.findAll("O");

this.preySeen.push(prey.length);

// 跟踪食草者数量的数组超过6后，删除最前面的值，保持数组的最新状态

if (this.preySeen.length > 6)

this.preySeen.shift();

// 当有食草者且平均数量大于0.25，随机选个方向，吃

if (prey.length && seenPerTurn > 0.25)

return {type: "eat", direction: randomElement(prey)};