

Trickle コーディングルール

～こまめな Ctrl+S は世界を救う～

文責：よこた

目次

- ①ネーミング編
- ②クラス設計編
- ③処理記述編
- ④コメント編

①ネーミング編

1. クラス名、関数名は先頭大文字、あと小文字(単語が続くならキャメル記法)

ex)

```
class GameObject{
    void FunctionHoge(int aaa);
};
```

2. 変数名は小文字始まり、単語が続くなら2つ目以降をキャメル記法

※先頭に_(アンダーバー)はつけないこと(規約で禁止されている)

ex)

```
int timeCount;    //OK
int _timecount;   //NG
int TimeCount;    //NG
```

3. 変数・関数などの名前は見ただけで役割がわかるものを

ex)

```
int cnt;                //何の???
int animCnt;            //OK

void Set(Vec2&);         //何を???
void SetPos(Vec2& pos);  //OK
```

4.マジックナンバーは厳禁、必ず const や enum など で名前をつけるべし

*MapNum を 見 よ 。
switch 文の case に数字が出たら負けだと思え。

5. 英単語は正しく使おう

せめてググれよ。
あるいはツールに頼るのもアリ
ネーミングツール [codic](https://codic.jp/engine)
URL : <https://codic.jp/engine>

※ちなみに「アップデート」は「update」で一単語です
「UpDate(up date)」は「日付アップ」とか訳分からん和訳を返されるので確認すべし

②クラス設計編

1. 一クラス一役割

※でも無闇矢鱈にクラスを増やすのもやめよう

2. 基本的にメンバ変数は private

必要に応じてゲッターセッターを作る

3. コンストラクタを使おう

変数の初期化はコンストラクタで行うこと(ヘッダーには書かない!)

変数の初期化にはなるべく初期化子リストを使うこと

※初期化の順番も注意(処理速度が(多少)変わる)

※というか知つとかなないと恥ずかしい

ex)

```
Object(Vec2& pos, Vec2& size)
:
  pos(pos),
  hitBase(Box2D(pos, size))
{
  InitializeHoge();
}
```

4. ヘッダーでの関数定義にも仮引数名をつけよう

マウスポインターを当てて出てくる説明はヘッダーの方です

『誰が見ても分かる』を徹底しよう

cpp を開かせたら負けだと思え。

ex)

```
void Func(Vec2&, Vec2&, int, bool = true);    //は？？？
void Func(Vec2& pos, Vec2& size, int animCnt, bool active = true); //OK
```

③処理記述編(cpp とか)

1. if 文の条件はなるべく簡潔に

※①条件が長い、②同じ条件の if が他にもある
→関数化！！

2. if 文のネストは浅くしよう

```
if(...){  
    if(...){  
        if(...){  
            //クソ  
        }  
        else{  
            //なんかもうクソ  
        }  
    }  
    else{  
        //もはやどの if と対応してんのかも分からん  
    }  
}
```

↓

```
if(〇〇 && ××){  
    if(checkHoge(...)){  
        //まだマシ  
    }  
}
```

3. 早期 return を活用しよう

```
if(active){  
    //処理  
    //だったら処理  
    //else いつ出てくんの???  
}
```

↓

```
if(!active){ return; }    //false ならここで終了！  
//処理
```

③コメント編

1. コメントを書くべし

2. コメントを書くべし

3. コメントを書くべし(現時点では書きすぎてもいいから書け)

コメントは基本ヘッダーに記述すること

目標は【ヘッダーはコメント9割！ cpp はコメント0割！】※無理はしないように

ただし書き方は **Doxygen** のルールに則ること

参考 URL : <https://qiita.com/inabe49/items/23e615649e8539d857a8>
<http://www.doxygen.jp/docblocks.html>

！ポイントは『マウスカーソルを当てて出てくる説明で使い方が分かること』！

ex)

Object クラスを定義するファイル

```
/**
 * @file Object.h
 * @brief 全オブジェクトの基底クラスを定義する
 * @date 2018.8.9
 * @author 横田麻梨子
 */

#include "Hoge.h"

/**
 * @brief 全オブジェクトの基底クラス
 * @detail 座標、当たり判定はこのクラスで管理する
          コンストラクタで座標とサイズを指定
 */
class Object{
    //! 現在座標
    Vec2 pos;
    //! 当たり判定用矩形
    Box2D hitBase;
```

```
public:
    /**
     * @brief 座標を変更する
     * @param Vec2& pos 変更後の座標
     */
    void SetPos(Vec2& pos);

    /**
     * @brief 当たり判定を行う
     * @param Box2D& obj 判定を行う相手の矩形
     * @return bool 矩形が重なっていれば true
     */
    bool Hit(Box2D& obj);
};
```

＊＊以下、随時更新＊＊