

コーディング規則

デザインパターンは GoF のやつを参考にして作ってね

↓簡単に説明してくれてるサイト

<http://itpro.nikkeibp.co.jp/article/COLUMN/20051123/225074/>

動作は関数合成で作成する(また、できるように作成する)

命名規則

クラス名は大文字から、アンダーバーはつけない

(例:TitleScene)

メンバ変数名は小文字から、アンダーバーはつけない

(例:colorData)

関数名は小文字から、アンダーバーはつけない

(例:touchEvent)

m_とかみたいに余計なものは絶対につけない

ゲッターやセッターの名前は `get`、`set` から始める

bool flag みたいに、なんの意味があるものなのかわからないやつは作らない。

(最悪でも、updateFlag みたいに、何の為なのかわかるようにする)

ファイルを作成したら、一番上に誰がいつ作ったかをコメントで書く

(.h 等)

↓これコピペして書き換えればいい

/*=====*/

//作成者:鷺津

//作成日:2014/07/10

$$/*=====*/$$

ファイルを編集したら、上の方に誰が何のために編集したかをコメントで書く

↓これコピペして書き換えればいい

/*>>>>>>>>>>>>>>*/

//編集者:鷺津

//編集日:2014/07/10

//編集内容:壁にめり込むバグを修正するため

/*>>>>>>>>>>>>>>*/

また、これらはクラスを作成、編集した場合には、全て.h 内に書くこととする

例

TitleScene.h

$$/*=====*/$$

//作成者:鷺津

//作成日:2014/07/10

$$/*=====*/$$

/>>>>>>>>>>>>>>>*/*

//編集者:鷺津

//編集日:2014/07/10

//編集内容:壁にめり込むバグを修正するため

/*>>>>>>>>>>>>>>>*/

/*>>>>>>>>>>>>>>>>*/

//編集者:鷺津

//編集日:2014/07/11

//編集内容:描画関係の修正

/*>>>>>>>>>>>>>>>>>>*/

```
#pragma once
```

```
#include "Scene.h"
```

```
class TitleScene: public Scene{
```

private:

protected:

public:

```
TitleScene ();
```

```
virtual ~ TitleScene ();
```

```
void initialize();
```

```
void update();
```

```
void render();
```

```
void finalize();
```

 $\};$

```
TitleScene.cpp
#include " TitleScene.h"
TitleScene:: TitleScene ()
:Scene(){
//コンストラクタ
}

TitleScene::~ TitleScene (){
//デストラクタ
}

void Logo::initialize(){
//初期化处理
}

void Logo::update(){
//更新処理
}

void Logo::render(){
//描画処理
}

void Logo::finalize(){
//終了処理
}
```

