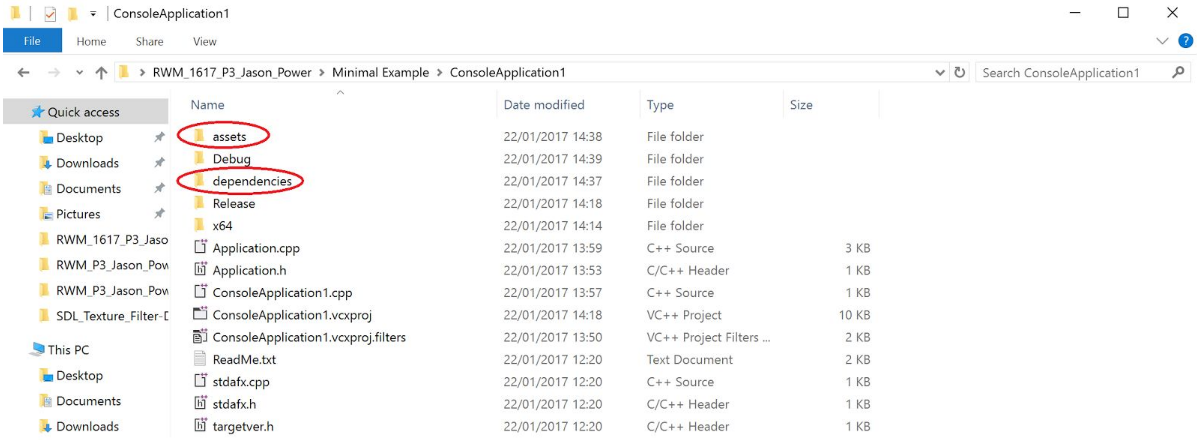
Setting up component

Create a new C++ Win32 Console Application project

Place the dependencies and assets folder into your project where the .cpp and .h files / src and include folders can be found.



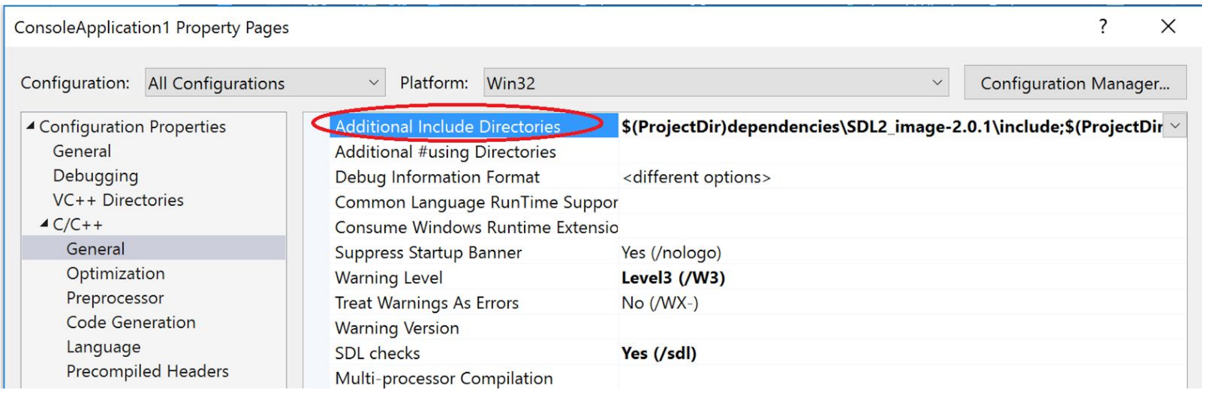
Name	Date modified	Type	Size
assets	22/01/2017 14:38	File folder	
Debug	22/01/2017 14:39	File folder	
dependencies	22/01/2017 14:37	File folder	
Release	22/01/2017 14:18	File folder	
x64	22/01/2017 14:14	File folder	
Application.cpp	22/01/2017 13:59	C++ Source	3 KB
Application.h	22/01/2017 13:53	C/C++ Header	1 KB
ConsoleApplication1.cpp	22/01/2017 13:57	C++ Source	1 KB
ConsoleApplication1.vcxproj	22/01/2017 14:18	VC++ Project	10 KB
ConsoleApplication1.vcxproj.filters	22/01/2017 13:50	VC++ Project Filters ...	2 KB
ReadMe.txt	22/01/2017 12:20	Text Document	2 KB
stdafx.cpp	22/01/2017 12:20	C++ Source	1 KB
stdafx.h	22/01/2017 12:20	C/C++ Header	1 KB
targetver.h	22/01/2017 12:20	C/C++ Header	1 KB

