# *Screen Commands*

Screen instructions and functions

**Screen Open INDEX, WIDTH, HEIGHT, NUMBEROFCOLOURS, PIXELMODE**
Open a new screen
*Parameters:*
INDEX: The index of the screen to open. Any existing screen will be replaced by the new one
WIDTH: The width of the screen in pixels
HEIGHT: The height of the screen in pixels
NUMBEROFCOLOURS: the number of colors of the palette (optional)
PIXELMODE: "Lowres", "Hires", "Laced" or any combination

**Screen close INDEX**
Destroys the current screen or a given screen
*Parameters:*
INDEX: The index of the screen to destroy, if ommited will destroy the current screen

**Screen Clone INDEX**
Create an exact and synchronized copy of the current screen that can be displayed at another position and Z-order. Both screen share the same internal pixel buffers. Graphical operations are fobidden in the cloned screen
*Parameters:*
INDEX: The index of the screen to create, will replace an existing screen

**Screen Hide INDEX**
Make a screen disappear from display. The screen will remain active and drawing operation are still possible after this instruction
*Parameters:*
INDEX: The index of the screen (optional)

**Screen Show INDEX**
Make a hidden screen reappear on display
*Parameters:*
INDEX: The index of the screen (optional)

**Screen Swap INDEX**
Swaps the physical and logical buffers of a screen. (Deprecated in AOZ, legacy instruction, has no real effect)
*Parameters:*
INDEX: The index of the screen (optional)

**Screen Display INDEX, X, Y, WIDTH, HEIGHT**
Defines the display position and width of a screen
*Parameters:*
INDEX: The index of the screen to display
X: The horizontal coordinate of the top-left pixel of the screen on display (hardware coordinate)
Y: The vertical coordinate of the top-left pixel of the screen on display (hardware coordinate)
WIDTH: The number of horizontal pixels to display
HEIGHT: The number of vertical pixels to display

**Screen Center INDEX, CENTERX, CENTERY**
TOTEST! Enforces the centering of a screen
*Parameters:*
INDEX: The index of the screen to display
CENTERX: True to center the screen horizontally, False to leave the horizontal position unchanged
CENTERY: True to center the screen vertically, False to leave the vertical position unchanged

**Screen Offset INDEX, OFFSETX, OFFSETY**
Set the offet in the internal screen buffer of the top-left diplayed pixel, allowing scrollings
*Parameters:*
INDEX: The index of the screen
OFFSETX: The horizontal offset (optional)
OFFSETY: The vertical offset (optional)

**Screen To Front INDEX**
Change the display order of the screen, and passes the screen in front of all other screens
*Parameters:*
INDEX: The index of the screen

**Screen To Back INDEX**
Change the display order of the screen, and passes the screen behind all other screens
*Parameters:*
INDEX: The index of the screen (optional)

**Screen Hotspot INDEX, X, Y**
Set the hot-spot of a screen to a given coordinate. The hot-spot is the position within the screen where the Screen Offset will have effect and around which rotation will be done
*Parameters:*
INDEX: The index of the screen (optional)
X: The horizontal position of the hot-spot
Y: The vertical position of the hot-spot

**Screen Hotspot INDEX, FLAGS**
Set the hot-spot of a screen to a given coordinate. The hot-spot is the position within the screen where the Screen Offset will have effect and around which rotation will be done
*Parameters:*
INDEX: The index of the screen (optional)
FLAGS: Flag of bits indicating the horizontal and vertical position of the hot-spot, 0: top or left, 1: center or middle, 2: right or bottom. Example: $11 centers the hot-spot horizontally and vertically

**Screen Rotate INDEX, ANGLE**
TOTEST! Rotate a screen around it's hot-spot on display. Warning, on software rendererss this instruction will slow down the application
*Parameters:*
INDEX: The index of the screen
ANGLE: The angle of the rotation of the screen. Default in radian, and degrees after the "Degree" instruction has been used

**Screen Rotate ANGLE**
TOTEST! Rotate the current screen around it's hot-spot on display. Does not affect the content of the screen. Warning, on software rendererss this instruction will slow down the application

*Parameters:*

ANGLE: The angle of the rotation of the screen. Default in radian, and degrees after the "Degree" instruction has been used

**Screen Skew INDEX, XSKEW, YSKEW**

Applies a horizontal and vertical distortion to a screen during the display process

