

# 並列プログラミング Parallel Programming

2022 2Q

演習 第7回

情報理工学院 情報工学系

# 本日の流れ

- 課題内容の説明
- 演習に取り組む

## ● 目的

- OpenCL で画像処理
- 並列計算総復習

## ● 題材

- 画像加工フィルタプログラム
- サーバクライアント対戦ゲーム

# 課題のダウンロード

- 以下からダウンロードしてください

学内アクセス

[\*\*www.img.cs.titech.ac.jp/lecture/para/\*\*](http://www.img.cs.titech.ac.jp/lecture/para/)

# 準備 (1)

- ダウンロードした **para7.zip** を展開する

```
unzip para7.zip
```

- 解凍後のディレクトリ

home directory

parawork(演習用作業トップディレクトリ)

javafx-sdk-18.0.1

Linux Windows

visualvm\_213

Para6

Para7

Makefile (後述)

README 説明文

resource

OpenCLのカーネルプログラムが入っているディレクトリ

bin

クラスファイル (\*.class) が格納される (解凍直後は空)

javadoc

ドキュメントが格納される (解凍直後は空)

src

ソースファイル

lib

実行に必要なjavaのライブラリ集(jarファイル)

## 準備 (2)

- Para7 以下の `._*` ファイルを再帰的に消去

```
cd Para7  
make cleanall
```

MacOS では、`._` で始まるファイルが作成されることがありますが、コンパイル時の障害になるので消去

## 準備 (3)

- ソースファイルを **javac** コマンドでコンパイルしてクラスファイルを作る

今回の演習ではトップディレクトリで

Main15.javaが他のクラスに依存する場合、順次コンパイルしてくれる

```
javac -d bin -encoding UTF-8 --module-path ../javafx-sdk-18.0.1/lib
--add-modules javafx.controls,javafx.swing -sourcepath src -classpath
lib/*: src/para/Main15.java
```

として下さい

実際は一行で書く

-d bin	コンパイル後のクラスファイルをディレクトリ bin に置く
-encoding UTF-8	ソースファイルの文字コードが UTF-8 であることを示す
-sourcepath src	ソースファイルがディレクトリ src 以下にあることをコンパイラに教える
--module-path	モジュールファイルがあるディレクトリをコンパイラに教える
--add-modules javafx.control	依存するモジュールを列挙する
-classpath	依存するクラスやライブラリのありかをコンパイラに教える

※ <https://docs.oracle.com/javase/jp/17/docs/specs/man/javac.html>  
でその他のオプションを確認すること

## 準備 (4)

### ● java コマンドでクラスファイルを実行する

- 今回の演習ではトップディレクトリにて

```
java --module-path /opt/Java/JavaFX/javafx-sdk-18.0.1/lib --add-modules  
javafx.controls,javafx.swing -cp bin:lib/*: para.Main15
```

パッケージ名 起点となるクラスの名前

として下さい（デモ用プログラムは para.Main?? と  
para.Game04 と para.GameServer01 があります）

--module-path javacのオプションと同じ役割

--add-modules javacのオプションと同じ役割

-cp bin:lib/\*:

-cp は -classpathの短縮形

実行に必要なコンパイル済みクラスファイルがディレクトリbin 以下に置かれていること、標準以外のjavaライブラリファイル(jarファイル)がlib/に置かれていることを java コマンドに教える

※ <https://docs.oracle.com/javase/jp/17/docs/specs/man/java.html>  
でその他のオプションを確認すること



## 準備 (5)

- javadoc コマンドでソースファイルのコメント文からHTMLのドキュメントファイルをつくる

```
package para.calc;

import javafx.application.Application;

...

/** JavaFXで作成する電卓プログラム. */
public class Calculator extends Application
{
    /** 入力文字列表示領域. */
    Label input;
```

外部の Javadoc  
文書へのリンク

実際は一行で  
書く

今回の演習では、Para7 直下で、

```
javadoc -sourcepath src -charset utf-8 -encoding UTF-8
        -link https://docs.oracle.com/javase/jp/17/docs/api
        -d javadoc -package para.game para para.graphic.shape
para.graphic.target para.graphic.parser
para.graphic.camera para.graphic.opengl
```

HTML文書の出力先  
ディレクトリ

パッケージ名

としてください

※ <https://docs.oracle.com/javase/jp/17/docs/specs/man/javadoc.html>  
でその他のオプションを確認すること

## 準備 (6)

### ● コマンドをいちいちタイプするのが面倒 ...

今回は Makefile を用意したので make コマンドで  
javac , javadoc の実行が簡単に行える

make Main15	Main15 から
make Main24	Main24 をコンパイルして、実行
make GameServer01	GameServer01 をコンパイルして、実行
make Game04	Game04 をコンパイルして、実行
PARAMETER=servname	servnameはGameServer01が実行されている サーバのアドレス
make clean	bin 以下のクラスファイルをすべて削除
make doc	javadoc コマンドを実行