Add the following to the Additional Include Directories in C++/General

$(ProjectDir)dependencies\SDL2_image-2.0.1\include;

$(ProjectDir)dependencies\SDL\include;

$(ProjectDir)dependencies\SDL_Texture_Filter\include;



Configuration: All Configurations

Platform: Win32

Additional Include Directories	$(ProjectDir)dependencies\SDL2_image-2.0.1\include;$(ProjectDir
Additional #using Directories	
Debug Information Format	<different options>
Common Language RunTime Suppor	
Consume Windows Runtime Extensio	
Suppress Startup Banner	Yes (/nologo)
Warning Level	**Level3 (/W3)**
Treat Warnings As Errors	No (/WX-)
Warning Version	
SDL checks	**Yes (/sdl)**
Multi-processor Compilation	

Add the following to the Additional Library Directories in Linker

Note – Use the x86 files for Win32

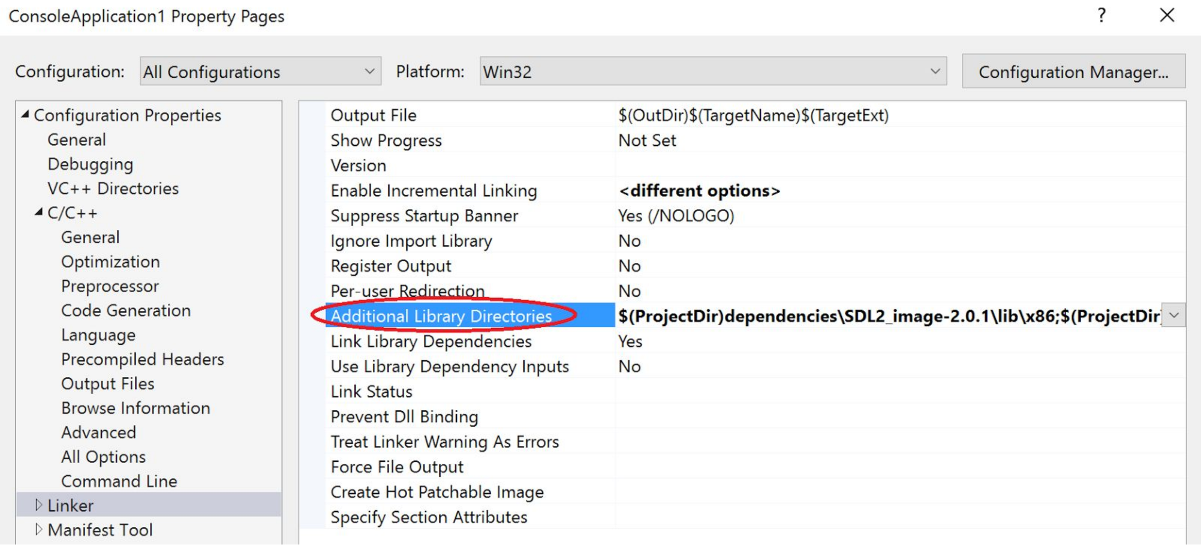
– Use the x64 files for x64

x86 example

```
$(ProjectDir)dependencies\SDL2_image-2.0.1\lib\x86;

$(ProjectDir)dependencies\SDL\lib\x86;

$(ProjectDir)dependencies\SDL_Texture_Filter\lib\x86;
```