*Parameters:*

INDEX: The index of the screen

XSKEW: The number of pixel to shift at each horizontal pixel

YSKEW: The number of pixel to shift at each vertical pixel

**Screen Skew XSKEW, YSKEW**

Applies a horizontal and vertical distortion to the current screen during the display process. Does not affect the content of the screen

*Parameters:*

XSKEW: The number of pixel to shift at each horizontal pixel

YSKEW: The number of pixel to shift at each vertical pixel

**Screen Scale INDEX, XSCALE, YSCALE**

Resize a screen during the display process. Does not affect the content of the screen

*Parameters:*

INDEX: The index of the screen

XSCALE: The horizontal scale. 1= no effect, 0.5= half the width, 2= twice the width, etc.

YSCALE: The vertical scale. 1= no effect, 0.5= half the height, 2= twice the height, etc.

**Screen Scale XSCALE, YSCALE**

Resize the current screen during the display process. Does not affect the content of the screen

*Parameters:*

XSCALE: The horizontal scale. 1= no effect, 0.5= half the width, 2= twice the width, etc.

YSCALE: The vertical scale. 1= no effect, 0.5= half the height, 2= twice the height, etc.

**Screen Copy To SOURCEINDEX, X1, Y1, X2, Y2, DESTINATIONINDEX, X3, Y3, X4, Y4, MODE**

Copy an area from one screen to another or itself, resizing the area

*Parameters:*

SOURCEINDEX: The index of the source screen

X1: The horizontal coordinate of the top-left corner of the origin area to copy

Y1: The vertical coordinate of the top-left corner of the origin area to copy

X2: The horizontal coordinate of the bottom-right corner of the origin area to copy

Y2: The vertical coordinate of the bottom-right corner of the origin area to copy

DESTINATIONINDEX: The index of the destination screen

X3: The horizontal coordinate of the top-left corner of the destination area

Y3: The vertical coordinate of the top-left corner of the destination area

X4: The horizontal coordinate of the bottom-right corner of the destination area

Y4: The vertical coordinate of the bottom-right corner of the destination area

MODE: TODO! The drawing mode to use while drawing

**Screen Copy To SOURCEINDEX, X1, Y1, X2, Y2, DESTINATIONINDEX, X3, Y3, MODE**

Copy an area from one screen to another or itself, preserving the proportion of the area

*Parameters:*

SOURCEINDEX: The index of the source screen

X1: The horizontal coordinate of the top-left corner of the origin area to copy

Y1: The vertical coordinate of the top-left corner of the origin area to copy

X2: The horizontal coordinate of the bottom-right corner of the origin area to copy
Y2: The vertical coordinate of the bottom-right corner of the origin area to copy
DESTINATIONINDEX: The index of the destination screen
X3: The horizontal coordinate of the top-left corner of the destination area
Y3: The vertical coordinate of the top-left corner of the destination area
MODE: TODO! The drawing mode to use while drawing

**Screen Copy To SOURCEINDEX, DESTINATIONINDEX, MODE**

Copy one screen to another

*Parameters:*

SOURCEINDEX: The index of the source screen
DESTINATIONINDEX: The index of the destination screen
MODE: TODO! The drawing mode to use while drawing

**Screen Copy SOURCEINDEX, SX, SY, SWIDTH, SHEIGHT, DESTINATIONINDEX, DX, DY, DWIDTH, DHEIGHT, MODE**

TOTEST! Copy an area from one screen to another or itself, resizing the area

*Parameters:*

SOURCEINDEX: The index of the source screen
SX: The horizontal coordinate of the top-left corner of the origin area to copy
SY: The vertical coordinate of the top-left corner of the origin area to copy
SWIDTH: The width of the source area
SHEIGHT: The height of the source area
DESTINATIONINDEX: The index of the destination screen
DX: The horizontal coordinate of the top-left corner of the destination area
DY: The vertical coordinate of the top-left corner of the destination area
DWIDTH: The width of the ddestination area
DHEIGHT: the height of the destination area
MODE: TODO! The drawing mode to use while drawing

**Screen Copy SOURCEINDEX, SX, SY, SWIDTH, SHEIGHT, DESTINATIONINDEX, DX, DY, MODE**

TOTEST! Copy an area from one screen to another or itself, preserving the size of the area

