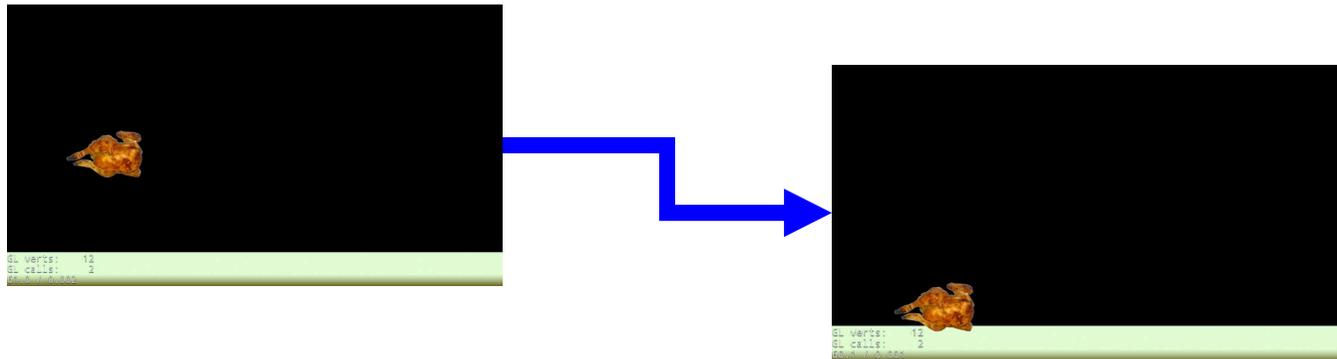


Cocos2d-xで作る物理演算ゲーム ～ 引っ張って飛ばす編 ～

= 2015年09月12日 =

前回のあらすじ

物理演算ライブラリchipmunkを使って
チキンが地面に落ちるところまで。。



今回は**チキン**を
引っ張って飛ばします

これ



チキンにタッチ
して・・・

スライド

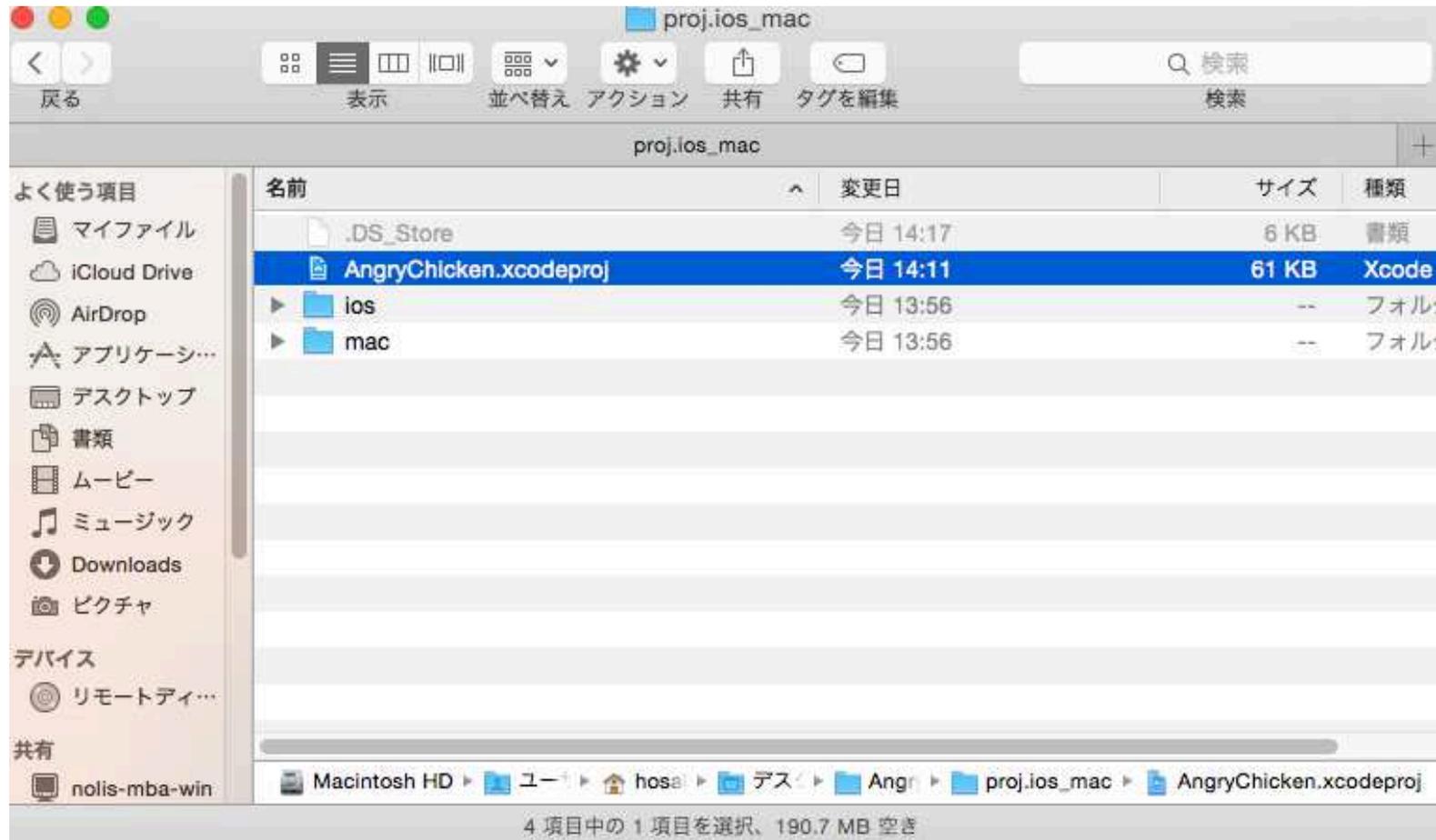


飛ぶ！

離す！

まずは起動しよう

AngryChickenをxcodeで実行。



やる処理を日本語でまとめる

- ①画面へのタッチイベントを取得する。
- ②チキンにタッチを判定する。
- ③スライドして引っ張るとチキンが飛ぶ！

①画面へのタッチイベントを取得する。

HelloWorld::initメソッドに以下のコードを追加

```
// Touchイベントを管理しているEventDispatcherインスタンスを取得
auto eventDispatcher = Director::getInstance()->getEventDispatcher();

// 登録するリスナー情報を生成
auto listener = EventListenerTouchOneByOne::create();
listener->onTouchBegan = CC_CALLBACK_2(HelloWorld::_onTouchBegan, this);
listener->onTouchMoved = CC_CALLBACK_2(HelloWorld::_onTouchMoved,
this);
listener->onTouchEnded = CC_CALLBACK_2(HelloWorld::_onTouchEnded, this);
listener->onTouchCancelled =
CC_CALLBACK_2(HelloWorld::_onTouchCancelled, this);

// リスナー情報を登録
eventDispatcher->addEventListenerWithSceneGraphPriority(listener, this);
```

EventDispatcherというシングルトンクラスがイベントを管理している。
EventListenerTouchOneByOneクラスを生成し、コールバック関数を登録し、**EventDispatcher**に登録する。

当然コールバックされるメソッドをHelloWorld.cppに作りましょう

```
//タッチ開始 最初に1度だけ呼ばれる
bool HelloWorld::_onTouchBegan(Touch* touch, Event* event)
{
    // falseを返すと以降のタッチイベントは呼ばれない
    return true;
}
// ドラッグする最中ずっと呼ばれる
void HelloWorld::_onTouchMoved(Touch* touch, Event* event)
{
}
// タッチを離すと呼ばれる
void HelloWorld::_onTouchEnded(Touch* touch, Event* event)
{
}
// タッチがキャンセルされると呼ばれる
void HelloWorld::_onTouchCancelled(Touch* touch, Event* event)
{
    // キャンセル時も一応線を削除
}
}
```

キャンセルは、システムメッセージなどが表示された際になど呼ばれるようだ。（バッテリー警告とかあの辺りと思われる）

