

# **Norfair**

## **System Calls**

## **:: sysClearScreen**

The sysClearScreen system call sets the color value of all pixels in the Norfair console window to black.

**Arguments:** None

**Returns:** Nothing

**Syntax:**

```
call sysClearScreen
```

## :: sysPelGet

The sysPelGet system call reads the color value of a given pixel in the Norfair console window.

**Arguments:** The x and y coordinate values of the location of the pixel to be read. These values are passed to the sysPelGet system call on the stack.

**Returns:** This function places the 8-bit color of the specified pixel onto the top of the stack.

### Syntax:

```
push xvalue           ; pass in x of pixel
push yvalue           ; pass in y of pixel
call sysPelGet
pop colorofpel        ; pop off color of pixel for later use
```

## :: sysPelSet

The sysPelSet system call sets the color of a given pixel in the Norfair console window to the value specified.

**Arguments:** The x and y coordinate values of the location of the pixel to be set, and the new color value of the pixel to be written. These values are passed to the sysPelSet system call on the stack.

**Returns:** Nothing

### Syntax:

```
push colorofpel      ; pass in new color of pixel
push xvalue          ; pass in x of pixel
push yvalue          ; pass in y of pixel
call sysPelSet
```

## **:: sysPseudoRand (aka sysPRNG)**

The sysPseudoRand system call generates a pseudo-random value in the range [0..255].

**Arguments:** None.

**Returns:** This function places the generated pseudo-random value onto the top of the stack.

**Syntax:**

```
call sysPseudoRand
pop prandval      ; pop off pseudo-random value for later use
```

```
;
; This program writes a series of pixels of pseudo-random color values to
; pseudo-random locations on the Norfair console window.
;
```

```
entry:
```

```
    ldi r31,1
    eor r30,r30
```

```
MakeThatCall:
```

```
    call RandPels
    add r30,r31
    cpi r30,3
    brbc 1, MakeThatCall
```

```
sink:
```

```
    rjmp sink
```

```
RandPels:
```

```
    push r16 ; x
    push r17 ; y
    push r18 ; color
    push r19 ; count
    push r1
```

```
    com r1 ; negone
```

```
    ldi r19,255
```

```
NextPel:
```

```
    call sysPRNG
    pop r18
    andi r18,0x0f
    call sysPRNG
    pop r16
    call sysPRNG
    pop r17
    push r18
    push r16
    push r17
    call sysPelSet ; set pel
```

```
    add r19,r1
    brbc 1,NextPel
```

```
    pop r1
    pop r19
    pop r18
    pop r17
    pop r16
    ret
```

