interface IPlayer{

int MaxHP{

get;

}

int HP{

get;

set;

}

event System.EventHandler OnMaxHPChanged;

event System.EventHandler OnHPChanged;

}

class Player{

int HP{

get;

set{

value = Math.Clamp( 0, MaxHP )

if( value != m\_hp ){

m\_hp = value

if( OnHPChanged != null )

OnHPChanged( this, System.EventArgs.Empty )

}

}

}

event System.EventHandler OnHPChanged;

}

enum Player{

One = 0,

Two = 1

}

Player.One

Player.Two

class PlayerInfoDisplayer{

IPlayer m\_player;

[Serialized]

int m\_index;

[Serialized]

NumberDisplayer m\_num

void Awake(){

m\_player = BattleManager.Singleton.GetPlayer( m\_index )

// ex

hpText.text = m\_player.HP.ToString()

// event

m\_player.OnHPChanged += OnHPChanged;

}

void OnDestroy(){

m\_player.OnHPChanged -= OnHPChanged;

}

void OnHPChanged( object sender, System.EventArgs e){

m\_num.Value = ((IPlayer)sender).HP.ToString()

}

}

interface INumberDisplayer{

int Value{

get;

set;

}

event System.EventHandler ValueChanged;

}

class abstract NumberDisplayer : MonoBehavior, INumberDisplayer{

}

class SprtieNumberDisplayer : NumberDisplayer{

}

class TextNumberDisplayer : NumberDisplayer{

[SerializeFiled]

UI.Text m\_text;

public int Value{

get;

set{ m\_text.Text = value.ToString(); }

}

}