ソースコードを更新してもmakeが感知しない場合があるので、コードを書き換えても結果に変化がない場合は、一度 make clean して再コンパイルして下さい

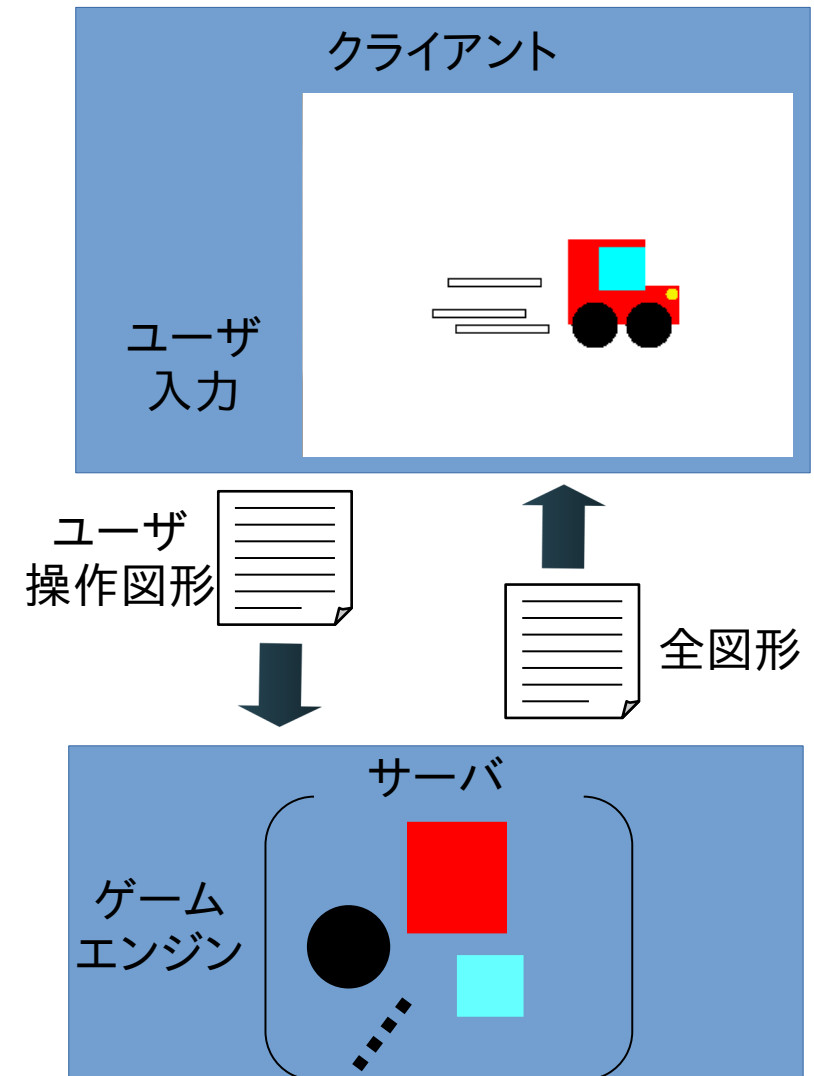
上を実行すると実際に発行されたコマンドが表示される  
Makefile を自分好みに変更してよいです

