**修改物质复制器**

**1、修改物质复制器的最大电力设置**

查找“if (game.global.race.governor.config.replicate.pow.on == false)”，在上方添加如下内容（需要更多电力请修改100000这个数，共两处）：

let inputEvent = new Event('input');

//最大电力设置

if (game.global.race.governor.config.replicate.pow.cap != 1000000) {

let repConfigPowCap = document.querySelectorAll(".tConfig")[8].querySelectorAll(".storage")[0].querySelectorAll(".b-numberinput input")[0];

repConfigPowCap.value = 1000000; //设置“至多使用电力”

repConfigPowCap.dispatchEvent(inputEvent);

}

添加后如图



**2、修改电力缓冲值，保留更多电力防止拉闸**

在上方添加内容的后面继续添加如下内容：

let buildingList = [...BuildingManager.managedPriorityList(), ...ProjectManager.managedPriorityList()].sort((a, b) => b.weighting - a.weighting);

//电力缓冲值设置

let repConfigPowBuffer = document.querySelectorAll(".tConfig")[8].querySelectorAll(".storage")[0].querySelectorAll(".b-numberinput input")[1];

let powerBuffer = settings.replicatorPowerBuffer;

for (let unlockedBuilding of buildingList) {

if (unlockedBuilding.powered > powerBuffer) {

powerBuffer = unlockedBuilding.powered;

}

}

repConfigPowBuffer.value = Math.ceil(powerBuffer \* 3); //设置“可用电力缓冲值”，为可建造建筑最大耗电量的3倍（倍率可自行修改）

repConfigPowBuffer.dispatchEvent(inputEvent);

添加后如图



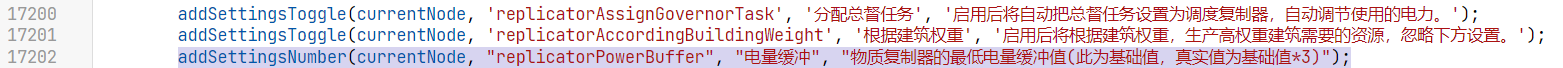
搜索“addSettingsToggle(currentNode, 'replicatorAssignGovernorTask',”，在这一行下方添加如下内容：

addSettingsNumber(currentNode, "replicatorPowerBuffer", "电量缓冲", "物质复制器的最低电量缓冲值(此为最低基础值，真实值为基础值\*3)");

PS1：倍率见上方添加内容中的蓝色字

PS2：最低基础值意为电量缓冲只会大于等于这个数的3倍，作用大概是建好飞升装置后让复制器多留些电给飞升装置，或其他类似用途

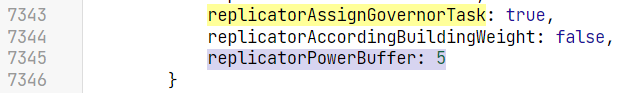
添加后如图（中间一行详见下述第3条）：



搜索“replicatorAssignGovernorTask: true”，在下方添加：

replicatorPowerBuffer: 5

添加后如图（除最后一行外，其余行最后若没有逗号请自行添加英文逗号）（中间一行详见下述第三条）：

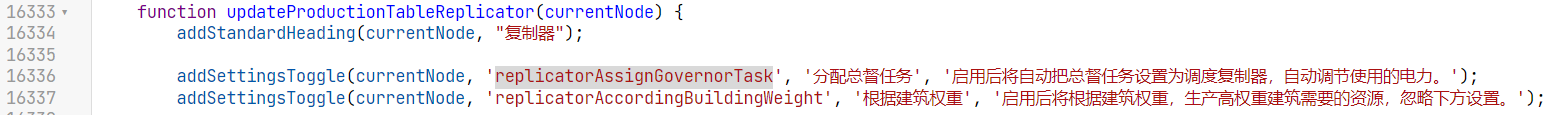


**3、根据建筑权重调整复制器生产（忽略生产设置中复制器对于资源权重和优先级的设置）（略费劲）**

搜索“addSettingsToggle(currentNode, 'replicatorAssignGovernorTask', '分配总督任务', '启用后将自动把总督任务设置为调度复制器，自动调节使用的电力。')”，这句后面没有分号的话在后面添加分号（英文分号），然后在这一行下面添加：

addSettingsToggle(currentNode, 'replicatorAccordingBuildingWeight', '根据建筑权重', '启用后将根据建筑权重，生产高权重建筑需要的资源，忽略下方设置。');

