GeneralPack.unitypack リファレンス

目次

■始めに

スクリプト

■Fade.cs

■LetterFade.cs

■SceneChange.cs

■Score.cs

■OnEffect.cs

■Gauge.cs

■始めに

GeneralPack.unitypackを使用する場合は

シーン内に「GameManager」PrefabをHierarchy内に配置してください。

GameManagerは

・Fade.cs

・SceneChanger.cs

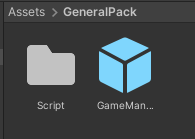
・Score.cs

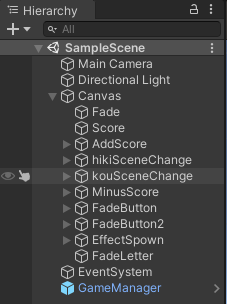
・OnEffect.cs

・Gauge.cs

5種類がアタッチされています

Inspectorの詳しい説明は、各スクリプトの説明を参考にしてください





■Fade.cs

・定数

Int ALPHA アルファ値の定数 　1固定

・構造体

float \_COLOR RGBA値を保存する構造体

・public変数

Bool isFadeIn フェードイン開始フラグ

true:　フェードイン開始

false:　フェードイン中断

Bool isFadeOut フェードアウト開始フラグ

true:　フェードアウト開始

false:　フェードアウト中断

・private変数

　　 GameObject fadeObj フェードさせたい画像オブジェクトを指定

float addAlpha アルファの増加値

float takeAlpha アルファの減少値

float alphaValue 現在のアルファ値

Image fadeImage Imageコンポーネント変数

\_COLOR fadeRGB \_COLOR構造体を格納する変数

・private関数

OnFadeIn() アルファ値の減算後、ImageコンポのColor項目に新たに

RGBA値を代入する関数

引数・返値 なし

OnFadeOut() アルファ値の増算後、ImageコンポのColor項目に新たに

RGBA値を代入する関数

引数・返値 なし

■使い方

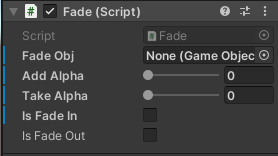
「Image」コンポがアタッチされているオブジェクトを

Inspectorの「FadeObj」にアタッチしてください

その後、「GameManager」オブジェクトから「Fade」コンポを取得し

「isFadeIn」 Or 「isFadeOut」のフラグを切り替えてください

■Inspector



・FadeObj … フェードさせたいオブジェクトを指定します

オブジェクトにはImageコンポーネントがアタッチされている必要があります

Imageコンポがアタッチされていない場合

UnassignedReferenceExceptionエラーが返されます。

・AddAlpha … アルファ値の増加量を設定します。

最小値:0

最大値:1

・TakeAlpha … アルファ値の減少量を設定します。

最小値:0

最大値:1

・IsFadeIn … フェードインのON/OFFを切り替えます。

チェックが入っている: フェードイン開始

チェックが外れている:フェードイン中止

・IsFadeOut … フェードアウトのON/OFFを切り替えます。

チェックが入っている: フェードアウト開始

チェックが外れている:フェードアウト中止

■LetterFade.cs

・定数

Int LALPHA アルファ値の定数 　1固定

・構造体

float \_LCOLOR RGBA値を保存する構造体

・private変数

Bool le\_isFadeIn フェードイン開始フラグ

true:　フェードイン開始

false:　フェードイン中断

Bool le\_isFadeOut フェードアウト開始フラグ

true:　フェードアウト開始

false:　フェードアウト中断

float le\_addAlpha アルファの増加値

float le\_takeAlpha アルファの減少値

float le\_alphaValue 現在のアルファ値

TextMeshPro fadeLetter TextMeshProコンポーネント変数

\_LCOLOR le\_fadeRGB \_LCOLOR構造体を格納する変数

・private関数

LOnFadeIn() アルファ値の減算後、TextMeshProコンポのColor項目に

新たにRGBA値を代入する関数

引数・返値 なし

LOnFadeOut() アルファ値の増算後、TextMeshProコンポのColor項目に

新たにRGBA値を代入する関数

引数・返値 なし

■使い方

TextMeshProコンポを含むオブジェクトに本スクリプトをアタッチしてください

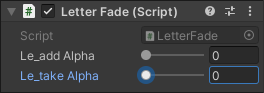
再生時に本スクリプトがアタッチされた文字が

フェードイン/フェードアウトを繰り返します

TextMeshProコンポがアタッチされていないオブジェクトの場合

UnassignedReferenceExceptionエラーが返されます

■Inspector



・Le\_AddAlpha … アルファ値の増加量を設定します。

最小値:0

最大値:1

・Le\_TakeAlpha … アルファ値の減少量を設定します。

最小値:0

最大値:1

■SceneChanger.cs

・public変数

String pSceneName シーン名

・public関数

SceneChange() pSceneNameで指定したシーン名へ移動する

引数・返値 なし

CallSceneChange(string sceneName) 引数で指定したシーン名へ移動する

引数:sceneName シーン名

返値:なし

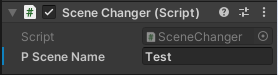
