

leoapi のちゆかいかた

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 radian)`
- `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)`