



本日はビデオを使ったアプリの作成







プロジェクトの作成

projectGeneratorを起動する

windows 版のパッケージ



macOS 版のパッケージ



空のプロジェクトの作成

× create/update	Project nam クトを作る (自分で設	ne は つ たびに 定して	プロ こ 変 こも	ジェ わる 可)
Project name:			_	
myGoodSketch	24	import		
Project path:				
<openframewor< th=""><td>ksの展開場所>¥apps¥</td><td>myApps</td><td>Q</td><td></td></openframewor<>	ksの展開場所>¥apps¥	myApps	Q	
Addons:	そのまま			
Addons		_	•	
Platforms:	空欄のま	ま		
Windows (Visual S	Studio 2017) 🗙		~	
	そのまま			
Generate プロジェク	フト作成			

- Project name:
 - 作成するプロジェクト(プログラム)の名前
- Project path:
 - 作成するプロジェクトのファイル を置く場所
 - openFrameworksのパッケージを展開した場所の中の apps¥myApps

プロジェクトの作成成功

	📙 💆 📙 🚽 myGoodSketch		_	
create / update	ファイル ホーム 共有 表示			~
	🗲 🔶 👻 🛧 🔤 « myApps » i	myGoodSketch	✓ ひ 𝒫 myGa	oodSketchの
Project name: 🕼		をダブ	種類	サイズ
myGoodSketch 24 import		リック **	ファイル フォルダー	
Project path:	addons.make	もよい 🖽	MAKE ファイル	o KB
	icon.rc	2020/00/21 22:10	Resource Script	1 KB
C.\oi_vo.rr.o_vs2orr_release\apps\myApps Q	myGoodSketch.sln	2020/07/24 9:43	Microsoft Visual S	3 KB
Addons:	wyGoodSketch.vcxproj	2020/07/24 9:43	VC++ Project	11 KB
	🚹 🛐 myGoodSketch.vcxproj.filters	2020/07/24 9:43	VC++ Project Filte	1 KB
Success! Your can now find your project in <openframeworksの展開場所>¥apps¥myApps¥myGoodSketch</openframeworksの展開場所>	🕢 myGoodSketch.vcxproj.user	2020/07/24 9:43	Per-User Project O	2 KB
<pre>[notice] [notice] setting OF path to: C:\of_v0.11.0_vs2017_release [notice] from -o option [notice] target platform is: vs [notice] project path is: C:\of_v0.11.0_vs2017_release\apps\myGoodSketch [notice] setting up new project C:\of_v0.11.0_vs2017_release\apps\myApps\myGoodSketch [notice] saving addons.make</pre>				
IDE で開く → Open in IDE Close	8 個の項目 1 個の項目を選択 2.00 K	(B		1

Visual Studio が起動する



ソリューションの再ターゲット

ソリューション操作の再ターゲット ×	
プロジェクトの再ターゲット	
次のプロジェクトは、以前のバージョンの Visual C++ プラットフォーム ツールセットを使用しています。これらのプロ ジェクトは、最新の Microsoft ツールセットをターゲットとするようにアップグレードできます。また、お使いのマシン にインストールされているものからターゲットの Windows SDK バージョンを選択することもできます。	
Windows SDK バージョン: 10.0 (最新のインストールされているバージョン) 、	
プラットフォーム ツールセット: v142 へのアップグレード v	
✓¥myGoodSketch¥myGoodSketch.vcxproj	
OK キャンセル	

