

leoapi のちゅかいかた

いんとろだくしょん

leoapi は gpong をつくるために開発されたグラフィカルなゲームを作るために開発された api だよ。要望があれば拡張していくよ。

使い方

スタティックライブラリ、又はシェアードライブラリとしてつかえるよ。

仕組み

Qt を使って画面に表示を使っているよ！だから理屈でわ、いろいろなアーキテクチャでうごくよ！

API を使うときは、lapi というディレクトリの中にある leoapi.h をインクルードしてね！

lapiobj というオブジェクトを使うことで、画像を画面に表示させることができるよ！

lapiphy っていうオブジェクトを lapiobj から生成すると、lapiobj に物理的な挙動をつけることができるよ！

画面は、3つのレイヤーから出来ているよ！バックグラウンド用のレイヤー(0)、通常のオブジェクト用のレイヤー(1)、特殊なオブジェクトの為のエディタ(2)だよ！ 数字の大きなレイヤーに置かれたものが画面上で上に表示されるよ！

オブジェクト

lapiobj	画面に画像を表示させるよ！
lapiphy	lapiobj に物理的挙動をつけるよ！
lapiLayer	lapiobj を置くためのレイヤーだよ！
lapiImage	画像を管理するためのオブジェクトだよ！

便利なかんすー

システム関数

- `lapiLayer lapiGetLayer(int index)`
- `int lapiGetFps()`
- `void lapiSetFps(int fps)`
- `lapiImage _loadImage(const char *fileName)`
- `void lapiSetKeyListener(void (*handler)(int flag,int keyType))`
- `void lapiRemoveKeyListener(void (*handler)(int flag,int keyType))`
- `void lapiSetNewFrameEvent(void (*newFrame)())`
- `void lapiRemoveNewFrameEvent(void (*newFrame)())`
- `int lapiGetWidth()`
- `int lapiGetHeight()`
- `void lapiSetSize(int width,int height)`
- `int lapiGetX()`
- `int lapiGetY()`
- `void lapiSetLocation(int x,int y)`
- `lapiobj lapiGetBall()`
- `lapiobj lapiGetPRacket()`
- `lapiobj lapiGetORacket()`

lapiobj 関数

- `lapiobj objGetFromLayer(lapiLayer layer,const char *key)`
- `lapiobj objInit(const char *fileName)`
- `void objPutOnLayer(lapiLayer layer,const char *key,lapiobj obj)`
- `void objRemoveKeyFromLayer(lapiLayer layer,const char *key)`
- `void objRemoveThisFromLayer(lapiLayer layer,lapiobj obj)`
- `int objGetWidth(lapiobj)`
- `int objGetHeight(lapiobj)`
- `int objGetSW(lapiobj)`
- `int objGetSH(lapiobj)`
- `int objGetSX(lapiobj)`
- `int objGetSY(lapiobj)`
- `int objGetX(lapiobj)`
- `int objGetY(lapiobj)`
- `double objGetTurnRadian(lapiobj)`

- `double objGetTurnDegree(lapiobj)`
- `double objGetTurn(lapiobj)`
- `void objTurnRadian(lapiobj,double radian)`
- `void objTurnDegree(lapiobj,double degree)`
- `void objSetSize(lapiobj,double newSize)`
- `void objSetSW(lapiobj,int val)`
- `void objSetSH(lapiobj,int val)`
- `void objSetSX(lapiobj,int val)`
- `void objSetSY(lapiobj,int val)`
- `void objSetX(lapiobj,int x)`
- `void objSetY(lapiobj,int y)`
- `void objSetImageFromFile(lapiobj,const char *fileName)`
- `void objSetImage(lapiobj,lapiImage)`
- `void objSetMouseListener(lapiobj,void (*listener)(lapiobj,int flag))`
- `void objRemoveMouseListener(lapiobj,void (*listener)(lapiobj,int flag))`
- `void objFinallize(lapiobj)`

lapiphy 関数

- `lapiphy phyInit(lapiobj,const char *name)`
- `unsigned int phyGetFlag(lapiphy *)`
- `void phyMoveVector(lapiphy *,double x,double y)`
- `void phyMoveTo(lapiphy *,int x,int y,double second)`
- `void phyTurnToRadian(lapiphy *,double radian,double second)`
- `void phyTurnToDegree(lapiphy *,double degree,double second)`
- `void phyTurnRadian(lapiphy *obj,double radian)`
- `void phyTurnDegree(lapiphy *obj,double degree)`
- `void phyMoveStop(lapiphy *)`
- `void phySetHitListener(lapiphy *obj,int (*whenHit)(lapiphy *,lapiphy *))`
- `void * phyGetHitListener(lapiphy *obj)`
- `void phySetName(lapiphy *obj,const char *name)`
- `void phyAddTarget(lapiphy *obj,const char *target)`
- `int phyGetX(lapiphy *obj)`
- `int phyGetY(lapiphy *obj)`
- `int phyGetXCenter(lapiphy *obj)`