*Parameters:*

SOURCEINDEX: The index of the source screen
SX: The horizontal coordinate of the top-left corner of the origin area to copy
SY: The vertical coordinate of the top-left corner of the origin area to copy
SWIDTH: The horizontal coordinate of the bottom-right corner of the origin area to copy
SHEIGHT: The vertical coordinate of the bottom-right corner of the origin area to copy
DESTINATIONINDEX: The index of the destination screen
DX: The horizontal coordinate of the top-left corner of the destination area
DY: The vertical coordinate of the top-left corner of the destination area
MODE: TODO! The drawing mode to use while drawing

**X Screen INDEX, X**

Convert a hardware horizontal coordinate into a screen coordinate. Hardware coordinates are different from screen coordinate only in retro-machine display emulation (Amiga, Atari etc.)

*Parameters:*

INDEX: The index of the screen
X: The hardware coordinate to convert

*Value returned:*

integer: The correponding horizontal coordinate in the Screen

**X Screen INDEX, X**

Convert a hardware horizontal coordinate into a coordinate in the current active screen. Hardware coordinates are different from screen coordinate only in retro-machine display emulation (Amiga, Atari etc.)

*Parameters:*

INDEX: The index of the screen

X: The hardware coordinate to convert

*Value returned:*

integer: The correponding horizontal coordinate in the current screen

**Y Screen INDEX, Y**

Convert a hardware vertical coordinate into a screen coordinate. Hardware coordinates are different from screen coordinate only in retro-machine display emulation (Amiga, Atari etc.)

*Parameters:*

INDEX: The index of the screen

Y: The hardware coordinate to convert

*Value returned:*

integer: The correponding vertical coordinate in the given screen

**Y Screen INDEX, Y**

Convert a hardware vertical coordinate into a coordinate in the current active screen. Hardware coordinates are different from screen coordinate only in retro-machine display emulation (Amiga, Atari etc.)

*Parameters:*

INDEX: The index of the screen

Y: The hardware coordinate to convert

*Value returned:*

integer: The correponding vertical coordinate in the current screen

**Def Scroll To INDEX, X1, Y1, X2, Y2, DX, DY**

Define a new scrolling zone in the current screen. This instruction has no visible effect until a "Scroll" instruction is used

*Parameters:*

INDEX: The index of the scrolling zone to define

X1: The horizontal coordinate of the top-left pixel of the rectangle to scroll

Y1: The vertical coordinate of the top-left pixel of the rectangle to scroll

X2: The horizontal coordinate of the bottom-right pixel of the rectangle to scroll

Y2: The vertical coordinate of the bottom-right pixel of the rectangle to scroll

DX: Signed horizontal shift to apply. 0= no scroll, -1= one pixel to the left, 1= one pixel to the right, etc

DY: Signed vertical shift to apply. 0= no scroll, -1= one pixel to the top, 1= one pixel to the bottom, etc

**Def Scroll INDEX, X, Y, WIDTH, HEIGHT, DX, DY**

Define a new scrolling zone in the current screen. This instruction has no visible effect until a "Scroll" instruction is used

*Parameters:*

INDEX: The index of the scrolling zone to define

X: The horizontal coordinate of the top-left pixel of the rectangle to scroll

Y: The vertical coordinate of the top-left pixel of the rectangle to scroll

WIDTH: The width in pixels of the rectangle to scroll

HEIGHT: The height in pixels of the rectangle to scroll

DX: Signed horizontal shift to apply. 0= no scroll, -1= one pixel to the left, 1= one pixel to the right, etc

DY: Signed vertical shift to apply. 0= no scroll, -1= one pixel to the top, 1= one pixel to the bottom, etc

**Scrollt INDEX**

Performs the action of scrolling for a pre-defined scrolling area. Moves the pixels in the desired direction. The empty zones at the extremities of the scrollign area, on left, top and/or right and bottom are left unchanged and will have to be

cleared
*Parameters:*
INDEX: The index of the scrolling area as defined with "Def Scroll"

**Dual Playfield SCREEN1, SCREEN2**
Associate the display of two screens into a parallax display. (Deprecated, use "Set Transparent" and "Screen Offset" to associate more than two screens together)
*Parameters:*
SCREEN1: The index of first screen, will be on top
SCREEN2: The index of first screen, will be in the back