Visual Studio は頻繁に更新しているので皆さんがお使いの Visual Studio SDK のバージョンと合わない場合がある

Visual Studio 起動 ビルド(B) デバッグ(D) ▼ ファイル(F) 編集(E) 表示(V) プロジェクト(P) テスト(S) 分析(N) ツール(T) 拡張機能(X) ウィンドウ(W) ヘルプ(H) 検索 (Ctrl+Q) 🔎 myG...etch 🖔 🖸 🗸 🗇 👘 👘 👘 🖓 🗸 🖓 🗸 🖓 🗸 🖓 🗸 🖓 🗸 🖓 Win32 - 🏓 🖾 🚽 Dーカル Windows デバッガー ・ 自動 Ŕ R Live Share ソリューション エクスプローラー ▼ ₽ × G D 🟠 🛗 - 🐻 - ≒ 🗗 🕼 🏓 🗕 ソリューション エクスプローラー の検索 (Ctrl+:) ۰ م 「フリューション 'myGoodSketch' (2/2 プロジェクト) Implement Provide the second secon openframeworksLib プロパテ 出力 - 🖆 🖆 🖉 🔁 出力元(S): 全般 プロジェクト 'myGoodSketch' をアップグレードしています... 構成 'Debug|Win32': ブラットフォーム ツールセットを 'v142' に変更しています(以前は 'v141')。 構成 'Debug|x64': ブラットフォーム ツールセットを 'v142' に変更しています(以前は 'v141')。 構成 'Release|Win32': ブラットフォーム ツールセットを 'v142' に変更しています(以前は 'v141')。 構成 'Release|x64': ブラットフォーム ツールセットを 'v142' に変更しています(以前は 'v141')。 構成 'Release|x64': ブラットフォーム ツールセットを 'v142' に変更しています(以前は 'v141')。 再ターゲットの終了: 完了 1、失敗 0、スキップ 0 ソリューション エ... クラス ビュー プロパティ マネ... チーム エクスプ... 4 11 ↑ ソース管理に追加 ▲ 🗇 準備完了

映像の入力

ビデオデータの取り扱い



ofApp クラスに映像入力のメンバ変数を追加する

#pragma once

#include "ofMain.h"

```
class ofApp : public ofBaseApp{
    ofVideoGrabber video;
```

public:

```
void setup();
void update();
void draw();
```

```
void keyPressed(int key);
void keyReleased(int key);
void mouseMoved(int x, int y);
(以下略)
```

ofVideoGrabber

 カメラからの映像入力(ビデオ キャプチャ)を行うクラス



ofApp.cpp で入力した映像を表示する

<pre>#include "ofApp.h"</pre>
<pre>// void ofApp::setup(){ video.setup(320, 240); }</pre>
<pre>// void ofApp::update(){ video.update(); }</pre>
<pre>// void ofApp::draw(){ video.draw(0, 0); }</pre>
(以下略)

video.setup(320, 240);

- 映像入力の初期設定
- 320, 240 は入力する映像の解像度
 - 640,480や1280,720に対応するかどうかはカメラ次第なので試して
- video.update();
 - 映像入力から1フレーム取り込む
- video.draw(0, 0);
 入力した映像の1フレームをウィンドウの原点(0,0、左上隅)から描画する



✓ ファイル(F) 編集(E) 表示(V) プロジェクト(P) ビルド	(B) デバッグ(D) テスト(S) 分析(N) ツール	(T) 拡張機能(X) 「	ウィンドウ(W) ヘルプ(H)	検索 (Ctrl+Q) 👂	myGetch	- 🗆	\times
🖁 🕒 🕶 🗎 🏜 🚔 💾 🖓 🗣 🖓 🕶 🖉 🗸 🖉 🔹	Win32 ・ トローカル Windows デパ	ッガ- ▼ 自動	• 🎜 🖾 🖕			🖻 Live Share	ጽ
ソリューション エクスプローラーの検索(Ctrl+:) シー ソリューション 'myGoodSketch' (2/2 プロジェクト) マリューション 'myGoodSketch' (2/2 プロジェクト) ・	<pre> MyGoodSketch</pre>	・ (グローバルスコ)); !(int key){	ユーブ)		ofA	App.cpp 🚡 🗙	サーバーエクスプローラー ツールボックス プロパティ
フリューション ユニー フリス しュー フロハリイ マネニー アーム エリス ノニー	100% ▼ ◎ 間題は見つかりませんでした				יתיו זינו <u>אר</u> <u>אר</u>	又子:1 SPC X管理に追加 ▲	