X64 example

```
$(ProjectDir)dependencies\SDL2_image-2.0.1\lib\x64;

$(ProjectDir)dependencies\SDL\lib\x64;

$(ProjectDir)dependencies\SDL_Texture_Filter\lib\x64;
```



ConsoleApplication1 Property Pages

Configuration: All Configurations   Platform: Win32

Output File	$(OutDir)$(TargetName)$(TargetExt)
Show Progress	Not Set
Version	
Enable Incremental Linking	**<different options>**
Suppress Startup Banner	Yes (/NOLOGO)
Ignore Import Library	No
Register Output	No
Per-user Redirection	No
Additional Library Directories	**$(ProjectDir)dependencies\SDL2_image-2.0.1\lib\x86;$(ProjectDir**
Link Library Dependencies	Yes
Use Library Dependency Inputs	No
Link Status	
Prevent Dll Binding	
Treat Linker Warning As Errors	
Force File Output	
Create Hot Patchable Image	
Specify Section Attributes	

Add the following to the Command Line in Build/Post-Build Event

Note – Use the x86 files for Win32

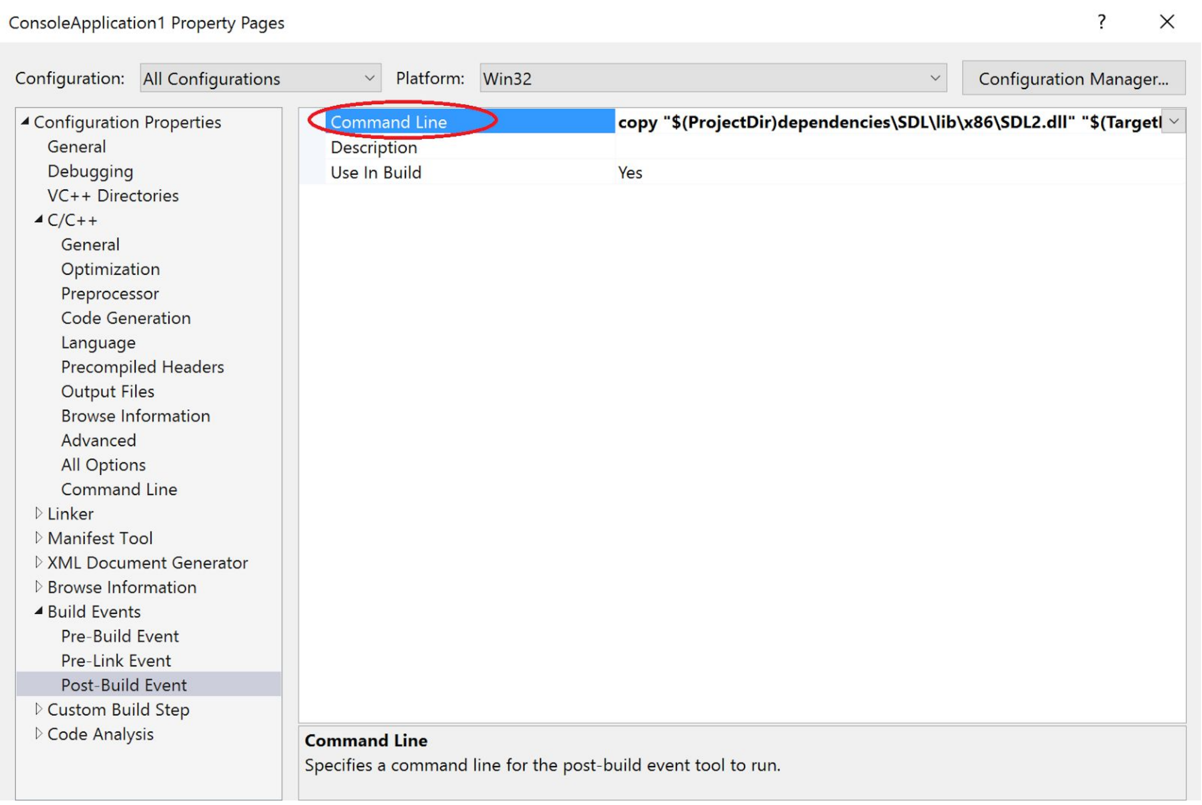
– Use the x64 files for x64

x86 example

```
copy "$(ProjectDir)dependencies\SDL\lib\x86\SDL2.dll" "$(TargetDir)"

copy "$(ProjectDir)dependencies\SDL2_image-2.0.1\lib\x86\*.dll" "$(TargetDir)"

copy "$(ProjectDir)dependencies\SDL_Texture_Filter\lib\x86\*.dll" "$(TargetDir)"

echoD|xcopy /E /Y "$(ProjectDir)assets" "$(TargetDir)assets"
```

x64 example

```
copy "$(ProjectDir)dependencies\SDL\lib\x64\SDL2.dll" "$(TargetDir)"

copy "$(ProjectDir)dependencies\SDL2_image-2.0.1\lib\x64\*.dll" "$(TargetDir)"

copy "$(ProjectDir)dependencies\SDL_Texture_Filter\lib\x64\*.dll" "$(TargetDir)"