**Dual Priority SCREEN1, SCREEN2**
Set the display priority of two screens associate in Dual Playfield. (Deprecated, use "Set Transparent" and "Screen Offset" to associate more than two screens together, and then "Screen To Front" or "Screen To Back" to handle the display priority)
*Parameters:*
SCREEN1: The index of first screen, will be on top
SCREEN2: The index of first screen, will be in the back

**Phybase INDEX**
Return the address of the first pixel in the bitmap buffer of the current screen (deprecated, there is no difference between logical and physical screen buffers in AOZ)
*Parameters:*
INDEX: The index of the screen
*Value returned:*
integer: A magical number representing the adress of the buffer, to be used later with "Poke" / "Doke" / "Loke" / "Peek" / "Deek" / "Leek" instructions. Not a real address in AOZ

**Logbase INDEX**
Return the address of the first pixel in the bitmap buffer of the current screen (deprecated, there is no difference between logical and physical screen buffers in AOZ)
*Parameters:*
INDEX: The index of the screen
*Value returned:*
integer: A magical number representing the adress of the buffer, to be used later with "Poke" / "Doke" / "Loke" / "Peek" / "Deek" / "Leek" instructions. Not a real address in AOZ

**Physic INDEX**
TODO! Return a magical number representing the physical buffer of the current screen, to be used in "Screen Copy" and all Screen instructions that necessitate a Screen Index. (deprecated, no such thing as physical or logic al screen in AOZ)
*Parameters:*
INDEX: The index of the screen
*Value returned:*
integer: A magical number representing the physical buffer of the screen. To be used wherever you need a screen index

**Logic INDEX**
TODO! Return a magical number representing the logical buffer of the current screen, to be used in "Screen Copy" and all Screen instructions that necessitate a Screen Index. (deprecated, no such thing as physical or logic al screen in AOZ)
*Parameters:*
INDEX: The index of the screen

*Value returned:*

integer: A magical number representing the logical buffer of the screen. To be used wherever you need a screen index

**Autoback INDEX, MODE**

Set the autoback background preservation system for graphical instruction (Deprecated, has no effect in AOZ)

*Parameters:*

INDEX: The index of the screen

MODE: A number from 0 to 2 included representing the mode to use

**Appear SOURCE, DESTINATION, PIXELS, RANGE**

TODO! Progressively draw one screen into another using a fading effect

*Parameters:*

SOURCE: Index of the source screen

DESTINATION: Index of the destination screen

PIXELS: Value used to perform the effect

RANGE: Range of the apparition in number of pixels (optional)

**Zoom ... To ... SOURCE, X1, Y1, X2, Y2, DESTINATION, X3, Y3, X4, Y4**

TODO! Copy and scale a rectangle from one screen to another.

*Parameters:*

SOURCE: Index of the source screen

X1: Horizontal coordinate of the top-left corner of the rectangle to zoom in the source screen

Y1: Vertical coordinate of the top-left corner of the rectangle to zoom in the source screen

X2: Horizontal coordinate of the bottom-right corner of the rectangle to zoom in the source screen

Y2: Vertical coordinate of the bottom-right corner of the rectangle to zoom in the source screen

DESTINATION: Index of the destination screen

X3: Horizontal coordinate of the top-left corner of the rectangle to zoom in the destination screen

Y3: Vertical coordinate of the top-left corner of the rectangle to zoom in the destination screen

X4: Horizontal coordinate of the bottom-right corner of the rectangle in the destination screen

Y4: Vertical coordinate of the bottom-right corner of the rectangle in the destination screen

**Zoom SOURCE, SX, SY, SWIDTH, SHEIGHT, DESTINATION, DX, DY, DWIDTH, DHEIGHT**

TODO! Copy and scale a rectangle from one screen to another.