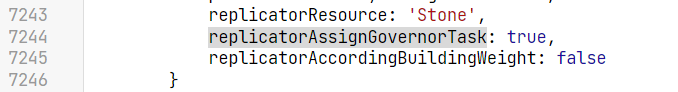
添加后如图：



搜索“replicatorAssignGovernorTask: true”，在后面添加逗号（英文逗号），在这一行下面添加（默认关闭，可将false改成true，即可默认开启）：

replicatorAccordingBuildingWeight: false

添加后如图：



将“修改电力缓冲值”中的第一行“let buildingList = [...BuildingManager.managedPriorityList(), ...ProjectManager.managedPriorityList()].sort((a, b) => b.weighting - a.weighting);”删除（没添加的话不管）

搜索“ReplicatorManager.setResource(selectedResource.id)”，将这行注释掉（前方添加//），在这一行下面添加：

var tarRes = selectedResource.id;

在刚才添加的内容下面一行的“}”的下面添加：

let buildingList = [...BuildingManager.managedPriorityList(), ...ProjectManager.managedPriorityList()].sort((a, b) => b.weighting - a.weighting);

if (settings.replicatorAccordingBuildingWeight && state.triggerTargets.length == 0 && buildingList.length > 0){

for (let i = Math.min(2, buildingList.length - 1); i >= 0; i--) {

let tarBuilding = buildingList[i];

let thisTime = 0, tarTime = 0;

for (let tarBuildRes in tarBuilding.cost) {

if (replicableResources.includes(tarBuildRes) && game.global.resource[tarBuildRes].amount < tarBuilding.cost[tarBuildRes]) {

thisTime = (tarBuilding.cost[tarBuildRes] - game.global.resource[tarBuildRes].amount) / game.global.resource[tarBuildRes].diff;

if (thisTime > tarTime && ReplicatorManager.\_industryVue.avail(tarBuildRes) == true) {

tarTime = thisTime;

tarRes = tarBuildRes;

}

}

}

}

}

try {

ReplicatorManager.setResource(tarRes);

}

catch(e) {

//console.log(tarRes);

}

添加后如图：



**4、优先生产队列所需资源（方便做挑战时手搓）**

搜索“// Disable focus queue”，将包括这一行在内的上下共三行替换为以下内容

if (game.global.queue.queue.length == 0 && game.global.race.governor.config.replicate.res.que) {

win.document.querySelector('#govOffice .options').getElementsByClassName('tConfig')[8].childNodes[2].childNodes[0].childNodes[0].click() // 没有建筑队列

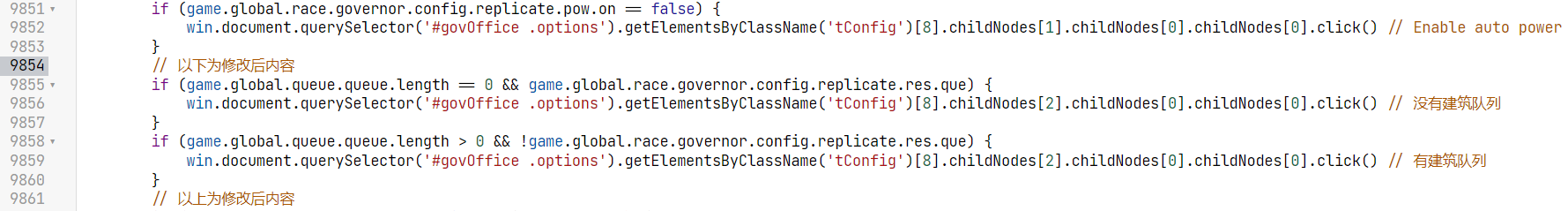
}

if (game.global.queue.queue.length > 0 && !game.global.race.governor.config.replicate.res.que) {

win.document.querySelector('#govOffice .options').getElementsByClassName('tConfig')[8].childNodes[2].childNodes[0].childNodes[0].click() // 有建筑队列

}

替换后如图：



将上述第三步（根据建筑权重调整…）的最后一部分（6行）替换为修改为以下内容：

try {

if(game.global.queue.queue.length == 0)

{

ReplicatorManager.setResource(tarRes);

}

}

catch(e) {

console.log(tarRes);

}

修改后如图：

