

C E S A 2019 命名規則

・スクリプト

自分の担当のスクリプトは自分の名字の namespace をつける
スクリプト名はアッパーキャメルケースでの記述

```
namespace Goto
{
}
```

・変数名

ローワーキャメルケースでの記述

文字の初めは小文字

文字をつなげる場合は先頭だけを大文字にする

メンバー変数 _をつける _time

ローカル変数 _をつけない time

public エリアと private エリアは分けて書く

変数についてのコメントはその変数の横に書く

```
public int _time;                     // 制限時間
public int _flag;                    // プレイヤーのフラグ
```

```
private Vector3 _playerPos;         // プレイヤーの座標
private Vector3 _playerVel;         // プレイヤーの向き
```

定数

CONSTANT ケースでの記述

すべて大文字

文字をつなげる際は_を使う

```
PLAYER_SPEED
MAX_SPEED
enum PlayerFlag
{
    HIT_TO_ENEMY = (1 << 0),         // 敵とのフラグ
    HIT_TO_GROUND = (1 << 1),        // 地面とのフラグ
    HIT_TO_OBJECT = (1 << 2),        // オブジェクトとのフラグ
};
```

- 関数名

アッパーキャメルケースでの記述

文字の初めは大文字

文字をつなげる場合は先頭だけを大文字にする

関数のコメントはスマートコメント編集の機能を利用する
引数と戻り値の説明も記述する

```
/// <summary>
/// 更新処理
/// </summary>
void FixedUpdate()
{

}

/// <summary>
/// 乱数の取得(int型)
/// </summary>
/// <param name="min">最小値</param>
/// <param name="max">最大値</param>
/// <returns>minからmaxの間の値</returns>
int GetRand(int min, int max)
{
    int random;    /// 習得した乱数を格納する

    srand((unsigned)time(NULL));
    random = rand() % (max - min + 1) + min;

    return random;
}
```

- フラグ

当たり判定などのフラグを使うとき bool 型は使用しない
Git に上がっている Flug クラスを利用する