Makefileの記述ではタブ\tは意味があります。スペースで置き換えると、makeが正しく解釈できません。  
Makefileの書き方は各自調べて下さい

# プログラムの説明 (1) 概要

UMLの詳細は 4Q開講の  
「オブジェクト指向設計」で

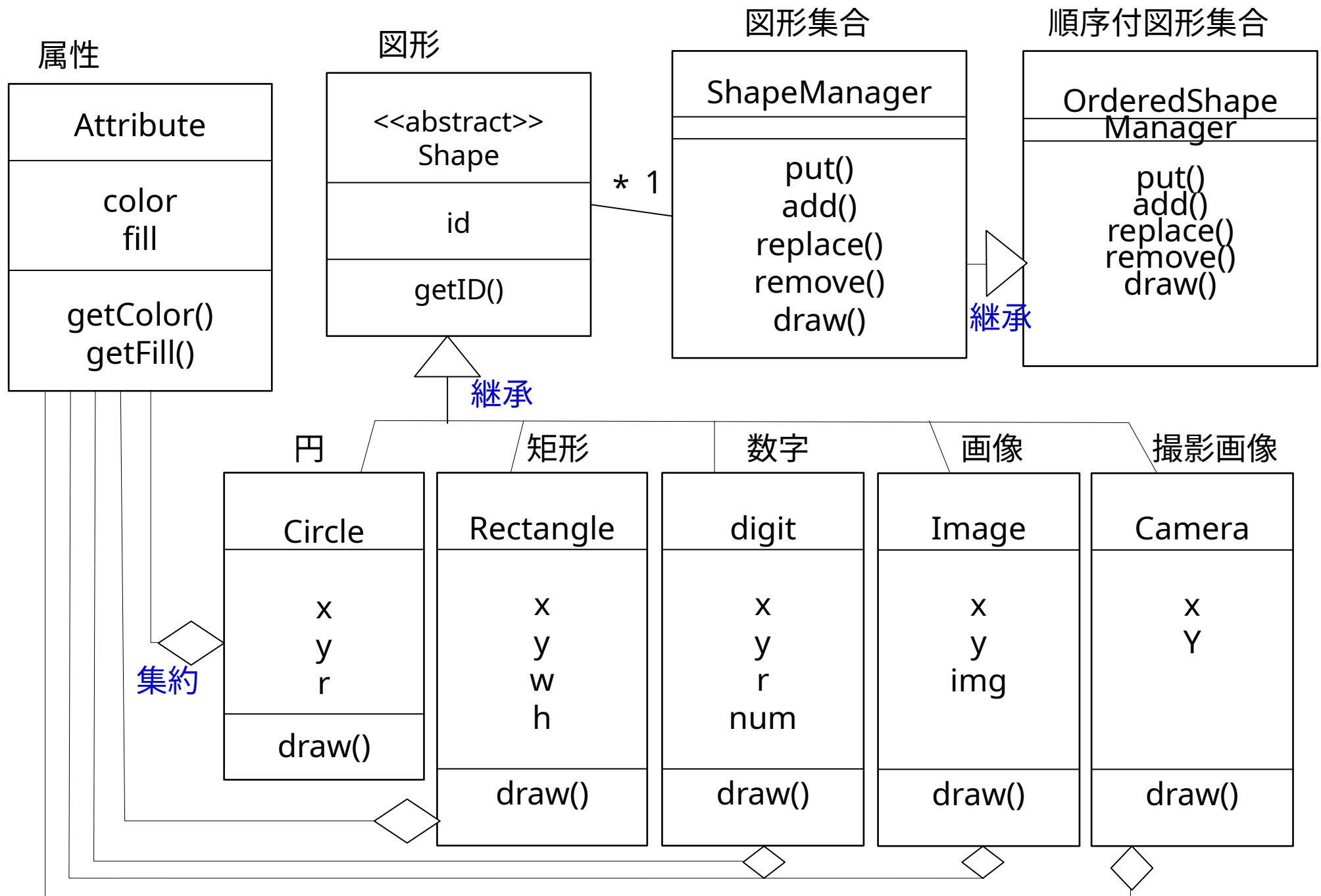
- プログラムは7つから成る
  - メインプログラム (para パッケージ)
  - 図形 (para.graphic.shape パッケージ)
  - 出力装置 (para.graphic.target パッケージ)
  - 構文解析器 (para.graphic.parse パッケージ)
  - 撮影装置 (para.graphic.camera パッケージ)
  - 並列処理 (para.graphic.opengl パッケージ)
  - 追加 ゲーム機能用 (para.game パッケージ)
    - 構文解析器拡張、テキスト出力装置拡張、
    - サーバフレーム、クライアントフレーム



- 基本的流れ
  - メインプログラム内で図形オブジェクトを生成し、出力装置に対して出力する
  - 描画命令が書かれた文字列から構文解析器で図形オブジェクトを生成することもある
  - 描画命令が書かれたファイルもしくは通信ストリームから図形オブジェクトを生成することもある

