# Documentation

https://github.com/halflkaka/Layerplus

# CImg

## Namespaces

cimg_library, cimg

## Classes

- struct CImg

  Class representing an image (up to 4 dimensions wide), each pixel being of type T

- struct CImgList

  Represent a list of images CImg

- struct CImgDisplay

  Allow the creation of windows, display images on them and manage user events (keyboard, mouse and windows events).

- struct CImgException

  Instances of CImgException are thrown when errors are encountered in a CImg function call.

## Detailed Description

Contains all classes and functions of the CImg library.

This namespace is defined to avoid functions and class names collisions that could happen with the inclusion of other C++ header files. Anyway, it should not happen often and you should reasonnably start most of your CImg-based programs with

```
#include "CImg.h"
using namespace cimg_library;
```

# LayerPlus

## Namespaces

cimg_extension

## Classes

- struct Layer

  Class representing a layer, containing a CImg instance of type T

- struct Layer_System

  Class of a layer processing system. Allow the manipulation on layers

## Detailed Description

Contains all classes and functinos of the LayerPlus library. Start the programs with

```
#include "Layer.h"
```

# Layer<T> Struct Template

Class representing a layer, containing a CImg instance of type T

## Private Elements

- CImg<T>* _data
- bool _is_visible

## Public Types

- typedef T* iterator
- typedef const T* const_iterator
- typedef T value_type

### Constructor/Destructor

- **~Layer()**

  Destroy layer

- **Layer()**

  Construct empty layer

- **Layer(const CImg& img)**

  Construct layer from loading a CImg instance, the layer is visible by default.
  Example: `Layer<float> layer(img);`

- **Layer(const CImg& img, const bool is_visible)**

  Construct layer from loading a CImg instance, set layer visibility to is_visible.
  Example: `Layer<float> layer(img, false); //construct an invisible layer`

### Instance Characteristics

- **void set_visible()**

  Set current layer to be visible

- **void set_invisible()**

  Set current layer to be invisble

- **bool visible()**

  Visibility of current layer

- **CImg data()**

  CImg instance binded with the layer

## Layer_System<T,N> Struct Template

Class of a layer processing system. Allow the manipulation on layers.

### Private Elements

- Layer<T> _layers[N];
- std::size_t index;

## Public Types

- typedef Layer<T>* iterator
- typedef const Layer<T>* const_iterator
- typedef Layer<T> value_type
- typedef Layer<T>& reference;
- typedef const Layer<T>& const_reference;
- typedef std::size_t size_type;

## Constructor/Destructor

- **~Layer_System()**

  Destroy layer system

- **Layer_System()**

  Construct a layer system with no layer

## Instance Characteristics

- **Layer<T>& front()**

  Layer at the bottom of the layers

- **Layer<T>& back()**

  Layer at the top of the layers

- **size_t size()**

  Size of the total layers

- **size_t get_index()**

  Index of current inserted layer

- **Layer<T>* data()**

  Return array of all the layers

- **Layer<T> data(size_t pos)**

  Return layer at pos position

## Overloaded Operators

- **Layer<T>& operator[](size_t pos)**

  Return layer at pos position

## Operations

- **void add_layer(Layer<T> layer)**

  Add a new layer on top of the layers
  Example: `sys.add_layer(layer)`

- **void remove_layer()**

  Remove the top layer
  Example: `sys.remove_layer()`

- **Layer<T> get_top_layer()**

  Get top layer

- **Layer<T>* smooth_layer(Layer<T> layer, const int index, const int iter)**

  Extract the CImg instance of the layer and smooth it by *iter* times, select the CImg on index position. Return a new layer containing the smooth CImg instance.
  Example: `Layer<float> smooth_layer = *(sys.smooth_layer(layer, 50, 100));`

- **Layer<T>* blur_gradient_layer(Layer<T> layer, const double sigma)**

  Blur CImg instance of the layer with a sigma variance. Return a new layer containing the blur CImg instance.
  Example: `Layer<float> blur_layer = *(sys.blur_gradient_layer(layer, 5));`

- **Layer<T>* exposure_layer(Layer<T> layer, const double gamma)**

  Adjust CImg instance's brightness and return a new layer containing the modified CImg instance.
  Example: `Layer<float> exposure_layer = *(sys.exposure_layer(layer, 0.5));`

- **void set_visible(Layer<T>& layer)**

Set layer to be visible

- **void set_invisible(Layer<T>& layer)**

  Set layer to be invisible

- **Layer<T>* merge_layer()**

  Merge all layers, return a new merged layer.
  Example: `Layer<float> merge_layer = *(sys.merge_layer());`