*Parameters:*

SOURCE: Index of the source screen

SX: Horizontal coordinate of the top-left corner of the rectangle to zoom in the source screen

SY: Vertical coordinate of the top-left corner of the rectangle to zoom in the source screen

SWIDTH: Width of the rectangle to zoom in the source screen

SHEIGHT: Height of the rectangle to zoom in the source screen

DESTINATION: Index of the destination screen

DX: Horizontal coordinate of the top-left corner of the rectangle in the destination screen

DY: Vertical coordinate of the top-left corner of the rectangle in the destination screen

DWIDTH: Width of the rectangle to zoom in the destination screen

DHEIGHT: Height of the rectangle to zoom in the destination screen

**XGr**

Return the current horizontal coordinate of the graphical cursor in the current screen

**XGr**

Return the current vertical coordinate of the graphical cursor in the current screen

**Reserve Zone NUMBER**
Reserve memory to store graphical detection zones in the current screen (deprecated: number of zonme is unlimited in AOZ)
*Parameters:*
NUMBER: (Optional) The number of zones to allocate. If ommited all zones will be erased (not deprecated without parameter)

**ScIn INDEX, X, Y**
Check if the given hardware coordinates are located above a given screen on display. Hardware coordinate are only different from screen coordinate for retro-machine emulation (Amiga, Atari etc.)
*Parameters:*
INDEX: The index of the screen to test
X: Horizontal hardware coordinate to test
Y: Vertical hardware coordinate to test
*Value returned:*
boolean: True if the given coordinate lay over the screen, False if not

**ScIn X, Y**
Find the top-most screen on display located under the given hardware coordinates. Hardware coordinate are only different from screen coordinate for retro-machine emulation (Amiga, Atari etc.)
*Parameters:*
X: Horizontal hardware coordinate to test
Y: Vertical hardware coordinate to test
*Value returned:*
integer: -1 if the coordinate lay outside of all the screen, or the index of the top-most screen if they lay inside of one

**Mouse Screen**
Return the index of the screen under the mouse
*Value returned:*
integer: -1 if the coordinate lay outside of all the displayed screens, or the index of the top-most screen if they lay inside of one

**Screen Colour INDEX**
Return the number of colours in the palette of a given screen
*Parameters:*
INDEX: The index of the screen to query
*Value returned:*
integer: The number of colours in the palette

**Screen Colour**
Return the number of colours in the palette of the current screen
*Value returned:*
integer: The number of colours in the palette of the curren screen

**Screen Base**
Not implemented, deprecated, will return 0
*Value returned:*
integer: Return 0

**Screen Width INDEX**

Return the width in pixels of a given screen
*Parameters:*
INDEX: The index of the screen to query
*Value returned:*
integer: The width of the screen in pixel

**Screen Width**
Return the width in pixels of the current screen
*Value returned:*
integer: The width of the current screen in pixel

**Screen Height INDEX**
Return the height in pixels of a given screen
*Parameters:*
INDEX: The index of the screen to query
*Value returned:*
integer: The height of the screen in pixel

**Screen Height**
Return the height in pixels of the current screen
*Value returned:*
integer: The height of the current screen in pixel

**Screen INDEX**
Set the given screen index as the current screen, all graphical operation being directed to this screen after this instruction
*Parameters:*
INDEX: The index of the screen

**Screen**
Return the index of the current screen, -1 if no screen is opened when the function is called
*Value returned:*
integer: The index of the current screen

**Hires**
Return a magical number to be used in the "Screen Open" instruction, enforce a horizontal compression by half of the pixels
*Value returned:*
integer: 1

**Lowres**
Return a magical number to be used in the "Screen Open" instruction, display pixel with their original horizontal ratio
*Value returned:*
integer: 0

**Laced**
Return a magical number to be used in the "Screen Open" instruction, enforce a vertical compression by half of the pixels
*Value returned:*
integer: 2

**Halfbright**
Return a magical number to be used in the "Screen Open" instruction, and set the screen in Amiga-compatible Halfbright colour mode
*Value returned:*
integer: 4

**X Hard**
TOTEST! Converts a horizontal coordinate in a given screen to its equivalent in hardware coordinates taking into account the position, scale and rotation factor of the screen
*Value returned:*
integer: The hardware equivalent of the horizontal screen coordinate

**X Hard**
TOTEST! Converts a horizontal coordinate in the current screen to its equivalent in hardware coordinates taking into account the position, scale and rotation factor of the screen
*Value returned:*
integer: The hardware equivalent of the horizontal current screen coordinate

**Y Hard**
Converts a vertical coordinate in a given screen to its equivalent in hardware coordinates, taking into account the position, scale and rotation factor of the screen
*Value returned:*
integer: The hardware equivalent of the vertical screen coordinate

**Y Hard**
TOTEST! Converts a vertical coordinate in the current screen to its equivalent in hardware coordinates taking into account the position, scale and rotation factor of the screen
*Value returned:*
integer: The hardware equivalent of the vertical current screen coordinate