左上の原点を左上隅にして映像が表示される



「■キー」→「設定」→「プライバシー」

設定

ふ カメラ

このデバイスのカメラへのアクセスを許可する

アクセスを許可すると、このデバイスを使用するユーザーはこのページの設定を使用し て、アプリにカメラへのアクセスを許可するかどうかを選択できます。アクセスを拒否す ると、Microsoft Store アプリや、ほとんどのデスクトップ アプリのカメラへのアクセス がブロックされます。Windows Hello はブロックされません。

オン

このデバイスのカメラへのアクセスはオンになっています



アプリがカメラにアクセスできるようにする

アクセスを許可する場合は、このページの設定を使用して、カメラにアクセスできるア プリを選択できます。アクセスを拒否すると、アプリからカメラへのアクセスがブロックさ れます。Windows Hello はブロックされません。



このページの設定をオフにしても、一部のデスクトップ アプリがカメラにアクセスできるこ

設定

ふ カメラ

デスクトップ アプリがカメラにアクセスできるようにする

意図した機能を果たすためにカメラにアクセスすることが必要なアプリや Windows 機能があります。ここでこの設定をオフにすると、デスクトップアプリと Windows でで きる処理が制限される可能性があります。



- 一部のデスクトップ アプリは、次の一覧に表示されないか、この設定の影響を受け ません。理由を確認する



Microsoft PowerPoint 最終アクセス日時 2020/05/03 21:23:10



Microsoft Teams 最終アクセス日時 2020/05/04 16:25:53





myGoodSketch_debug.exe 最終アクセス日時 2020/07/24 10:48:02

試しにデバイス番号を変えてみる



使えるデバイスを調べる



描画サイズをウィンドウサイズに合わせる

```
#include "ofApp.h"
void ofApp::setup(){
   (途中略)
}
void ofApp::update(){
  video.update();
void ofApp::draw(){
  video.draw(0, 0, ofGetWidth(), ofGetHeight());
}
 (以下略)
```

void ofVideoGrabber::draw(float x, float y)

- ofVideoGrabber クラスの内部画像
 データの左上隅がウィンドウの (x, y)の位置になるように描画する
- 内部画像データの画素数がそのまま反映される
- void ofVideoGrabber::draw(float x, float y, float w, float h)
 - 内部画像データの画素数を幅w、
 高さhに変換して描画する

解像度は低いが全画面に表示される



デバイスの設定パネルを呼び出せるようにする

#include "ofApp.h"

(途中略)

```
void ofApp::keyPressed(int key){
    if (key == 's' || key == 'S'){
        video.videoSettings();
    }
```

's' キーか 'S' キーをタイプした
 ら映像入力デバイスの設定パネ
 ルを開く

■ 映像入力デバイスが対応している 場合

(以下略)

制御パネルを呼び出す

ofApp クラスに画素データとテクスチャのメン バ変数を追加する

#pragma once

```
#include "ofMain.h"
```

```
class ofApp : public ofBaseApp{
   ofVideoGrabber video;
   ofPixels color;
   ofTexture texture;
```

public:

```
void setup();
void update();
void draw();
```

```
void keyPressed(int key);
void keyReleased(int key);
void mouseMoved(int x, int y);
(以下略)
```

ofPixels

画像の画素データのクラス

ofTexture

描画する画像を保持するクラス
 テクスチャマッピング用のデータ

ofApp.cpp で画素データ用のメモリを確保する

```
#include "ofApp.h"
void ofApp::setup(){
   (途中略)
  video.setDeviceID(0);
  video.setup(320, 240);
  color = video.getPixels();
void ofApp::update(){
  video.update();
}
 (以下略)
```