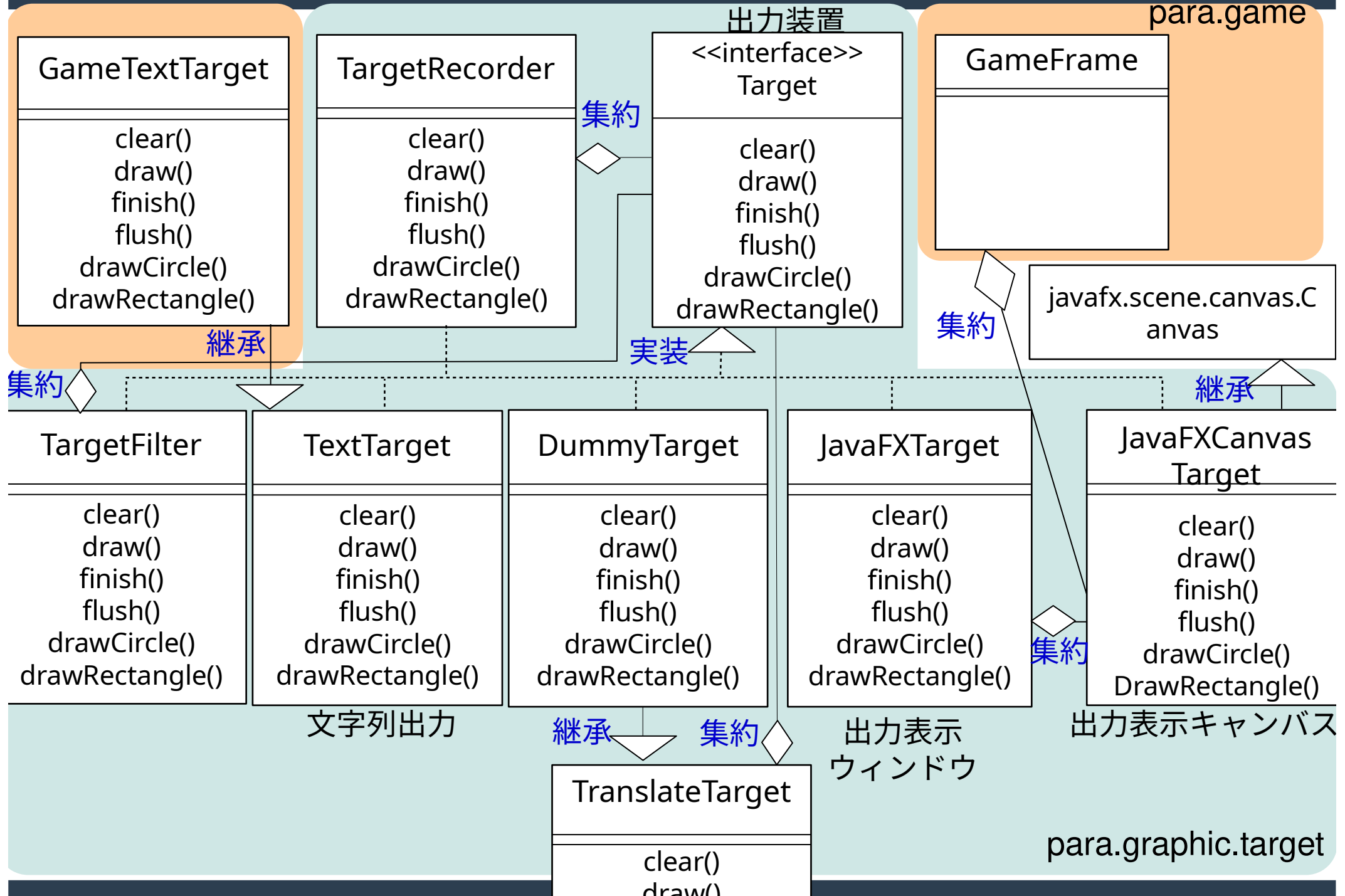
# プログラムの説明 (2) 図形

UMLの詳細は 4Q開講の  
「オブジェクト指向設計」で



# プログラムの説明（3）出力装置

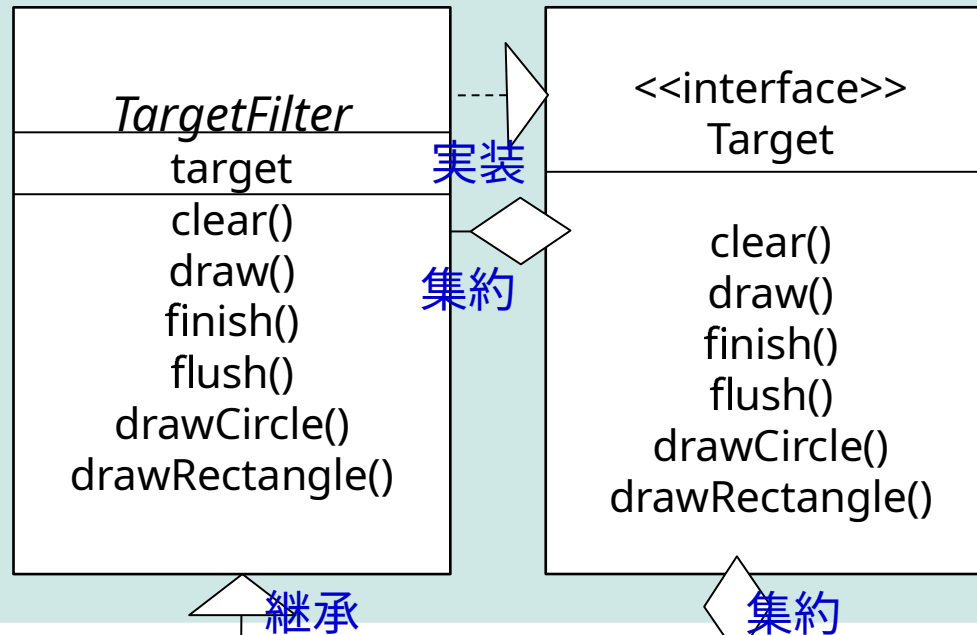
UMLの詳細は 4Q開講の  
「オブジェクト指向設計」で



# プログラムの説明（４）並列処理

para.graphic.target

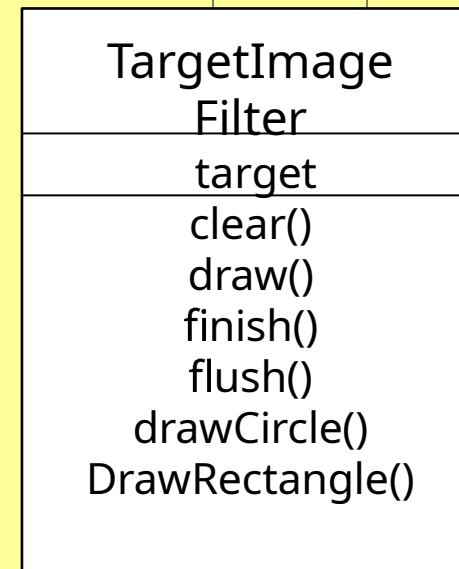
カメラ像の加工



para.graphic.opencil



画像でカメラ像を被う修飾



カメラ像を具体的に加工

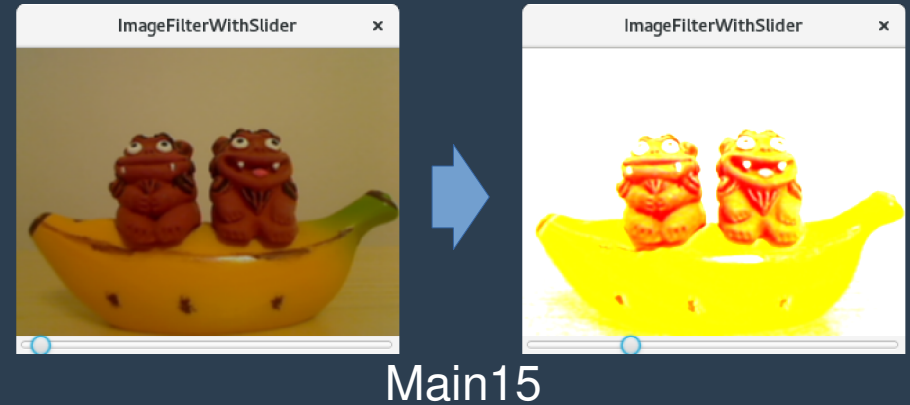
# 課題 0

- javadoc コマンドを実行して HTML 文書を生成し，ブラウザで閲覧する
  - Mac OS X では `open HTML ファイル名` とすればブラウザが起動する
  - 各クラスのパッケージ名などを確認する
  - ブラウザのエンコーディングの設定は UTF-8 にする
  - コンパイルエラーが起こる場合は展開直後にトップディレクトリで一度 `make doc` とタイプすると解決する場合がある

MacOSではダウンロード直後のファイルは安全を疑い、`._`ファイルを作りアクセス制限をOSがする。その`._`ファイルを消す作業が`make doc`には含まれている

# 課題 1

OpenCLを用いてカメラ画像を加工する  
para.Main15を書き換え、imagefilter.clに定義されている様々なカーネルを用いて画像加工を行うプログラムをそれぞれpara.Main16からpara.Main23として作成せよ



1.1) Main15はimagefilter.cl内のカーネルプログラムFilter1を呼び出しているが、代わりにカーネルプログラムFilter2から9を呼び出すようにしたプログラムMain16からMain23を作成せよ。ただし、スライダの値をカーネルプログラムの最後の引数として与えることで、ワークアイテムに送ること。

