ライブラリ仕様書

東京ゲームショウ ゲーム開発

CTaskクラス

class CTask

タスクリストの項目となる基底クラスです。

1. メンバ変数

CTask *mpPrev	タスクリストで、自分の前のタスクのポインタ
CTask *mpNext	タスクリストで、自分の後のタスクのポインタ
bool mEnabled	有効フラグ　true:有効　false:無効
int mPriority	タスクの優先度　大きい値程優先度が高く、リストの先頭へ並ぶ

2. メンバメソッド

(1)コンストラクタ

CTask()

mEnabled(true) mPriority(0) mpPrev(0), mpNext(0)

CTask(bool enabled, int priority)

enabled	有効フラグ：mEnabledへ代入
priority	優先度：mPriorityへ代入

mpPrev(0) mpNext(0)

CTaskクラス

（2）デストラクタ
　　virtual ~CTask() {}　デストラクタは仮想関数にすること。

（3）更新処理
　　virtual void Update() {}　1フレームで実行する処理を定義します。

（4）描画処理
　　virtual viod Render() {}　1フレームで描画する処理を定義します。

（5）衝突処理
　　virtual void Collision(CTask* m, CTask* y) { retuan false; }　　衝突処理を定義します。

CTaskManagerクラス

class CTaskManager

タスクリストを管理するクラスです。

1. メンバ変数

CTask *mpHead	タスクリストの先頭タスクへのポインタ
CTask *mpTail	タスクリストの最終タスクへのポインタ
static CTaskManager* mpInstance	タスクマネージャへのポインタ

2. メンバメソッド

(1)コンストラクタ

CTask()

mpHead(0), mpTail(0)

(2)タスクマネージャの取得

CTアskManager* Get()

mpInstanseが0の場合、タスクマネージャを生成し mpInstanceに代入する。mpInstanceの値を返す。

(3)タスクの追加

void Add(CTask* task)

taskをタスクマネージャに追加する、taskのmPriorityが大きい値ほど、先頭に近くします。

CTaskManagerクラス

(4)タスクの更新
　　void Update()
　　　　タスクリストの先頭から順に、タスクの更新処理を呼び出す。

(5)タスクの衝突処理
　　void Collision()
　　　　タスクリストの先頭から順に、タスクの衝突処理を呼び出す。

(6)タスクの描画
　　void Render()
　　　　タスクリストの先頭から順に、タスクの描画処理を呼び出す。

(7)タスクの削除
　　void Remove()
　　　　タスクリストから、無効(mEnabledがfalse)なタスクを削除し deleteする。

　　void Remove(CTask* task)
　　　　taskに該当するタスクを、タスクリストから削除し、 deleteする。

(8)タスクマネージャの破棄
　　void Destory()
　　　　タスクリストのタスクを全て deleteし、タスクマネージャも deleteする。

CRectangleクラス

class CRectangle：public CTask

四角形のデータを保持し描画を行います。

1. メンバ変数

CVector2 mPosition	四角形の中心座標	.x 四角形の中心のX座標　.y 四角形の中心のY座標
CVector2 mScale	四角形の幅と高さ	.x 四角形の中心からX軸方向への幅 .y 四角形の中心からY軸方向への高さ
float mRotation	四角形の回転角度	四角形の回転角度
CTexture *mpTexture	四角形が使用するテクスチャへのポインタ	
float mUv[4]	テクスチャマッピングデータ	0:左座標 1:右座標 2:下座標 3:上座標

2. メンバメソッド

(1)コンストラクタ

CRectangle(const CVector2& position, const CVector2& scale, CTexture* texture)

position	四角形の中心座標：m Positionへ代入
scale	四角形の幅と高さ: mScaleへ代入
texture	テクスチャへのポインタ　0の場合はテクスチャなし ポインタが入力された場合、mUvを画像の大きさで設定する

CRectangleクラス

（2）四角形の描画

void Render()

四角形を描画する。

テクスチャが有る場合、テクスチャをマッピングして描画する。

CCollisionクラス

class CCollision

衝突判定を行います。

1. メンバ変数
　　なし

2. メンバメソッド
（1）衝突判定処理
　　static bool Collision(CRectangle* rect1, CRectangle* rect2)
　　　　rect1　　四角形へのポインタ
　　　　rect2　　四角形へのポインタ

　　　　戻り値　　rect1とrect2が重なっている場合、trueを、それ以外はfalseを返します。