echoD|xcopy /E /Y "$(ProjectDir)assets" "$(TargetDir)assets"
```



ConsoleApplication1 Property Pages

Configuration: All Configurations

Platform: Win32

Command Line: copy "$(ProjectDir)dependencies\SDL\lib\x86\SDL2.dll" "$(TargetI

Description

Use In Build: Yes

Command Line

Specifies a command line for the post-build event tool to run.

Using the component

Include the following to enable both debug and release builds of the library

```
#ifdef _DEBUG
#pragma comment(lib, "SDL_Texture_Filter-D.lib")
#else
#pragma comment(lib,"SDL_Texture_Filter.lib")
#endif
```

Include the following to use the component

```
#include "TextureFilter.h"
```

" I recommended that you create an enum of texture ID's as Jtextures are stored in a map and accessed / retrieved by passing in an int ID"

```
enum TextureID
{
	Player,
	Background
};
```

Load an asset

```
// CreateJTexture takes 3 parameters, Image Path, image ID, SDL_Renderer

TextureFilter::Instance()->createJtexture("assets/image0.png", 0, m_renderer);
```

Render an asset

The value 0 represents the ID of the texture which was used to create the texture

```
SDL_RenderCopy(m_renderer, TextureFilter::Instance()->getTexture(0), NULL, 
&TextureFilter::Instance()->getTextureBounds(0));
```

Apply a texture filter to an image

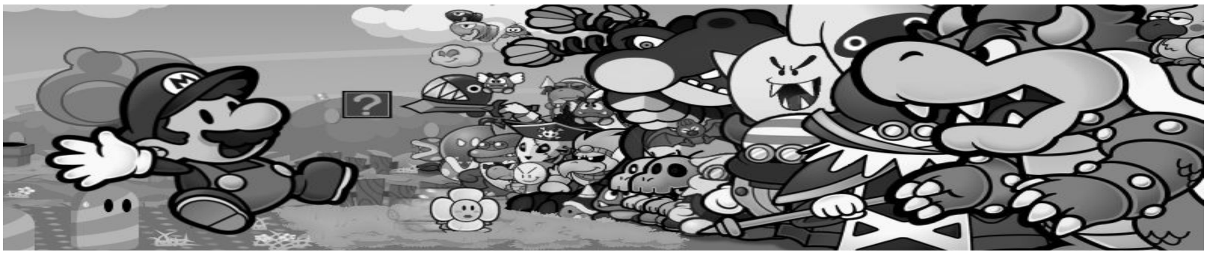
The first value 0 represents the ID of the texture which was used to create the texture

Resets the texture back to its original state

```
TextureFilter::Instance()->resetPixels(0);
```

