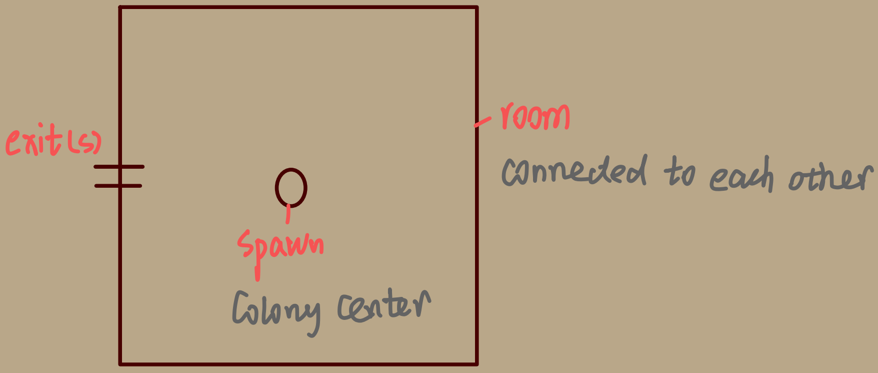
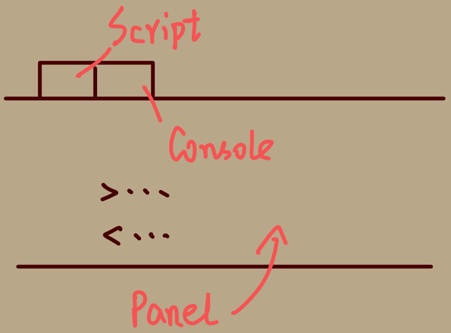
Steam Screeps

Game UI and Basic Scripting





Browser Console (Ctrl+⇧+J)

Open / close (Alt+↵)

Spawn $\xrightarrow{\text{creates}}$ Creeps { name : (certain body parts) have some skills

[WORK, CARRY, MOVE] 这样的数组.

* Address the spawn:

Game.spawns['Spawn1']

* Create a worker creep:

```
Game.spawns['Spawn1'].spawnCreep([WORK, CARRY, MOVE], 'Harvester1');
```

⇓ Needs a few seconds

The name is important

* See all the characters of your creep (or other objects)

by: Utilizing the "View" Action

直接点击 Spawn ： 看详情

* 🟨 Valuable game resource

可用 Creeps 收获（需要至少一个"WORK" body）

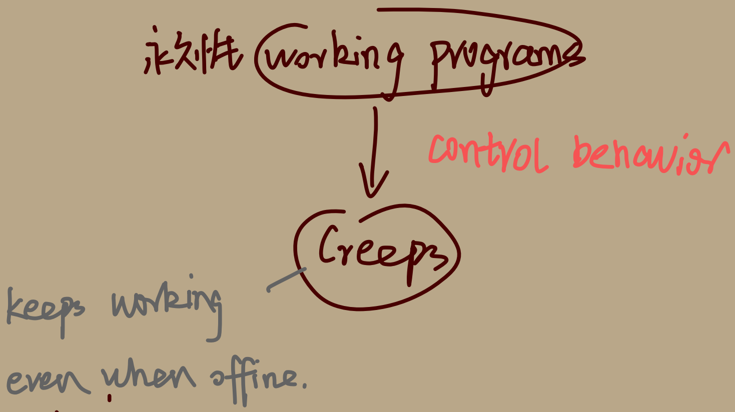
可用 Creeps 运输（需要有"CARRY" parts）

* Permanently working command:

Script tab ✓

Console tab ✗

* Game: 循环 (Loop)



基本格式:

```
module.exports.loop = function() {


}
```

* 具体如何使 Creep 去 harvest energy

Game tick: 执行一次 Commands

只可收集邻近的 (adjacent) energy source

```
module.methods.loop = function() {
    var creep = Game.creeps["Harvester1"];
    var sources = creep.room.find(FIND_SOURCES);
    if (creep.harvest(sources[0]) == ERR_NOT_IN_RANGE) {
        creep.moveTo(sources[0]);
      }
  }
```

采集 energy from the source

*

* 使 Creep Transfer energy back to spawn.