**Alpha INDEX**
Return the transparency Alpha value of a given colour index in the current screen palette
*Parameters:*
INDEX: The index of the coloour in the colour palette
*Value returned:*
float: The alpha value, from 0 (fully transparent) to 1 (fully opaque)

**Remap ... To ... SCOLOR, DCOLOR, X1, Y1, X2, Y2**
Transforms the values of all the pixels in a rectangle matching a specific RGB value to another RGBA value. Warning, this operation can take a long time to process and make the browser irresponsive during a while
*Parameters:*
SCOLOR: The RGB value of the color to look for
DCOLOR: The RGBA value of the color to replace with
X1: The horizontal coordinate of the top-left corner of the origin area to scan
Y1: The vertical coordinate of the top-left corner of the origin area to scan
X2: The horizontal coordinate of the bottom-right corner of the origin area to scan
Y2: The vertical coordinate of the bottom-right corner of the origin area to scan

**Remap ... To ... SCOLOR, DCOLOR, X, Y, WIDTH, HEIGHT**
Transforms the values of all the pixels in a rectangle matching a specific RGB value to another RGBA value. Warning, this operation can take a long time to process and make the browser irresponsive during a while

*Parameters:*

SCOLOR: The RGB value of the color to look for

DCOLOR: The RGBA value of the color to replace with

X: The horizontal coordinate of the top-left corner of the origin area to scan

Y: The vertical coordinate of the top-left corner of the origin area to scan

WIDTH: The horizontal coordinate of the bottom-right corner of the origin area to scan

HEIGHT: The vertical coordinate of the bottom-right corner of the origin area to scan

**Remap SCOLOR, DCOLOR**

Transforms the values of all the pixels in the current screen matching a specific RGB value to another RGBA value. Warning, this operation can take a long time to process and make the browser irresponsive during a while

*Parameters:*

SCOLOR: The RGB value of the color to look for

DCOLOR: The RGBA value of the color to replace with

**Screen Mode**

Return the "mode" parameter of a screen, as used in the "Screen Open" instruction

*Value returned:*

integer: integer:The magical number generated by the combination of Lowres, Hires, Laced or Halfbright

**Screen Hot Spot X, Y**

Set the hot-spot of the current screen. The pixel displayed at the coordinates set by "Screen Offset" will be located at these coordinates inside of the screen, and the screen will be shifted on the display

*Parameters:*

X: The horizontal coordinate of the hot-spot

Y: The vertical coordinate of the hot-spot

**Screen Hot Spot INDEX, X, Y**

Set the hot-spot of the given screen. The pixel displayed at the coordinates set by "Screen Offset" will be located at these coordinates inside of the screen, and the screen will be shifted on the display

*Parameters:*

INDEX: The indexd of the screen to modify

X: The horizontal coordinate of the hot-spot

Y: The vertical coordinate of the hot-spot

**Screen Hot Spot FLAGS**

Calculates the hot-spot of the current screen

*Parameters:*

FLAGS: Flag of bits indicating the horizontal and vertical position of the hot-spot, 0: top or left, 1: center or middle, 2: right or bottom. Example: $11 centers the hot-spot horizontally and vertically

**Double Buffer**

Turns the display system into double-buffering. Deprecated: Double Buffering has no effect in AOZ

**Set Transparent ..., ..., ... COLOR_INDEXES**

Set one or more colours of the current screen as transparent. Will invoke a complete remapping of the screen that migght take a long time. Use this instruction to make parralax scrolling diplays

*Parameters:*

COLOR_INDEXES: A list of indexes of colours in the screen palette eparated by a comma

**Stop Transparent COLOR_INDEXES**

Set one or more colours of the current screen as opaque. Will invoke a complete remapping of the screen that migght take a long time.
*Parameters:*
COLOR_INDEXES: A list of indexes of colours in the screen palette eparated by a comma

**Set Alpha ... To ..., ... To ... INDEXES, ALPHA#**
Set the alpha of one or more colours in the current screen. This effect of this instruction will only be visible when the next graphical instruction using this coloour are executed.
*Parameters:*
INDEXES: The index of the colour in the screen palette
ALPHA#: The value of the alpha, 0= totally opaque, 1= totally transparent (colour will not be visible and leave "holes" in the screen)