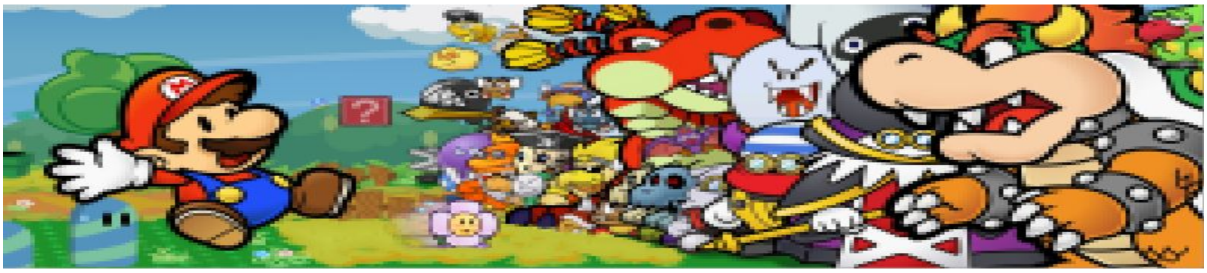
Applies grayscale filter to the image

```
TextureFilter::Instance()->grayscaleFilter(0);
```



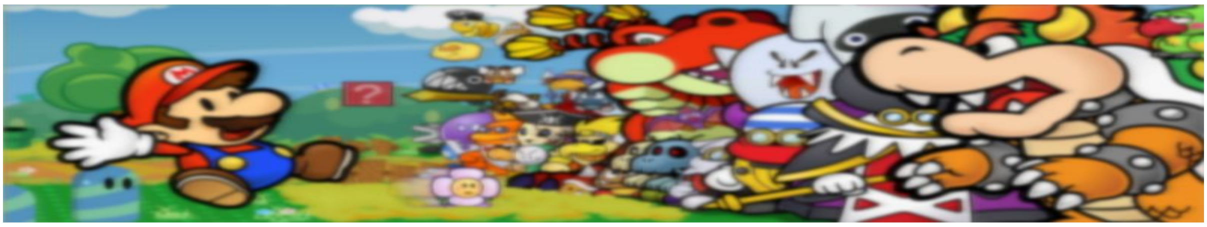
Applies pixelate filter to the image, pixel size value is passed as a parameter

```
// PIXEL SIZE: 1 - 100
TextureFilter::Instance()->pixelateFilter(0, 10);
```



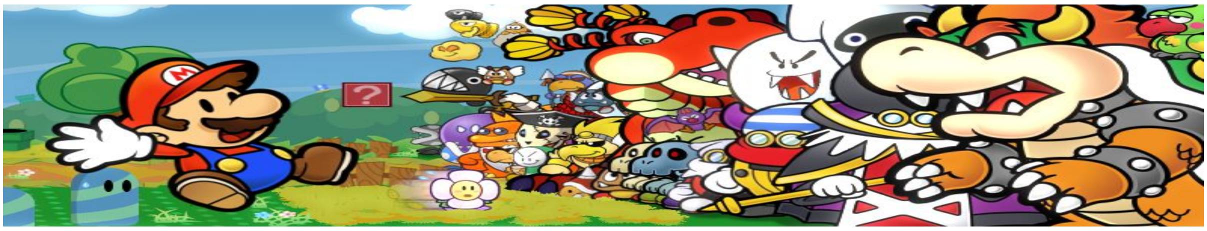
Applies blur filter to the image, blur radius is passed as a parameter

```
// BLUR RADIUS: 1 - 100
TextureFilter::Instance()->gaussianBlur1D(0, 15);
```



