```
info: SashaBot01, Sasha
// changed how far to move in one firection before changing direction
////////////////////////////////////////////////////////////////////////

main:                       // select a random direction and distance to move
        rand    [dir], 4    // computes a random number between 0 and max-1 inclusive and
                            // stores this random number into dest
        rand    [count], 15
        add     [count], 1

loop:                   // check if I am on top of food and eat if so
        sense   r2      // If the organism is on a square that contains sludge or a collection
                        // point, then dest is set to the ID type number of the sludge, or a
                        // value of 65535 if the organism is on a collection point. Dest will
                        // be set to 0 otherwise. Flags: If the organism is on a square that
                        // contains food or a collection point, the SUCCESS flag is set,
                        // otherwise the SUCCESS flag is cleared
        jns     noFood  // jumps to the specified address if the SUCCESS flag is NOT set
        eat             // If the organism is on a square that contains sludge/food then it
                        // eats the food and the food disappears from the current square. It
                        // then receives 2000 energy units. If eating the sludge will push it
                        // over 65535 energy units, then it will fail to eat it. Flags: If
                        // the organism successfully eats food, the SUCCESS flag is set,
                        // otherwise the SUCCESS flag is cleared

noFood:                     // see if we're over a collection point and release some energy
        energy  r0          // places the organism's current energy value into the dest
                            // register or memory location Flags: No effect on flags
////////////////////////////////////////////////////////////////////////
        cmp     r0, 2000
        jl      notEnufEnergy // Jumps to the specified address if the LESS flag is set.
                            // Otherwise continue execution at the nxt instruction
        sense   r5
        cmp     r5, 0xFFFF  // are we on a colleciton point?
        jne     notEnufEnergy // jumps to the address if the EQUAL flag is NOT set
        release 100         // drain my energy by 100, but get 100 points, assuming
                            // that we're releasing on a collection point if the release
                            // is successful, the SUCCESS flag is set, otherwise cleared

notEnufEnergy:              // move me
        cmp     [count], 0  // moved enough in this direction; try a new one
        je      newDir      // jumps to the address if the EQUAL flag is set
        travel  [dir]       // Moves the organism one slot in the specified direction
                            // assuming the space is not occupied by another organism or
                            // outside the sludge tank. This instruction costs 10 energy
                            // points if successful; otherwise it costs 1 energy point.
                            // When an organism moves: North: their y=y-1, South:
                            // their y=y+1 West: their x=x-1, East: their x=x+1. If it
                            // moves success, the SUCCESS flag is set, otherwise cleared
                            // travel 0 is N, travel 1 is N, travel 2 is E, travel 3 is W
        jns     newDir      // bumped into another org or the wall
                            // jumps to the address if the SUCCESS flag is NOT set
        sub     [count], 1  // sub dest, src // dest = dest - src
        jmp     loop

newDir:
        rand    [dir], 4    // select a new direction
        rand    [count], 15 // select a new count between 0 and 14
        jmp     loop

dir:
        data { 0 }          // our initial direction

count:                      // our initial count of how far to move in the cur dir
        data { 0 }
```

```
info: SashaBot02, Sasha
// changed how far to move in one firection before changing direction
// also, changed such that the nanorg always releases half of its energy

main:                       // select a random direction and distance to move
        rand    [dir], 4    // computes a random number between 0 and max-1 inclusive and
                            // stores this random number into dest
        rand    [count], 15
        add     [count], 1

loop:                   // check if I am on top of food and eat if so
        sense   r2      // If the organism is on a square that contains sludge or a collection
                        // point, then dest is set to the ID type number of the sludge, or a
                        // value of 65535 if the organism is on a collection point. Dest will
                        // be set to 0 otherwise. Flags: If the organism is on a square that
                        // contains food or a collection point, the SUCCESS flag is set,
                        // otherwise the SUCCESS flag is cleared
        jns     noFood  // jumps to the specified address if the SUCCESS flag is NOT set
        eat             // If the organism is on a square that contains sludge/food then it
                        // eats the food and the food disappears from the current square. It
                        // then receives 2000 energy units. If eating the sludge will push it
                        // over 65535 energy units, then it will fail to eat it. Flags: If
                        // the organism successfully eats food, the SUCCESS flag is set,
                        // otherwise the SUCCESS flag is cleared

noFood:                     // see if we're over a collection point and release some energy
        energy  r8          // places the organism's current energy value into the dest
                            // register or memory location Flags: No effect on flags
        div     r8, 2       // reduce value by half
        cmp     r8, 2000
        jl      notEnufEnergy // Jumps to the specified address if the LESS flag is set.
                            // Otherwise continue execution at the nxt instruction
        sense   r5
        cmp     r5, 0xFFFF  // are we on a colleciton point?
        jne     notEnufEnergy // jumps to the address if the EQUAL flag is NOT set
        release r8          // drain half of my energy, but get points, assuming
                            // that we're releasing on a collection point if the release
                            // is successful, the SUCCESS flag is set, otherwise cleared

notEnufEnergy:              // move me
        cmp     [count], 0  // moved enough in this direction; try a new one
        je      newDir      // jumps to the address if the EQUAL flag is set
        travel  [dir]       // Moves the organism one slot in the specified direction
                            // assuming the space is not occupied by another organism or
                            // outside the sludge tank. This instruction costs 10 energy
                            // points if successful; otherwise it costs 1 energy point.
                            // When an organism moves: North: their y=y-1, South:
                            // their y=y+1 West: their x=x-1, East: their x=x+1. If it
                            // moves success, the SUCCESS flag is set, otherwise cleared
                            // travel 0 is N, travel 1 is N, travel 2 is E, travel 3 is W
        jns     newDir      // bumped into another org or the wall
                            // jumps to the address if the SUCCESS flag is NOT set
        sub     [count], 1  // sub dest, src // dest = dest - src
        jmp     loop

newDir:
        rand    [dir], 4    // select a new direction
        rand    [count], 15 // select a new count between 0 and 14
        jmp     loop

dir:
        data { 0 }          // our initial direction

count:                      // our initial count of how far to move in the cur dir
        data { 0 }
```