画素データの表示に用いる変数 colorのサイズやチャネル数を 入力映像と同じにするために入 力画像 videoの画素データ自体 をコピーする

フレームが更新されたか調べる

```
#include "ofApp.h"
void ofApp::setup(){
   (途中略)
}
void ofApp::update(){
 video.update();
  if (video.isFrameNew()){
  (次ページに続く)
```

video.update();

- 映像入力からフレームを取り込む
- if (video.isFrameNew()){
 - もしフレームが更新されていたら {}内の処理をする
 - ofApp::update()や ofApp::draw()は 画面表示のタイミングで実行され るが映像入力のタイミングは必ず しもそれと一致しない
 - そのため video.update(); しても前のフレームのまま更新されていないことがある

更新されたフレームの画素データをコピーする

- ofPixels &input{ video.getPixels() };
 - 更新されたフレームの画素データを input で参照できるようにする
- for (size_t i = 0; i < input.size(); ++i){</pre>
 - 更新されたフレームの画素データ input の数だけ { } 内を繰り返す
 - color[i] = input[i];
 - 更新されたフレームの画素データ input を表示用画素データ color にコピーする
- texture.loadData(color);
 - コピーした画素データを描画に使う テクスチャデータに転送する

<pre>#include "ofApp.h"</pre>
<pre>// void ofApp::setup(){ (途中略) }</pre>
<pre>// void ofApp::update(){ (途中略) }</pre>
<pre>//void ofApp::draw(){ texture.draw(0, 0, ofGetWidth(), ofGetHeight()); } (以下略)</pre>

ofPixels クラスの画素データは 直接描画できない

ofPixels::draw() などというメソッ
 ドはない

特に変わらない

明るさを反転する

```
#include "ofApp.h"
//--
void ofApp::setup(){
   (途中略)
}
void ofApp::update(){
  video.update();
  if (video.isFrameNew()){
    ofPixels &input{ video.getPixels() };
    for (size_t i = 0; i < input.size(); ++i){</pre>
      color[i] = 255 - input[i];
    texture.loadData(color);
}
 (以下略)
```


加工方法の切り替え

キー操作で処理を切り替えられるようにする

ofApp クラスにタイプしたキーを保持するメン バ変数を追加する

#pragma once

#include "ofMain.h"

```
class ofApp : public ofBaseApp{
   ofVideoGrabber video;
   ofPixels color;
   ofTexture texture;
   int select;
```

public:

```
void setup();
void update();
void draw();
```

```
void keyPressed(int key);
void keyReleased(int key);
void mouseMoved(int x, int y);
(以下略)
```

ofApp クラスに int 型の select というメンバ変数を追加する
 これにキー操作の結果を入れる

of App.cpp でキー操作の初期設定を行う

```
#include "ofApp.h"
```

```
void ofApp::setup(){
  (途中略)
  video.setDeviceID(0);
  video.setup(320, 240);
  color = video.getPixels();
  select = '0';
```

```
(以下略)
```

select の初期値は文字定数の '0'
 にしておく

 実はこの後(次の次のページ)で select が '0' の時に color = input; と いう代入をして color に input のコ ピーを用意するので、ここでメモ リを確保する必要はなくなる

タイプしたキーを select に代入する

```
#include "ofApp.h"
void ofApp::draw(){
  texture.draw(0, 0, ofGetWidth(), ofGetHeight());
}
void ofApp::keyPressed(int key){
  if (key == 's' || key == 'S'){
    video.videoSettings();
  else{
    select = key;
```

key を select に代入する 's', 'S' 以外の文字コードが入る
select の内容によって異なり処理をする

```
void ofApp::update(){
  video.update();
  if (video.isFrameNew()){
    ofPixels &input{ video.getPixels() };
    switch (select){
    case '0':
     color = input;
     break;
   case '1':
      for (size t i = 0; i < input.size(); ++i){</pre>
        color[i] = 255 - input[i];
      break;
    default:
      break;
    texture.loadData(color);
}
```

