**MAIN**

var roleHarvester = require('role.harvester');

var roleUpgrader = require('role.upgrader');

var roleBuilder = require('role.builder');

module.exports.loop = function () {

for(var name in Memory.creeps) {

if(!Game.creeps[name]) {

delete Memory.creeps[name];

console.log('Clearing non-existing creep memory:', name);

}

}

var harvesters = \_.filter(Game.creeps, (creep) => creep.memory.role == 'harvester');

console.log('Harvesters: ' + harvesters.length);

if(harvesters.length < 2) {

var newName = Game.spawns['Spawn1'].createCreep([WORK,CARRY,MOVE], undefined, {role: 'harvester'});

console.log('Spawning new harvester: ' + newName);

}

var harvesters = \_.filter(Game.creeps, (creep) => creep.memory.role == 'upgrader');

console.log('Upgrader: ' + harvesters.length);

if(harvesters.length < 2) {

var newName = Game.spawns['Spawn1'].createCreep([WORK,CARRY,MOVE], undefined, {role: 'upgrader'});

console.log('Spawning new upgrader: ' + newName);

}

var harvesters = \_.filter(Game.creeps, (creep) => creep.memory.role == 'builder');

console.log('Builder: ' + harvesters.length);

if(harvesters.length < 2) {

var newName = Game.spawns['Spawn1'].createCreep([WORK,CARRY,MOVE], undefined, {role: 'builder'});

console.log('Spawning new builder: ' + newName);

}

for(var name in Game.rooms) {

console.log('Room "'+name+'" has '+Game.rooms[name].energyAvailable+' energy');

}

for(var name in Game.creeps) {

var creep = Game.creeps[name];

if(creep.memory.role == 'harvester') {

roleHarvester.run(creep);

}

if(creep.memory.role == 'upgrader') {

roleUpgrader.run(creep);

}

if(creep.memory.role == 'builder') {

roleBuilder.run(creep);

}

}

}

**ROLE.BUILDER**

var roleBuilder = {

/\*\* @param {Creep} creep \*\*/

run: function(creep) {

if(creep.memory.building && creep.carry.energy == 0) {

creep.memory.building = false;

creep.say('harvesting');

}

if(!creep.memory.building && creep.carry.energy == creep.carryCapacity) {

creep.memory.building = true;

creep.say('building');

}

if(creep.memory.building) {

var targets = creep.room.find(FIND\_CONSTRUCTION\_SITES);

if(targets.length) {

if(creep.build(targets[0]) == ERR\_NOT\_IN\_RANGE) {

creep.moveTo(targets[0]);

}

}

}

else {

var sources = creep.room.find(FIND\_SOURCES);

if(creep.harvest(sources[0]) == ERR\_NOT\_IN\_RANGE) {

creep.moveTo(sources[0]);

}

}

}

};

module.exports = roleBuilder;

**ROLE.HARVESTER**

var roleHarvester = {

/\*\* @param {Creep} creep \*\*/

run: function(creep) {

if(creep.carry.energy < creep.carryCapacity) {

var sources = creep.room.find(FIND\_SOURCES);

if(creep.harvest(sources[0]) == ERR\_NOT\_IN\_RANGE) {

creep.moveTo(sources[0]);

}

}

else {

var targets = creep.room.find(FIND\_STRUCTURES, {

filter: (structure) => {

return (structure.structureType == STRUCTURE\_EXTENSION || structure.structureType == STRUCTURE\_SPAWN) &&

structure.energy < structure.energyCapacity;

}

});

if(targets.length > 0) {

if(creep.transfer(targets[0], RESOURCE\_ENERGY) == ERR\_NOT\_IN\_RANGE) {

creep.moveTo(targets[0]);

}

}

}

}

};

module.exports = roleHarvester;

**ROLE.UPGRADER**

var roleUpgrader = {

/\*\* @param {Creep} creep \*\*/

run: function(creep) {

if(creep.carry.energy == 0) {

var sources = creep.room.find(FIND\_SOURCES);

if(creep.harvest(sources[0]) == ERR\_NOT\_IN\_RANGE) {

creep.moveTo(sources[0]);

}

}

else {

if(creep.upgradeController(creep.room.controller) == ERR\_NOT\_IN\_RANGE) {

creep.moveTo(creep.room.controller);

}

}

}

};

module.exports = roleUpgrader;