Documentation: Turtle Drawing

Setters instructions

Those instructions allow the user to define the curser values

`setdefault`

`setdefault` reset all cursor values to default, it means:

- the pen set on "on"
- the position (0, 0)
- the orientation 0°
- the color: white
- the thickness set on 1

Example:

```
setcolor 255 10 10
forward 200

setdefault
forward 100
```

↑ Back to top

`setpen <on/off>`

`setpen on` makes the cursor draw will moving

`setpen off` denies the cursor from drawing will moving

Example:

```
setpen off
forward 100
setpen on
forward 100
```

↑ Back to top

`setx <x>`

`setx` will redifine the x coordonate value of the cursor

Example:

```
setx 200
forward 100
```

↑ Back to top

`sety <y>`

`sety` will redifine the y coordonate value of the cursor

Example:

```
sety 200
forward 100
```

↑ Back to top

`setposition <x> <y>`

`setposition` is a mix betwenn `setx` and `sety`, it will redifine the x and y coordonate values of the cursor

Example:

```
setposition 200 200
forward 100
```

↑ Back to top

`setorientation <angle>`

`setorientation` will redifine the current cursor orientation

Example:

```
setorientation 45
forward 100
```

↑ Back to top

`setcolor <r> <g> <b>`

`setcolor` will redifine the cursor color and tus the drawing shapes color

Example:

```
setcolor 255 14 14
forward 100
```

↑ Back to top

`setcolorhex <hex>`

`setcolorhex` will do the same thing as `setcolor` but with the hexa value of the color instead of the rgb values.

Be careful, the letters of the hexa value have to be in **capital letter**

Example:

```
setcolorhex FF0000
forward 100
```

↑ Back to top

`setthickness <thickness>`

`setthickness` will redifine the thickness of every lines drawed by the cursor

Example:

```
setthickness 5
forward 100
```

↑ Back to top

Draw/move instructions

Those instructions allow the user to make the cursor move and thus draw is the pen status is "on"

`rotate <angle>`

`rotate` instruction allow the cursor to inscrease his rotation to the defined angle

Example:

```
forward 100
rotate 95
forward 100
```

↑ Back to top

`forward <distance>`

`forward` allow the cursor to go in his direction for the specified distance

Example:

```
forward 100
```

↑ Back to top

`backward <distance>`

`backward` is the same as `forward`, but the cursor will go in the opposite direction than his orientation

Example:

```
forward 100
rotate 90
forward 100
rotate 90
backward 100
```

↑ Back to top

`left <angle>`

`left` is the same as `rotate`, but will take the opposite value of the absolute value of the angle

Example:

```
setposition 200 200
forward 100
```

```
left 100
forward 100
```

↑ Back to top

`right <angle>`

`right` is like `left`, but il will only take the absolute value of the angle

Example:

```
forward 100
right 100
forward 100
```

↑ Back to top

`arc <angle> <radius>`

`arc` allow the cursor to draw an arc of a circle of the specified `radius`

Example:

```
forward 100
arc 180 100
```

↑ Back to top

Blocks

The following methods will allow the user to create and run blocks of instructions or juste repeat instructions

`repeat <times>`

`repeat` will allow the user to repeat `times` times some instructions. All the tabulated instructions will be take. Line break also breaks the repeat block

Example:

```
repeat 4
    forward 100
    rotate 90
forward 200
```

Is the same as :

```
repeat 4
    forward 100
    rotate 90

    forward 200
```

↑ Back to top

`block <name>`

`block` allow the user to create block of instructions that can be called by the `call` instruction

Example:

```
block square
    repeat 4
        forward 100
        rotate 90
```

↑ Back to top

`call <name>`

The `call` method allow the user to execute a defined `block`

Example:

```
block square
    repeat 4
        forward 100
        rotate 90

call square
forward 100
call square
```

↑ Back to top