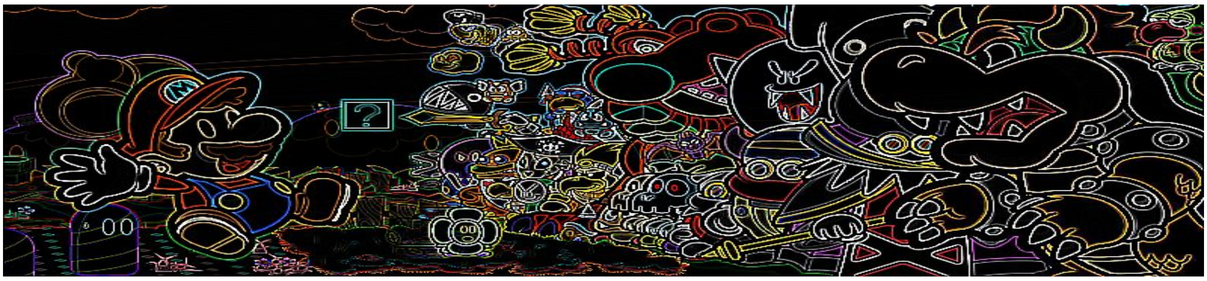
Applies bloom filter to the image, threshold, blur radius and additive blend are passed as parameters

```
// THRESHOLD: 1 - 255
// BLUR RADIUS: 1 - 100
// ADDITIVE BLEND: 0.1f – 1.0f
TextureFilter::Instance()->bloomFilter(0, 150, 20, 0.7f);
```



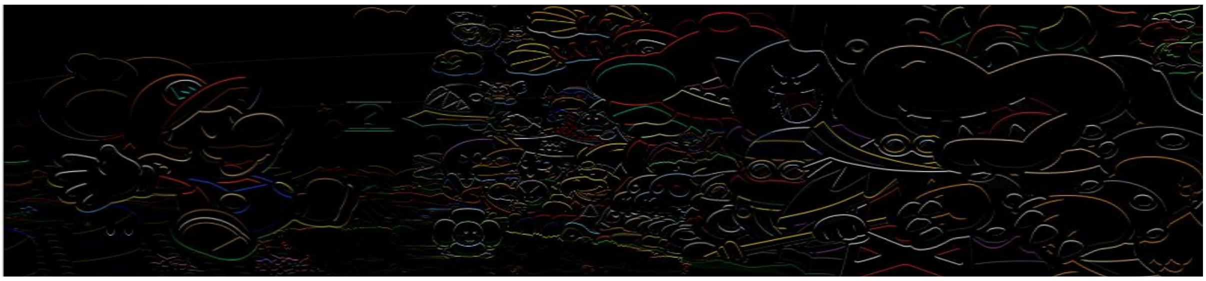
Applies edge detect filter to the image

```
TextureFilter::Instance()->edgeDectection(0, TextureFilter::EdgeDetect);
```



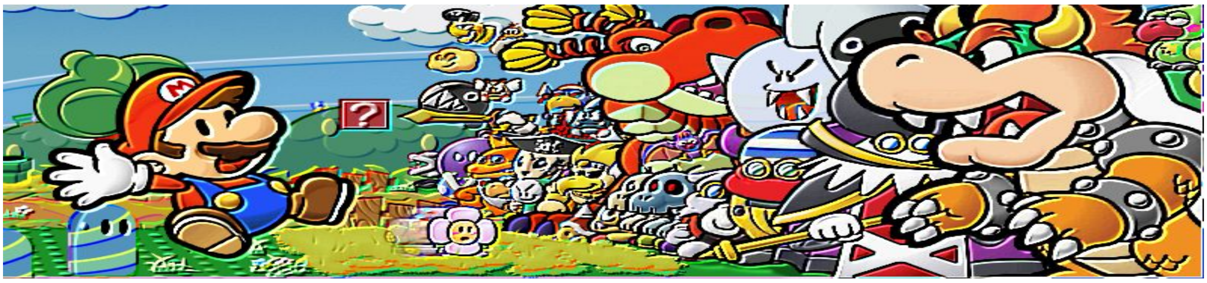
Applies edge enhance filter to the image

```
TextureFilter::Instance()->edgeDectection(0, TextureFilter::EdgeEnhance);
```



Applies emboss filter to the image

```
TextureFilter::Instance()->edgeDectection(0, TextureFilter::Emboss);
```



Please see the Minimal Example if there are any issues setting up the component or how to use it.

All files can be found here - [GitHub Link]