さらに講義中に示したFilter2の一部コメント部分を有効にしてみるとどのような効果が得られるかを確認せよ。なお、para.Main16からpara.Main23はpara.Main15のクラスを利用して10行以下の行数で実装すること。

**hint** 10行も必要ありません。

1.2) コメント部分を有効にしたFilter2および、Filter6から9についてアルゴリズムを意味を踏まえて説明せよ。

説明には、赤、緑、青、輝度、透明度、半径、微分、二階微分、平均、輪郭、重み付き平均、ガウス関数、ソート、部分画像、量子化の語を適宜用いること。

上記の語の意味が分からない場合は調べること。



## 課題 2

para.Main15ではimagefilter.clを使用してカメラ映像を加工しているが、輪郭を抽出するresource/para/edge.clを指示に従い定義して、それを代わりに用いるpara.Main24を作成せよ



Main24

2.1)カメラ映像を以下で定義される2つの3x3のフィルタを畳み込むことで得られる結果を式(1)で統合して、輪郭抽出した画像を表示するプログラムpara.Main24を作成せよ。実装には、OpenCLを使用し、そのカーネルプログラムの記述されているファイルはresource/para/edge.cl とせよ。

-1	0	1
-2	0	2
-1	0	1

フィルタA

-1	-2	-1
0	0	0
1	2	1

フィルタB

a	b	c
d	e	f
g	h	i

部分画像C

A・CはフィルタAと部分画像Cの対応する要素毎の積の和、B・Cも同様  
sはスライダの値

$$((A \cdot C)^2 + (B \cdot C)^2)s/1600 \quad \text{式(1)}$$

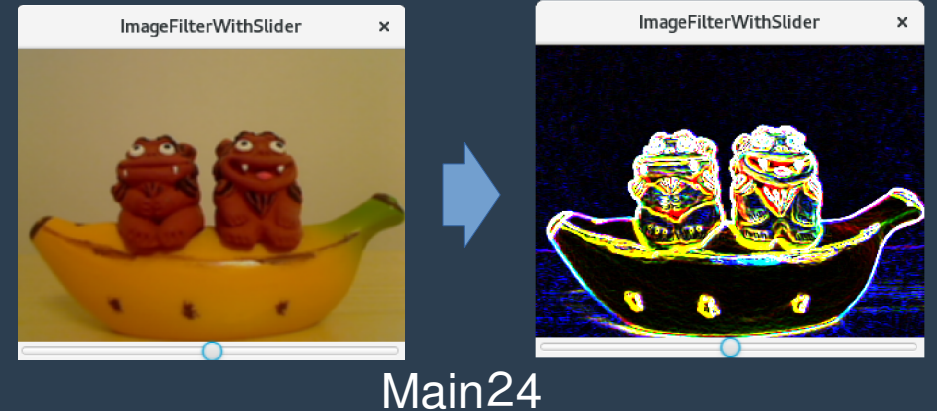
なお、上記の演算はRGBそれぞれにおいて行うこと。またホストプログラムであるpara.Main24の実装はpara.Main15のクラスを利用して10行以下の行数で実装すること