(使用 Creep. transfer)

```
module.methods.loop = function() {
    var creep = Game.creeps["Harvester1"];
    if (creep.store.GetFreeCapacity()>0 ) {
        var sources = creep.room.find(FIND_SOURCES);
        if (creep.harvest(sources[0]) == ERR_NOT_IN_RANGE) {
          creep.moveTo(sources[0]);
        }
    }
    else {
       if (Creep.transfer(Game.spawns["spawn1"], RESOURCE_
         ENERGY) == ERR_NOT_IN_RANGE) {
        creep. moveTo(Game.spawns["Spawn1"]
      }
    }
};
```

* Creeps 生命周期:

几乎均为 1500 game ticks => dies

创建新 Creep: 200 energy

```
Game.spawns["Spawn1"].spawnCreep([WORK, CARRY, MOVE], "
    Harvester2")
```

→ 返回值: { 0 成功
          ? 能量不足

```
module.methods.loop = function() {
    for(var name in Game.creeps) {    ← 循环
        var creep = Game.creeps[name];
        if (creep.store.getFreeCapacity()>0) {
            var sources = creep.room.find(FIND_SOURCES);
            if (creep.harvest(sources[0])== ERR_NOT_IN_RANGE) {
                creep.moveTo(sources[0]);
            }
        }
        else {
            if (Creep.transfer (Game.spawns["Spawn1"], RESOURCE_
               ENERGY)== ERR_NOT_IN_RANGE) {
              creep.moveTo(Game.spawns["Spawn1"]
            }
        }
    }
}
```

* 封装: behavior → seperate module

Create a module called role.harvester

```
var roleHarvester = {

    /** @param {Creep} creep **/
    run: function (creep) {
    if (creep.store.getFreeCapacity()>0) {
        var sources = creep.room.find(FIND_SOURCES);
        if (creep.harvest(sources[0])== ERR_NOT_IN_RANGE) {
            creep.moveTo(sources[0]);
        }
    }
    else {
       if (Creep.transfer(Game.spawns["spawn1"], RESOURCE_
          ENERGY)== ERR_NOT_IN_RANGE) {
         creep. moveTo(Game.spawns["Spawn1"]
       }
   }
}

module.exports = roleHarvester;
```

* 封装后运行.

```
var roleHarvester = require("role.Harvester")
module.exports.loop = function () {

    for (var name in Game.creeps) {
        var creep = Game.creeps[name]
        roleHarvester.run(creep)
    }
}
```

Upgrading Controller:

引入新对象: Room Controller

无敌, 可用于建造设施, 可升级

可建更多

New Creep: Upgrader 1

```
Game.spawns["Spawn1"].spawnCreep([WORK,CARRY,MOVE],
    "Upgrader1");
```

* 利用 Creep 的 memory property 决定其行为

可用 Memory 查看所有 memory 属性, 也可直接属性界面看.

```
Game.creeps["Harvester1"].memory.role = "harvester"
Game.creeps["Upgrader1"].memory.role = "upgrader"
```

* 设定 Upgrader 的行为

- 同样需要 harvest energy
- 不同在于. upgrader 要去 Controller 并

apply the function upgradeController

* Controller 对象: Creep.room.controller

* 新 module : role.upgrader.

```
var roleUpgrader= {

    /** @param {Creep} creep **/
    run: function(creep) {
        if(creep.store[RESOURCE_ENERGY]==0) {


        }
        else {

    }   }
}
                 if(creep.upgradeController(creep.room.Controller)
                   == ERR_NOT_IN_RANGE) {
                     creep.moveTo(creep.room.~~Co~~ controller)
                 }
```

* 应用 New Module

```
if (creep.memory.role == "harvester")
{
    roleHarvester.run(creep)
}
```

```
if(    ...      )
{
    ...
}
```

* Controller 规则:

若 20000 game ticks 内未升级, 则降一级.