- switch (select){
 - select の内容に従って { } 内の case ラ ベルに分岐する
- case '0':
 - select の内容が '0' のとき、これ以降 が実行される
- color = input;
 - input の全画素データを color にコピー する
- break;
 - switchの{}内の処理から抜け出る
- default:
 - select がどの case ラベルとも一致しないとき、これ以降が実行される

キーのタイプで結果が変わる

'0' のタイプ以降



'1' のタイプ以降



課題5-1 2值化

画像の各チャネルを二値化して表示する

'2' キーをタイプしたら R, G, B
 のそれぞれのチャネルを 2 値化
 して表示するようにしなさい

■ 2 値化

 入力データ x を閾値(いきち) t と比較して、x < t なら値を最小 値(ここでは 0)に、それ以外な ら最大値(ここでは 255)にする

ここでは閾値 t を 128 とする









作成したプログラムの実行結果のスクリーンショットを撮って 5-1.png というファイル名で保存し、Moodle の第5回課題にアップロードしてください



課題5-2 ポスター化

画像の各チャネルを4階調化して表示する

- '3' キーをタイプしたら映像を4 階調化(4階調のポスター化) して表示するようにしなさい
 画素値xは0~255の値を持つ
- x を 64 で割り小数点以下を切り 捨てれば 0~3 の値になる
 - 整数の除算をすれば小数点以下は 切り捨てられる
- これに 85 を掛ければ 0, 85, 170, 255 のいずれかの値が得られる









作成したプログラムの実行結果のスクリーンショットを撮って 5-2.png というファイル名で保存し、Moodle の第5回課題にアップロードしてください



チャネルの入れ替え RとBを入れ替えて RGBを BGR にする

'4' キーで赤と青を入れ替えて表示する

```
void ofApp::update(){
 video.update();
 if (video.isFrameNew()){
    ofPixels &input{ video.getPixels() };
    switch (select){
     (涂中略)
   case '4':
     for (size t i = 0; i < input.size(); i += 3){</pre>
        int r{ input[i] };
       int g{ input[i + 1] };
        int b{ input[i + 2] };
       color[i] = b;
        color[i + 1] = g;
        color[i + 2] = r;
      break;
    default:
      break;
     (以下略)
```

- ofPixels &input{ video.getPixels() };
 - inputは1画素の各チャネルが要素 ごとに入っている
 - RGB (アルファチャネルなし)
 - input[0] \rightarrow R, input[1] \rightarrow G, input[2] \rightarrow B
 - input[3] \rightarrow R, input[4] \rightarrow G, input[5] \rightarrow B
 - 従って要素の番号 i = 0 から始めて i < input.size()の間以下を繰り返す
 - r に input[i], g に input[i + 1], b に input[i + 2] を代入する
 - color[i] に b, color[i + 1] に g, color[i + 2]
 に r を代入する
 - iを3増やす(i+=3)





課題5-3



画像をグレースケール化して表示する

・
5'キーをタイプしたら映像をグレイスケール化(白黒画像化)して表示するようにしなさい

■ グレイスケール化

- Y = (R + G + B)/3
 RGB 平均、または
- Y = 0.299R + 0.587G + 0.114B
 - ITU-R Rec BT.601
 - 人間の目はGに対して感度が高くBに 対して低いという特性があるためRGB 平均だとGの影響が低めに出る

■ 他にも多くの変換式がある

- ofPixels &input{ video.getPixels() };
 - inputは1画素の各チャネルが要素 ごとに入っている
 - RGB (アルファチャネルなし)
 - input[0] \rightarrow R, input[1] \rightarrow G, input[2] \rightarrow B
 - input[3] \rightarrow R, input[4] \rightarrow G, input[5] \rightarrow B
 - 従って要素の番号 i = 0 から始めて i < input.size()の間以下を繰り返す
 - input[i], input[i + 1], input[i + 2] をグレー スケール化して y を求める
 - color[i], color[i + 1], color[i + 2] にいずれ も y を代入する
 - iを3増やす(i+=3)