- `int phyGetYCenter(lapiphy *obj)`
- `int phyGetWidth(lapiphy *obj)`
- `int phyGetHeight(lapiphy *obj)`
- `const char * phyGetName(lapiphy *obj)`
- `char ** phyGetTargets(lapiphy *obj)`
- `int phyGetTargetsCount(lapiphy *obj)`
- `lapiobj phyGetlapiObject(lapiphy *obj)`
- `void phyDisableDefaultCollisionBehaviour(lapiphy *obj, bool disable)`
- `void phyDisableCollision(lapiphy *obj, bool disable)`
- `void phyDisableMovement(lapiphy *obj, bool disable)`
- `lapiobj phyFinalize(lapiPhysicalObject *obj)`

名前	返回值	引数	説明
LapiGetLayer	lapiLayer オブジェクト	レイヤーのインデックス	指定されたレイヤーを返す
lapiGetFps	現在の FPS		現在設定されている FPS の値を返す
LapiSetFps		新しい FPS	新しく FPS を設定する
_loadImage	新しくロードされた lapiImage オブジェクト	画像ファイルへのパス	新しく画像ファイルをロードする
lapiSetKeyListener		関数へのポインタ	新しく、キーが押されたときのイベントハンドラを設定する。
lapiRemoveKeyListener		関数へのポインタ	キーが押された時のイベントハンドラを取り除く
LapiSetNewFrameEvent		関数へのポインタ	新しいフレームが描画される時のイベントハンドラを設定する
lapiRemoveNewFrameEvent		関数へのポインタ	新しいフレームが描画される時のイベントハンドラを取り除く
lapiGetWidth	ウィンドウの横の長さ		現在のウィンドウの横の長さを返す
lapiGetHeight	ウィンドウの縦の長さ		現在のウィンドウの縦の長さを返す
lapiSetSize		1) 新しいウィンドウの横の長さ 2) 新しいウィンドウ	新しいウィンドウのサイズを決定する

		の縦の長さ	
lapiGetX	ウインドウの位置(X)		ウインドウのX座標を返す
lapiGetY	ウインドウの位置(Y)		ウインドウのY座標を返す
lapiSetLocation		1) 新しいウインドウのX座標 2) 新しいウインドウのY座標	ウインドウの位置を指定する
lapiGetBall	ball オブジェクト		ball オブジェクトを返す
lapiGetPRacket	プレイヤーのラケット		プレイヤーのラケットを返す
lapiGetORacket	相手のラケット		相手のラケットを返す
objGetFromLayer	指定されたオブジェクト、又は NULL	1) レイヤーオブジェクト 2) オブジェクトを指定するキー	オブジェクトをレイヤーから取り出す。キーを指定して取り出すことが可能
objInit	新しい lapiobj	画像ファイルへのパス	新しく lapiobj を生成する。
objPutOnLayer		1) オブジェクトを置くレイヤー 2) キー 3) 置きたいオブジェクト	オブジェクトをレイヤーの上に置く、キーを指定することで用意に取り出すことが出来る。
objRemoveKeyFromLayer		1) レイヤー 2) 取り除きたいオブジェクトのキー	キーを指定してオブジェクトをレイヤーから取り除く
objRemoveThisFromLayer		1) レイヤー 2) オブジェクト	指定されたオブジェクトをレイヤーより取り除く。(少し遅い)
objGetWidth	オブジェクトの横の長さ	オブジェクト	指定されたオブジェクトの横の長さを返す
objGetHeight	オブジェクトの縦の長さ	オブジェクト	指定されたオブジェクト縦の長さを返す
objGetSW	オブジェクトのX軸の始点	オブジェクト	指定されたオブジェクトのX軸の始点を返す
objGetSH	オブジェクトのY軸の始点	オブジェクト	指定されたオブジェクトのY軸の始点を返す
objGetSX	オブジェクトのX軸の終点	オブジェクト	指定されたオブジェクトのX軸の終点を返す
objGetSY	オブジェクトのY軸の終点	オブジェクト	指定されたオブジェクトのY軸の終点を返す