若等级降为0. 失去对房间的控制权 (其它玩家可 capture it freely)

确保至少一个在执行 UpgradeController.

Building Structures:

Walls.　　Ramparts

下一章讲

Extensions

建造更大 Creeps 所必须

必需

* 更高效的 creeps:

方法: Give it several WORKs.

目前限制: A lone spawn: 仅有 300 energy

Costly Creep

* 2级 controller 可建5个 spawn extensions

* Extensions 位置可随意放

* 具体建造 Extensions 过程:

1. 孵新 Creeps {memory: {role: "builder"}}

```
Game.spawns["Spawn1"].spawnCreep([WORK, CARRY, MOVE], "
  Builder1", {memory: {role: "builder"}});
```

2. 定义 behavior:

role.builder 新 module

重要语句: `Room.find(FIND_Construction_Sites)`

`Creep.build`

3. Structure 需 energy ← Creep 可自给 harvest)

需避免来回奔波，消耗货物

⇓

逻辑策略: creating a new boolean variable

creep.memory.building

作用: tell creeps when to switch tasks

4. 可视化: Creep.say visualizePathStyle 添加到 moveTo method

5. Work with extensions

① harvesters 不仅给 spawn 供能还有 extensions

② 可用 Game.structures 或 Room.find(FIND_STRUCTURES) 获取 structure 对象

③ 使用 structure.structureType == STRUCTURE_EXTENSION 来 filter 出类型为 extension 的 structure.

(或 structure instanceof StructureExtension)

6. 此 Room 内总能量: Room.energyAvailable

将该数值 console 出来

7. All the structures are filled with energy

=> 可以建造大 Creeps 了

8. 现在 spawn. Extensions 共有 550 个 energy

可建大的 Creeps: 其 body 为

[WORK, WORK, WORK, WORK, CARRY, MOVE, MOVE]

为其命名为 harvesterBig

9. 此来 big creeps 去 restrain source

=> 殖民地有了 maximum energy boost 之概念.

10. 结果:

升 Controller → 建 Extension → 造 powerful creeps

本质是提升 Colony 的效率

并 节约 CPU 资源 (控制 creeps 靠 CPU)

=> 由此才能在 online mode 游戏

自动生成 Creeps

此游戏的难点所在

1 老Creeps 死亡. 需建造新的.

游戏不提供死亡事件. 自行数一判断

2. 统计方法:

① _.filter(Game.creeps, ...

② 用 memory 中的 role

3. 任务：孵2个 harvesters (任何时候)

方法： Game.spawns["Spawn1"].spawnCreep()

在数量<~~3~~2时运行.

测试： Game.creeps["Harvester1"].suicide()

4 内存溢出：

Creeps死去后的 memory 依旧为同名 creep 保存

故随机新名字 leads to memory overflow

解决： clear it in the beginning tick

(creat creeps之前执行)

5. 另一种方法

StructureSpawn.renewCreep

Tutorial中暂不讲

Defending Your Room

Threats: Other players + NPC

=> Colony Defense

1. Safe Mode:

```
Game.spawns["Spawn1"].room.controller.activateSafeMode()
```

2. 清理敌人:

① Towers: 可在任意位置攻击/治疗
效果与距离相关

```
Game.spawns["Spawn1"].room.createConstructionSite
        (23,22, STRUCTURE_TOWER)
```

② 将 Tower 设为能量补给目标

③ Tower 的 methods: attack. heal. repair
10 energy each.

对敌使用

距离很重要!

④ 查看属性: 直接点击/ Game.getObjectById()

③ Set up Auto Repair

可修理

Damaged Structures ← { Creeps
Towers

对于 tower: 调用 method.repair

Room.find and a filter to locate the damaged walls.

注: Walls are not belonged to any players
So requires the constant FIND_STRUCTURES
instead of FIND_MY_STRUCTURES