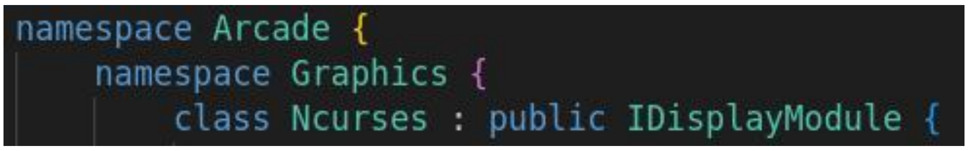
How To : Implement a new graphical lib

1) Create a new class that inherits from IDisplayModule



```
namespace Arcade {
    namespace Graphics {
        class Ncurses : public IDisplayModule {
```

2) Implement IDisplayModule's methods

• void setFrameRateLimit (int limit)

Set the window's Frame Rate Limit.

• void putpixel (int x, int y, unsigned int color)

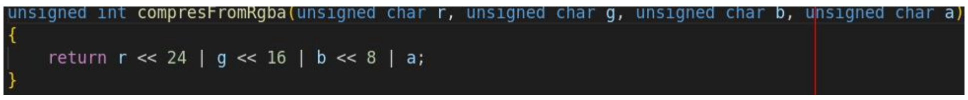
Put a pixel on the window.

x : x coordinate

y : y coordinate

color : color

Use this to compress an RGBA color to an int ;



```
unsigned int compresFromRgba(unsigned char r, unsigned char g, unsigned char b, unsigned char a)
{
    return r << 24 | g << 16 | b << 8 | a;
}
```

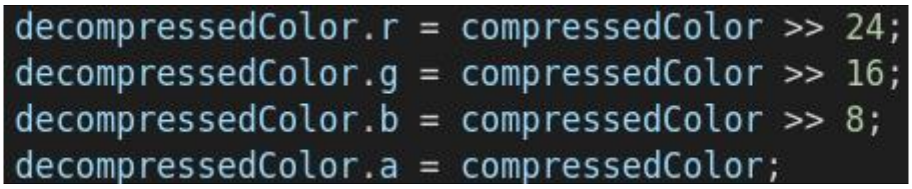
R : Red color intensity (0 – 255)

B : Blue color intensity (0 – 255)

G : Green color intensity (0 – 255)

A : Alpha (0 – 255)

And decompress it this way :



```
decompressedColor.r = compressedColor >> 24;
decompressedColor.g = compressedColor >> 16;
decompressedColor.b = compressedColor >> 8;
decompressedColor.a = compressedColor;
```

• void puttext (int x, int y, unsigned int color, const std::string &str)

Put text on the window.

x : x coordinate

y : y coordinate

color : color

str : text to put on the window

• std::vector<Events >getEvents ()

Get the window's events.

• std::pair<int, int >getMousePos ()

Get the Mouse Position.

• bool isOpen ()

Get the window's status. True if opened, false otherwise.

• void clearWin ()

Clear the window.

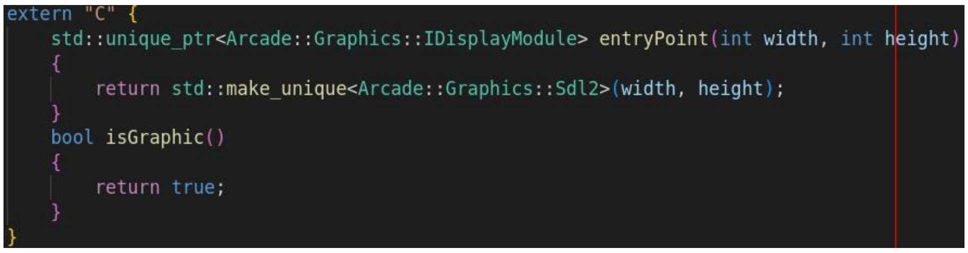
• void refreshWin ()

Refresh the window.

3) Add your class' constructor and destructor

Your class' constructor needs to initialize a window and open it and its destructor needs to close it.

4) Add entryPoint and isGraphic functions



```
extern "C" {
    std::unique_ptr<Arcade::Graphics::IDisplayModule> entryPoint(int width, int height)
    {
        return std::make_unique<Arcade::Graphics::Sdl2>(width, height);
    }
    bool isGraphic()
    {
        return true;
    }
}
```

EntryPoint is called by the Arcade's Core and needs to return an instance of your graphical class

IsGraphic returns true

5) Compile your graphical lib

g++ -o lib/arcade_${GraphicalLibName}.so ${GraphicalLibFiles} -I ${GraphicalLibCompilerFlags} -shared -fPIC