objGetX	オブジェクトのX座標	オブジェクト	指定されたオブジェクトのX座標を返す
objGetY	オブジェクトのY座標	オブジェクト	指定されたオブジェクトのY座標を返す
objGetTurnRadian	オブジェクトの回転ラジアン数	オブジェクト	指定されたオブジェクトがどのくらい回転しているかをラジアンで返す
objGetTurnDegree	オブジェクトの回転度数	オブジェクト	指定されたオブジェクトがどのくらい回転しているかを度数で返す
objTrunRadian		1) オブジェクト 2) ラジアンの値	指定されたオブジェクトをラジアンの値で回転する。
objTurnDegree		1) オブジェクト 2) 回転する度数	指定されたオブジェクトを度数で回転する。
objSetSize		1) オブジェクト 2) 新しいオブジェクトのサイズへの率	指定されたオブジェクトのサイズを何倍にするかを指定する。
objSetSW		1) オブジェクト 2) 新しい始点	与えられたオブジェクトのX軸始点を指定する
objSetSH		1) オブジェクト 2) 新しい始点	与えられたオブジェクトのY軸始点を指定する
objSetSX		1) オブジェクト 2) 新しい終点	与えられたオブジェクトのX軸終点を指定する
objSetSY		1) オブジェクト 2) 新しい終点	与えられたオブジェクトのY軸終点を指定する
objSetX		1) オブジェクト 2) 新しいX座標	与えられたオブジェクトのX軸を指定する
objSetY		1) オブジェクト 2) 新しいY座標	与えられたオブジェクトのY軸を指定する
objSetImageFromFile		1) オブジェクト 2) ファイルへのパス	与えられたオブジェクトの画像をファイルから新しく設定する
objSetImage		1) オブジェクト 2) イメージオブジェクト	与えられたオブジェクトの画像を設定する
objSetMouseListener		1) オブジェクト 2) 関数へのポインタ	与えられたオブジェクトにマウスリスナーを設定する
objRemoveMouseListener		1) オブジェクト 2) 関数へのポインタ	与えられたオブジェクトから指定されたマウスリスナーを取り除く

objFinallize		オブジェクト	与えられたオブジェクトを破棄する
phyInit	新しい lapiphy オブジェクト	1) lapiphy の生成に使用する lapiobj 2) lapiphy オブジェクトの名前、ターゲットの指定に使用される	lapiphy を与えられたオブジェクトから生成する
phyGetFlag	lapiphy 内部の flag	オブジェクト	lapiphy より flag を取り出す。(高度な知識を持ったユーザ用)
phyMoveVector		1) オブジェクト 2) 移動したい長さ(X) 3) 移動したしたい長さ(Y)	1 秒間に何ピクセル動かかを指定する。 (pixel/second) オブジェクトにベクトルを与える。
phyMoveTo		1) オブジェクト 2) 移動したい座標(X) 3) 移動したい座標(Y) 4) 移動にかかる時間(秒)	指定された秒数の間に、指定された座標まで移動する。
phyTurnToRadian		1) オブジェクト 2) 回転する大きさ(ラジアン) 3) 回転にかかる秒数	回転する大きさを指定し、その回転にかかる時間も指定する。
phyTurnToDegree		1) オブジェクト 2) 回転する大きさ(度数) 3) 回転にかかる秒数	回転する大きさを指定し、その回転にかかる時間も指定する。
phyTurnRadian		1) オブジェクト 2) 角速度(radian/second)	オブジェクトの角速度をラジアンで指定する
phyTurnDegree		1) オブジェクト 2) 角速度(degree/second)	オブジェクトの角速度を度数で指定する
phyMoveStop			
phySetHitListener			
phyGetHitListener			
phySetName			

phyAddTarget			
phyGetX			
phyGetY			
phyGetXCenter			
phyGetYCenter			
phyGetWidth			
phyGetHeight			
phyGetName			
phyGetTargets			
phyGetTargetsCount			
phyGetIapiObject			
phyDisableDefaultCollisionBehaviour			
phyDisableCollision			
phyDisableMovement			
phyFinalize			