**hint** 10行も必要ありません。

つづく

## 課題 2

para.Main15ではimagefilter.clを使用してカメラ映像を加工しているが、輪郭を抽出するresource/para/edge.clを指示に従い定義して、それを代わりに用いるpara.Main24を作成せよ

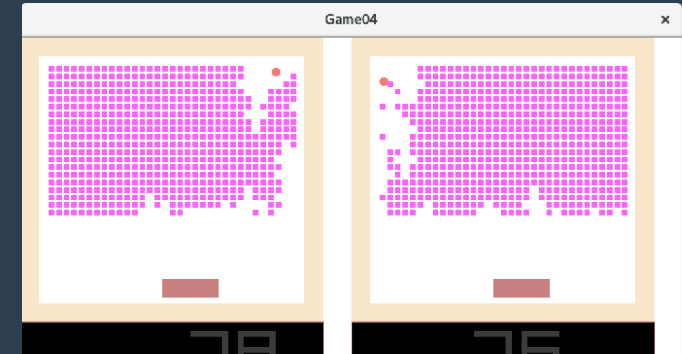


**hint** 実際の実装では、次のようにすると良い。

カメラ画像の注目画素を中心とする3x3の部分画像CとフィルタAとの内積結果の二乗と部分画像CとフィルタBとの内積結果の二乗の和とスライダの値 $s$ (0~255)を1600で割った値の積を部分画像の中心の新たな値とする。注目画像を次々とずらすことで画像全域を処理する。

## 課題 3

para.GameServer01とpara.Game04は対になり、ユーザにゲームを提供するプログラムである。ゲームとして未完成であるので、対戦ゲームを提供できるように改造せよ。



Game04

3.1) 配布されたpara.GameServer01は、socket接続されたクライアント毎に独立してゲームを開始し、それぞれ全く独立したゲームをクライアントに提供している。これを、2つ目のソケット接続が開通した後で、それぞれのゲームが開始されるように変更し、ソースコードの変更箇所を説明せよ。

3.2) ゲームの終了条件を設定し、GameServer01に実装せよ。ゲームの終了時には2つの接続を閉じること。終了条件とその実装方針を説明し、実装部分を記せ。

ただし、3.5)と合わせて回答したい場合には、3.5)の回答を参照と記せ

**hint** GameTextTarget.finish()を呼び出すとStream.close()が呼ばれる。

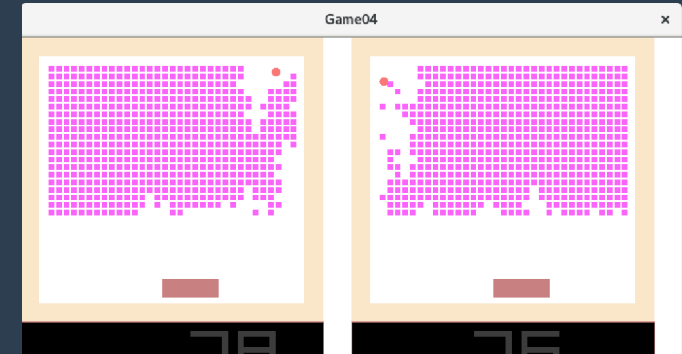
3.3) para.Game04は、ゲーム中にスタートボタンをクリックしても新たなゲームの開始をしないが、ゲーム中でないときは、サーバへの接続を開始してゲームを始める。ゲーム中であるかの判定に用いられている変数  はスレッド  とスレッド  で参照、代入がされるが、スレッドセーフであろうか。

空欄に入る変数名、スレッド名をそれぞれ答えよ。さらに、スレッドセーフであるかを考察し、スレッドセーフであると考えられる場合はその理由を述べ、スレッドセーフでないと考えられる場合は問題が生じる状況を挙げよ。

つづく

## 課題 3

para.GameServer01とpara.Game04は対になり、ユーザにゲームを提供するプログラムである。ゲームとして未完成であるので、対戦ゲームを提供できるように改造せよ。



Game04

3.4) 2接続されたゲームが対戦ゲームとなるようにルールを考え、そのゲームルールを実装しゲーム全体の完成度を高めよ。考えたルールとその実装について説明せよ。ゲーム性が高められるようにルールで工夫した点を説明すること。またゲーム画面の改良など付随して行った改良についても説明すること。

3.5) 対戦ゲームとしてのゲームの終了条件を設定し、その勝敗決定規則とその実装法について説明せよ。

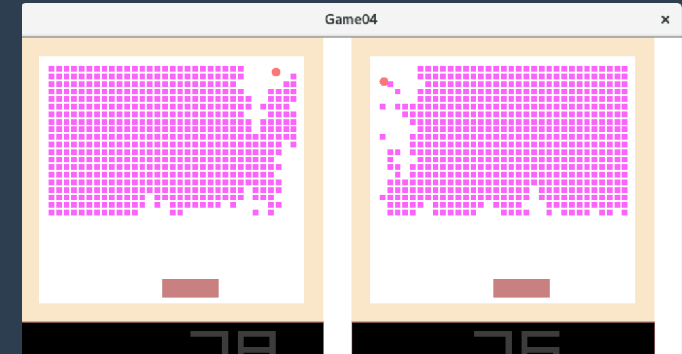
3.6) 対戦ゲーム中はカメラ画像への加工をせず、勝敗が決定した後、勝者のカメラ画像はそのまま表示させつつ、敗者のカメラ画像には敗北感を表現する画像加工をOpenCLで演算して施されるようにせよ。

