

他のゲームオブジェクトにアクセスしたい場合

▼ ステータス	読んでください
▼ タグ	プログラミング
🕒 作成日時	@2021年11月12日 午前 11:08

※例なので、同じことはしないでください。

例) プレイヤーの動きに合わせて、背景、壁、ターゲットも動かしたい！

↓

新しく、GO_SS_Movementを作成

欲しいクラスの前方宣言

```
class GO_SS_BackGround;  
class GO_SS_Player;  
class GO_SS_Wall;  
class GO_SS_ShotString;  
class GO_SS_Target;
```

ほしいクラスのポインタを用意

```
private:
    //ゲームシーン
    const int GAME_SCENE = GAMESCENE_GAME_TEST;

    GO_SS_BackGround* m_pBackGround;
    GO_SS_Player* m_pPlayer;
    GO_SS_Wall* m_pWall;
    GO_SS_ShotString* m_pShotString;
    GO_SS_Target* m_pTarget;
```

セッターを作成

```
//セッター
void SetBackGround(GO_SS_BackGround* p_BackGround) { ... }
void SetPlayer(GO_SS_Player* p_Player) { ... }
void SetWall(GO_SS_Wall* pWall) { ... }
void SetShotString(GO_SS_ShotString* p) { ... }
void SetTarget(GO_SS_Target* pTarget) { ... }
```

そしたら、GO_SS_Managerに行きます

いつもどおり、Registreに登録。

多分、一番下の方にこれがあるので、定義へ飛びます。

```
//他のゲームフレームワークに干渉したい場合使用
void SetGameObject();
```

今はこうなってる

```

- /*-----
  :   SetGameObject()
  :-----
- */
- void GO_SS_Manager::SetGameObject()
- {
-     //mp_ssComunication
-     {
-         mp_ssMovement->SetBackGround(mp_ssBackGround);
-         mp_ssMovement->SetPlayer(mp_ssPlayer);
-         mp_ssMovement->SetShotString(mp_ssShotString);
-         mp_ssMovement->SetWall(mp_ssWall);
-         mp_ssMovement->SetTarget(mp_ssTarget);
-
-         mp_ssShotString->SetPlayer(mp_ssPlayer);
-         mp_ssShotString->SetTarget(mp_ssTarget);
-     }
- }
-

```

これで、GO_SS_Movementクラスのポインタを使用することができます

```

mp_ssMovement->SetBackGround(mp_ssBackGround);
mp_ssMovement->SetPlayer(mp_ssPlayer);
mp_ssMovement->SetShotString(mp_ssShotString);
mp_ssMovement->SetWall(mp_ssWall);
mp_ssMovement->SetTarget(mp_ssTarget);

```