作成したプログラムの実行結果のスクリーンショットを撮って 5-3.png というファイル名で保存し、Moodle の第5回課題にアップロードしてください



背景差分法

背景を切り抜く





画像は2次元の(画素の)データの配列⇒対応する画素同士の引き算が可能 同じ画像同士を引き算すると、当然全部0すなわち**黒**



差異部分の抽出

したがって二つの画像に異なる部分があれば、その部分だけが0でなくなる 逆に0に近い部分は内容が似ているので、その部分が背景だと判断できる

ofApp.h で ofApp クラスに背景を保存するメン バ変数を追加する

#pragma once

#include "ofMain.h"

```
class ofApp : public ofBaseApp{
   ofVideoGrabber video;
   ofPixels color, saved;
   ofTexture texture;
   int select;
```

public:

```
void setup();
void update();
void draw();
```

```
void keyPressed(int key);
void keyReleased(int key);
void mouseMoved(int x, int y);
(以下略)
```

ofApp クラスのメンバ変数に
 ofPixels クラスの saved というインスタンスを追加する
 saved に背景の画像を保存する

ofApp.cpp で保存用のメモリを確保する

```
#include "ofApp.h"
void ofApp::setup(){
   (途中略)
  video.setDeviceID(0);
  video.setup(320, 240);
  saved = color = video.getPixels();
  select = '0';
}
void ofApp::update(){
  video.update();
}
 (以下略)
```

画素データの保存に用いる変数 savedのサイズとチャネル数も 入力映像と同じにする ofApp.cpp で '6' キーをタイプした時にフレーム を保存する

```
void ofApp::update(){
  video.update();
  if (video.isFrameNew()){
    ofPixels &input{ video.getPixels() };
    switch (select){
    case '0':
      color = input;
      break;
    case '1':
      for (size t i = 0; i < color.size(); ++i){</pre>
        color[i] = 255 - input[i];
      break;
     (涂中略)
    case '6':
      saved = input;
      select = '7';
      break;
    default:
     (以下略)
```

select が '6' なら現在のフレームの画素データ input を saved にコピーして取っておく
 select に '7' を代入しておく

■ この後で select が '7' の時に saved の内容を表示するようにする





- select が '7' なら保存したフレーム saved を表示するフレーム color にコピーする
- この処理を case '6' の break の前 に置く
 - したがって case '6' と次の case '7' 間には break がない
 - そのため select が '6' の時は case
 '6' と case '7' の両方の処理が実行 され、select が '7' の時は case '7' 以降の処理だけが実行される

'6' キーをタイプして画像を保存していれば

'0' で現在の映像が表示される



'7' で保存した画像が表示される



課題 5 - 4

入力映像と背景画像の差の絶対値

入力映像と背景画像の差の絶対値を表示する

・ 省、キーをタイプしたら入力映像と背景画像の差の絶対値を 表示するようにしなさい

- input[i] と saved[i] の差の絶対値を color[i] に代入する
- ■絶対値を求める標準ライブラリ関数 abs()
 - z = abs(x y);
 - zはxとyの差の絶対値
 - https://cpprefjp.github.io/reference/cmath/abs.html

カメラの露出・利得を固定する

これは一例

- この方法はカメラの露出や利得 (ゲイン)を固定しておかない と背景がきれいに黒くならない
- 's' または 'S' キーをタイプして 制御パネルで調整する
 - カメラが制御パネルの呼び出しに 対応していない場合はあきらめて ください



結果の例(入力映像と背景画像の差の絶対値)