**hint** 画像加工は敗者のクライアント側で行う。サーバ側からは加工の指示を出す。

つづく

## 課題 3

para.GameServer01とpara.Game04は対になり、ユーザにゲームを提供するプログラムである。ゲームとして未完成であるので、対戦ゲームを提供できるように改造せよ。



Game04

3.7) ゲームルールが分かるような、楽しくゲームをしている対戦(風な)画面をTargetRecorderを利用して録画せよ。録画ファイルはgame2012345.mp4という名前にして回答のzipに入れること。

自分の7桁の学籍番号

対戦可能なネットワーク環境が準備できる人は対戦風景の録画をしましょう。

- プライベートネットワーク内のもう一人と対戦できる場合 (情報工学系VPNを利用する場合)
- グローバルIPでサーバを起動できて通信経路でブロックされない場合
- proxyやrouterの設定を行うことで、プライベートIP間で通信できる場合

対戦相手が居ない場合や相手と通信できない場合は対戦風な録画をしましょう。

**hint** ServerSocketの監視PORTはGameServerFrameクラスの定数で指定されてます

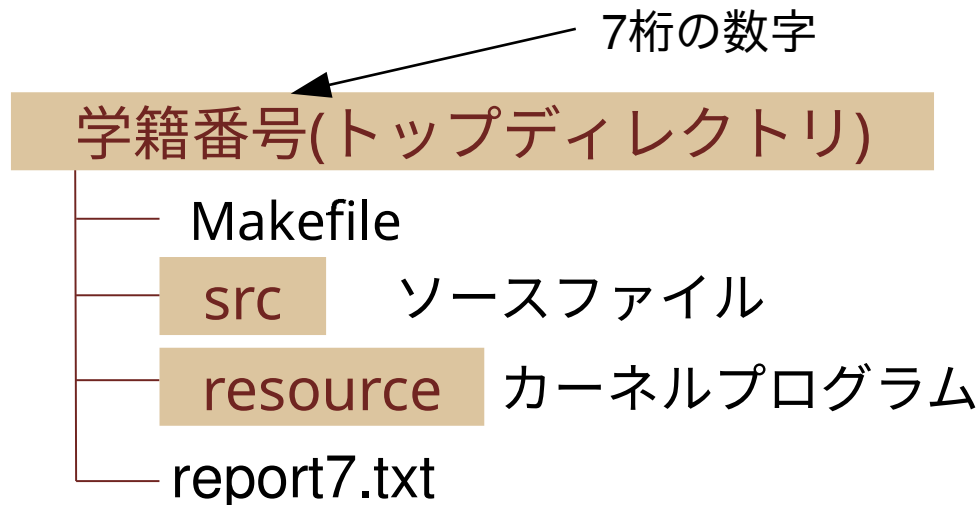
**hint** クライアントのユーザ入力に対するイベント処理の定義はGameFrameクラスにあります

録画で工夫した点、行ったこと(ルールが分かるように録画で工夫したこと、通信環境の準備のこと、対戦風にするために特別なクライアントを作成したこと、コンピュータ対戦できるようにしたなど)を説明せよ。

# 課題 1,2 提出方法 (1 of 3)

締め切りが  
2回に分かれています。

- para7.zip を展開したディレクトリ構造を保ったまま，課題 の変更作業を行う
- 課題 1,2 で自分が変更したファイルの先頭には自分の名前と学籍番号を書いておく
  - プログラムの場合はコメント内に書く
- 課題 1 の回答文および感想を書いた report7.txt を用意する (雛形は課題のウェブページ)



回答プログラムと  
レポートの作成終  
了後、次のようにフ  
ァイルを配置したディ  
レクトリを作成

次ページに具体的な作業手順あり



# 課題 1,2 提出方法 (2 of 3)

締め切りが  
2回に分かれています。

- 提出用ディレクトリを作成する 学籍番号から7桁の数字にすること

```
mkdir dir
```

- ソースファイルとカーネルファイルのディレクトリの<sup>今回はPara7</sup>コピーを作る

```
cp -R トップディレクトリ/src トップディレクトリ/resource dir
```

- dir* に Makefile report7.txt もコピーする

```
cp トップディレクトリ/Makefile トップディレクトリ/report7.txt dir
```

- 次のコマンドを実行する

```
zip ex7-2012345.zip -r dir
```

学籍番号に対応する7桁の数字にすること

*dir* 以下の内容が圧縮され、ex7-2012345.zip が作られます

- 圧縮後に内容を“unzip ex7-2012345.zip”で確認すると提出ミスを防げて安全

- 作成した zip ファイルを T2SCHOLA にアップロードする
- 締め切り
  - 8 月 10 日 (水) 12:00 noon (JST)

注意  
正午です!

