## 把Trigger类的updateComplete()方法改成这样

updateComplete() {

if (this.complete) {

return false;

}

if (this.actionType === "research" && techIds[this.actionId].isResearched()) {

this.complete = true;

return true;

}

if (this.actionType === "build"){

let completedCount = buildingIds[this.actionId].isMission() ? Number(buildingIds[this.actionId].isComplete()) : buildingIds[this.actionId].count;

if(completedCount >= this.actionCount) {

this.complete = true;

return true;

}else{

this.leftCount = this.actionCount - completedCount;

return false;

}

}

if (this.actionType === "arpa") {

let completedCount = arpaIds[this.actionId].count;

if(completedCount >= this.actionCount) {

this.complete = true;

return true;

}else{

this.leftCount = this.actionCount - completedCount;

return false;

}

}

return false;

}

## 把TriggerManager的resetTargetTriggers()方法改成这样（可选，用于无视资源冲突）

resetTargetTriggers() {

this.targetTriggers = [];

for (let trigger of this.priorityList) {

trigger.updateComplete();

if (!trigger.complete && trigger.areRequirementsMet() && trigger.isActionPossible() && !this.targetTriggers.some(t=>t.actionId==trigger.actionId)) {

this.targetTriggers.push(trigger);

}

}

},

## 修改 function autoTrigger()

function autoTrigger() {

let triggerActive = false;

for (let trigger of state.triggerTargets) {

let leftCount = state.triggerTargetsLeftCount[trigger.\_vueBinding];

if(leftCount && leftCount > 0){

for(let ii=0;ii<leftCount;ii++){

if (trigger.click()) {

triggerActive = true;

state.triggerTargetsLeftCount[trigger.\_id]--;

}else{break;}

}

}else if (trigger.click()) {

triggerActive = true;

}

}

return triggerActive;

}

## 修改function updatePriorityTargets()

开头加一句state.triggerTargetsLeftCount = {};

state.triggerTargets.push(obj);下面加一句if(trigger.leftCount)state.triggerTargetsLeftCount[obj.\_vueBinding]=trigger.leftCount;

对于汉化后的主脚本3.3.1.112，该方法修改后应该是这样的：

function updatePriorityTargets() {

state.triggerTargetsLeftCount = {};

state.conflictTargets = [];

state.queuedTargets = [];

state.queuedTargetsAll = [];

state.triggerTargets = [];

state.unlockedTechs = [];

state.unlockedBuildings = [];

// Building and research queues

let queueSave = settings.prioritizeQueue.includes("save");

[{type: "queue", noorder: "qAny", map: (id) => buildingIds[id] || arpaIds[id]},

{type: "r\_queue", noorder: "qAny\_res", map: (id) => techIds[id]}].forEach(queue => {

if (game.global[queue.type].display) {

for (let item of game.global[queue.type].queue) {

let obj = queue.map(item.id);

if (obj) {

state.queuedTargetsAll.push(obj);

if (obj.isAffordable(true)) {

state.queuedTargets.push(obj);

if (queueSave) {

state.conflictTargets.push({name: obj.title, cause: "队列", cost: obj.cost});

}

}

}

if (!game.global.settings[queue.noorder]) {

break;

}

}

}

});

if (SpyManager.purchaseMoney && settings.prioritizeUnify.includes("save")) {

state.conflictTargets.push({name: techIds["tech-unification"].title, cause: "收购", cost: {Money: SpyManager.purchaseMoney}});

}

if (settings.autoFleet && FleetManagerOuter.nextShipAffordable && settings.prioritizeOuterFleet.includes("save")) {

state.conflictTargets.push({name: FleetManagerOuter.nextShipName, cause: "舰船", cost: FleetManagerOuter.nextShipCost});

}

if (settings.autoTrigger) {

TriggerManager.resetTargetTriggers();

let triggerSave = settings.prioritizeTriggers.includes("save");

// Active triggers

state.triggerTargetsLeftCount = {};

for (let trigger of TriggerManager.targetTriggers) {

let id = trigger.actionId;

let obj = arpaIds[id] || buildingIds[id] || techIds[id];

if (obj) {

state.triggerTargets.push(obj);

if(trigger.leftCount)state.triggerTargetsLeftCount[obj.\_vueBinding]=trigger.leftCount;

if (triggerSave) {

state.conflictTargets.push({name: obj.title, cause: "触发器", cost: obj.cost});

}

}

}

}