作成したプログラムの実行結果のスクリーンショットを撮って 5-4.png というファイル名で保存し、Moodle の第5回課題にアップロードしてください



課題 5 - 5

入力映像と背景画像の差の絶対値が閾値以下の画素を黒にする

差の絶対値が閾値以下の画素を黒で表示する

- '9' キーをタイプしたら入力映像 と背景画像の差の絶対値が閾値 以下の画素を黒で表示するよう にしなさい
- 画素値の差の絶対値との比較
 - 入力映像と背景画像の R, G, B の チャネルごとの差の絶対値のそれ ぞれと閾値を比較する
 - 全てのチャネルの差の絶対値が閾 値を超えていれば入力映像の画素 を表示し、そうでなければ黒(R = G = B = 0)を表示する

- ofPixels &input{ video.getPixels() };
- 要素の番号 i = 0 から始めて i < input.size()の間以下を繰り返す
 - input[i]とsaved[i]、input[i + 1]と saved[i + 1]、input[i + 2]とsaved[i + 2]の差の絶対値をそれぞれ求める
 - これらの値のそれぞれと閾値を比 較し、すべて閾値を超えていたら color[i], color[i + 1], color[i + 2] に input[i], input[i + 1], input[i + 2]、そ うでなければ0を代入する
 - i を 3 増やす (i += 3)

結果の例(差の絶対値が閾値以下の画素を黒)





作成したプログラムの実行結果のスクリーンショットを撮って 5-5.png というファイル名で保存し、Moodle の第5回課題にアップロードしてください



課題 5 - 6

入力映像と背景画像の差の絶対値が閾値以下の領域に別の画像を表示する

ofApp クラスに別の画像を保持するメンバ変数 を追加する

#pragma once

```
#include "ofMain.h"
```

```
class ofApp : public ofBaseApp{
   ofVideoGrabber video;
   ofPixels color, saved;
   ofTexture texture;
   ofImage image;
   int select;
```

public:

```
void setup();
void update();
void draw();
```

```
void keyPressed(int key);
void keyReleased(int key);
void mouseMoved(int x, int y);
(以下略)
```

ofApp クラスに ofImage クラスの image というメンバ変数を追加する
ofApp.cppの setup() で image に画像を読み込む

```
#include "ofApp.h"
```

```
//----
void ofApp::setup(){
   (途中略)
   video.setDeviceID(0);
```

```
video.setDeviceID(0);
video.setup(320, 240);
saved = color = video.getPixels();
select = '0';
image.load("image.jpg");
image.resize(video.getWidth(),
video.getHeight());
```

(1) 1

(以下略)

- image.load("image.jpg");
 - プロジェクトのフォルダの bin の data の中にある image.jpg という画 像ファイルを image に読み込む
- "image.jpg" は画像ファイル名
 JPEG, PNG, GIF 画像が読み込める
- image.resize(video.getWidth(), video.getHeight());
 - 読み込んだ画像のサイズを入力映像のサイズに合わせる

画像は bin フォルダの中の data に配置する

🔥 💆 📴 🖛 🗸 bin	管理	_	
ファイル ホーム 共有 表示	ピクチャ ツール		× (?
← → ∽ ↑ 📙 « myGoodSketch	ı → bin →	✓ ບ້	検索
	更新日時	種類	サイズ
📙 data	2020/07/25 22:48	ファイル フォルダー	
🚳 fmodex.dll	2020/06/27 22:10	アプリケーション拡張	1,237 KB
🚳 fmodexL.dll	2020/06/27 22:10	アプリケーション拡張	1,282 KB
🚳 Freelmage.dll	2020/06/27 22:10	アプリケーション拡張	5,878 KB
💴 myGoodSketch_debug.exe	2020/07/25 23:55	アプリケーション	9,669 KB
🕼 myGoodSketch_debug.exp	2020/07/24 10:47	Exports Library File	51 KB
💭 myGoodSketch_debug.ilk	2020/07/25 23:55	Incremental Linker	23,651 KB
🔝 myGoodSketch_debug.lib	2020/07/24 10:47	Object File Library	89 KB
周 myGoodSketch_debug.pdb	2020/07/25 23:55	Program Debug D	30,940 KB