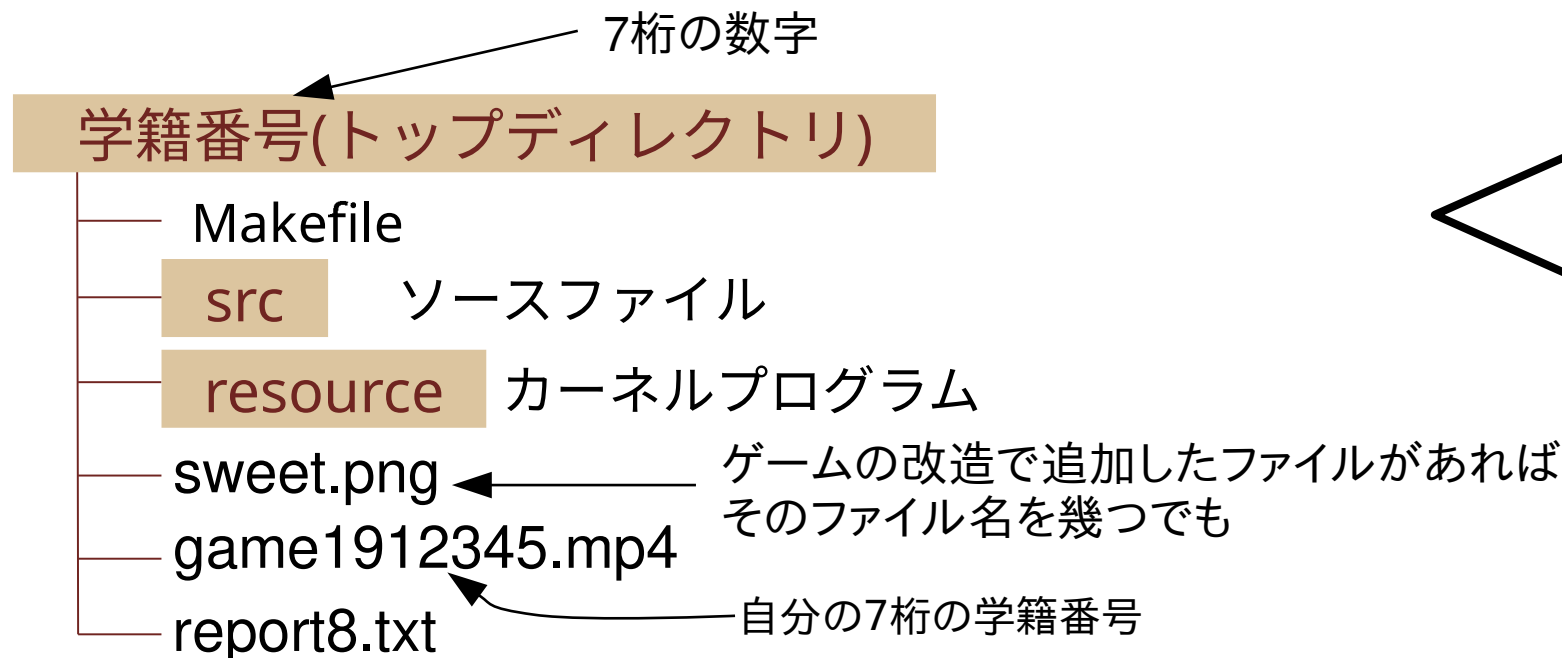
締め切り後程なくしてキャンパス停電のため演習室のシステムは停止作業に入ります。



# 課題 3 提出方法 (1 of 3)

締め切りが  
2回に分かれています。

- para7.zip を展開したディレクトリ構造を保ったまま，課題 の変更作業を行う
- 課題 3 で自分が変更したファイルの先頭には自分の名前と学籍番号を書いておく
  - プログラムの場合はコメント内に書く
- 課題 3 の回答文および感想を書いた report8.txt を用意する (雛形は課題のウェブページ)



回答プログラムとレポートの作成終了後、次のようにファイルを配置したディレクトリを作成

次ページに具体的な作業手順あり

## 課題 3 提出方法 (2 of 3)

締め切りが  
2回に分かれています。

- 提出用ディレクトリを作成する 学籍番号から7桁の数字にすること

```
mkdir dir
```

- ソースファイルとカーネルファイルのディレクトリのコピーを作る

今回はPara7

```
cp -R トップディレクトリ/src トップディレクトリ/resource dir
```

- dir* に Makefile report8.txt もコピーする

```
cp トップディレクトリ/Makefile トップディレクトリ/report8.txt dir
```

- dir* に課題 3 のゲーム改造で新たに必要とするファイルがもしあれば全て所定のディレクトリへコピーする

必要とするファイル名

```
cp トップディレクトリ/sweet.png dir
```

- dir* に game2012345.mp4 もコピーする

```
cp トップディレクトリ/game2012345.mp4 dir
```

- 次のコマンドを実行する

```
zip ex8-2012345.zip -r dir
```

学籍番号に対応する7桁の数字にすること

*dir* 以下の内容が圧縮され、ex8-2012345.zip が作られます

ー 圧縮後に内容を“unzip ex8-2012345.zip”で確認すると提出ミスを防げて安全

- 作成した zip ファイルを T2SCHOLA にアップロードする
- 締め切り
  - 8 月 17 日 (水) 20:00 (JST)



夜8時です!