■使い方

「GameManager」オブジェクトから「SceneChanger」コンポを取得し

「SceneChange」 Or 「CallSceneChange」を呼び出してください

指定したシーンへ移動します

■Inspector



・P Scene Name … 移動先のシーン名

■Score.cs

・public変数

Int addScore 数値の加算量

Int nowScore 現在表示しているスコア

・private変数

　　 Int totalScore 総スコア

・private関数

AddScore (int getPoint) 取得したポイントを総スコアに加算する

引数: getPoint 取得ポイント

返値:なし

OnFadeOut() 取得したポイントを総スコアから減算する

引数: getMinusPoint取得ポイント

返値:なし

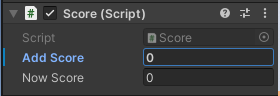
■使い方

「GameManager」オブジェクトから「Score」コンポを取得し

変数「nowScore」を呼び出し、text等に代入してください

スコアが加算/減算されると、カウントアップ/カウントダウンが行われます

■Inspector



・AddScore … カウントの増加量

・NowScore…現在のスコア

■OnEffect.cs

・SerializeField変数

　　 List<GameObject> effectObjList 使用するエフェクトのPrefabオブジェクトリスト

・private変数

Vector3 effectPosition　　　　　　　　　 発生させるエフェクトの座標

List<GameObject> spawnObjList ゲーム上に生成されたエフェクトリスト

・public関数

SpawnEffect(Vector3 effectPo,int index) 登録したエフェクトリストから指定した座標に

エフェクトを発生させます

引数: effectPo 発生座標

index 使用するエフェクト

オブジェクトの要素数

返値:なし

■使い方

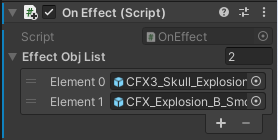
「GameManager」オブジェクトの「EffectObjList」に使用したいPrefabオブジェクトを登録します

スクリプトから「OnEffect」コンポを取得し「SpawnEffect」を呼び出してください

引数には、Vector3型のposition、「EffectObjList」内の発生させたいエフェクトの要素数を

指定してください

■Inspector



・EffectObjList … 使用するPrefabオブジェクトのリスト

Prefabオブジェクトには「ParticleSystem」コンポーネントがアタッチされている

必要があります

アタッチされていないPrefabオブジェクトを登録するとエフェクトとして

判定しません

■Gauge.cs

・定数

float MAXFILL ゲージ量の最大値　1.0固定

float EMPFILL ゲージ量の最小値　0.0固定

・SerializeField変数

　　 GameObject gaugeObj 使用するImageオブジェクト

float totalValue 総ゲージ量

float totalTime 総タイム

・public 変数

bool isAddGauge ゲージ増加開始フラグ

bool isTakeGauge ゲージ減少開始フラグ

・private変数

int isGaugeFlg　　　　　　　　　 ゲージの前回のフラグの状態

0:どちらもfalse

1:isAddGaugeがtrue

2:isTakeGaugeがtrue

float timeSpeed ゲージの加減速度

Image gaugeImage Imageコンポーネント

・private関数

GaugeAddFill () 1秒ごとのゲージ増加量を総ゲージ量に加算し

ImageコンポのFillAmountに代入します

引数:なし

返値:なし

GaugeTakeFill() 1秒ごとのゲージ減少量を総ゲージ量から減算し

ImageコンポのFillAmountに代入します

引数:なし

返値:なし

・public関数

float OnDisplayTime() 現在の秒数を計算し、返値として引き渡します

引数:無し

返値:displayTime 現在の秒数

■使い方

「GameManager」オブジェクトの「Gauge Obj」にImageコンポがアタッチされた

ゲームオブジェクトを登録します

「Total Value」にImage画像の長さ(0 ~ 1)を指定

「Total Time」にゲージが満タン/空になるまでの時間を指定してください

「Is Add Gauge」にチェックが入っているとゲージが増加

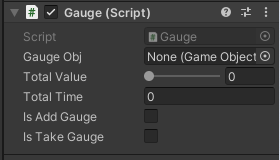
「isTakeGauge」にチェックが入っているとゲージが減少します

両方にチェックが入っているとゲージは変化しません

外部から操作したい場合は「GameManager」オブジェクトから「Gauge」コンポを取得し

「isAddGauge」 Or 「isTakeGauge」のフラグを切り替えてください

■Inspector



・GaugeObj … ゲージとして使用するゲームオブジェクト

Imageコンポーネントがアタッチされている必要があります

アタッチされていないオブジェクトの場合、エラーとなります

・TotalValue … ゲージの長さを指定します

0~1.0までの範囲で指定してください

・TotalTime … ゲージが満タン/空になるまでの時間を指定します

秒単位で指定してください

・IsAddGauge … チェックが入っている場合、ゲージが増加します

・IsTakeGauge … チェックが入っている場合、ゲージが減少します