data -管理 × 表示 ピクチャ ツール 共有 ✓ myGoodSketch → bin → data .gitkeep image.jpg

17 E



入力映像と背景画像の差の絶対値が閾値以下の 領域に別の画像を表示する

- 入力映像と背景画像の差の絶対 値が閾値以下の画素を黒にする 代わりに別の画像の同じ位置の 画素の色を表示する
- ofImageの画像から画素データ を取り出すにはgetPixels()メ ソッドを使う
 - ofPixels &back{ image.getPixels() };
 - back は入力映像と同じサイズにした画像 image の画素データを参照する





作成したプログラムの実行結果のスクリーンショットを撮って 5-6.png というファイル名で保存し、Moodle の第5回課題にアップロードしてください



テクスチャマッピング

入力映像を3Dオブジェクトに貼り付ける

ofApp クラスに箱とライト、カメラのメンバ変 数を追加する

#pragma once

#include "ofMain.h"

```
class ofApp : public ofBaseApp{
   ofVideoGrabber video;
   ofPixels color, saved;
   ofTexture texture;
   ofImage image;
   ofBoxPrimitive box;
   ofLight light;
   ofEasyCam camera;
   int select;
```

public:

```
void setup();
void update();
void draw();
```

(以下略)

- ofBoxPrimitive は箱のクラス
- ofLight はライトのクラス
- ofEasyCam はマウスで制御できるカメラのクラス

カメラとライトの設定を行う

```
#include "ofApp.h"
```

- camera.setPosition(0.0f, 0.0f, 200.0f);
 カメラ camera の位置を設定する
- light.setPosition(60.0f, 80.0f, 100.0f);
 - ライト light の位置を設定する

light.enable();
 ライト light を有効にする

表示する画像を箱の表面のサイズに合わせる

- box.mapTexCoordsFromTexture(te
 xture);
 - box の表面に texture をぴったり貼り付ける設定を行う

表示している画像に重ねて箱を描く

void ofApp::draw(){
 ofDisableLighting();
 texture.draw(0, 0, ofGetWidth(), ofGetHeight());
 ofEnableLighting();
 camera.begin();
 ofEnableDepthTest();
 texture.bind();
 box.draw();
 texture.unbind();
 ofDisableDepthTest();
 camera.end();

- ofEnableLighting();~
 ofDisableLighting();
 - この間の 3D CG の描画で陰影付け が有効になる
- ofEnableDepthTest();~
 ofDisableDepthTest();
 - この間の 3D CG の描画で隠面消去 処理を有効にする
- texture.bind(); texture.unbind();
 - この間に描画する 3D CG の図形で texture のマッピング(貼り付け) を有効にする

映像が箱の表面にマッピングされる



課題 5 - 7

箱の代わりに映像を球にマッピングする

球にマッピングした結果



箱の代わりに映像を球にマッピングしなさい

ofBoxPrimitive を ofSpherePrimitive に替えるだけでできる
 球のサイズが小さいので setRadius() メソッドで設定してください
 光源の位置は変更したほうがいいかもしれません

 余裕があれば ofConePrimitive, ofCylinderPrimitive, ofIcoSpherePrimitive, ofPlanePrimitive でも試してみてください

課題のアップロード

作成したプログラムの実行結果のスクリーンショットを撮って 5-7.png というファイル名で保存し、Moodle の第5回課題にアップロードしてください

ソースプログラム ofApp.h と ofApp.cpp を Moodle の第5回
 課題にアップロードしてください



時間の余った人向け課題

第3回の「階層構造」や「課題3-4」の球の部分に映像を貼り付けてみ てください



グレースケール化について

グレースケール化の方法による違い(1)

RGB 平均

ITU-R Rec BT.601



グレースケール化の方法による違い(2)

RGB 平均

ITU-R Rec BT.601