当然HelloWorld.hにもメソッド定義を書きましょうね。

```
private:
```

```
    // Touchイベント用
```

```
    bool _onTouchBegan(cocos2d::Touch* touch, cocos2d::Event* event);
```

```
    void _onTouchMoved(cocos2d::Touch* touch, cocos2d::Event* event);
```

```
    void _onTouchEnded(cocos2d::Touch* touch, cocos2d::Event* event);
```

```
    void _onTouchCancelled(cocos2d::Touch* touch, cocos2d::Event* event);
```

*ちなみに・・・

私はprivateなフィールドやメソッドには「アンダーバー」を規則でつけます。

**ここまでで画面をタッチすると
各メソッドが呼ばれます。**

②チキンにタッチを判定する。

もう少しひも解くところなる

タッチしたポイント座標 (x,y) が
チキン画像の縦横幅内
だったらイロイロ処理をする。

つまり上記条件を満たさない場合は
引っ張り処理はしないということ

チキンノードを取得できるように

helloWorld::init のチキン読み込み部分のsetTagを赤文字に変更

```
// キャラクター
auto* character = Sprite::create("chicken.png");
character->setPosition(Point(winSize.width / 5 , winSize.height / 2));
auto* charaPb = PhysicsBody::createCircle(40);
charaPb->setMass(1.0f);          // 重さを指定（ここが無いと後で飛ばせなくなる）
character->setPhysicsBody(charaPb);
character->setTag(CHAR_OBJTAG);
this->addChild(character);
```

helloWorld.cppの冒頭に以下のマクロ定義

```
// tag list
#define CHAR_OBJTAG 100
```

HelloWorld::_onTouchBeganに以下の処理を追加

```
// チキンの矩形情報を求める
auto* charSprite = (Sprite*)this->getChildByTag(CHAR_OBJTAG);
Point point = charSprite->getPosition();
int width = charSprite->getContentSize().width;
int height = charSprite->getContentSize().height;
Rect charSpriteRect = Rect(point.x - (width / 2), point.y - (height / 2), width,
height);

//タッチの位置を取得
auto tPos = touch->getLocation();

// タッチ位置がチキン矩形内か判定
if ( charSpriteRect.containsPoint(tPos) == true )
{
    // チキンがタッチされた！
    // ここに処理をいろいろと書くのだよ。
    return true;
}

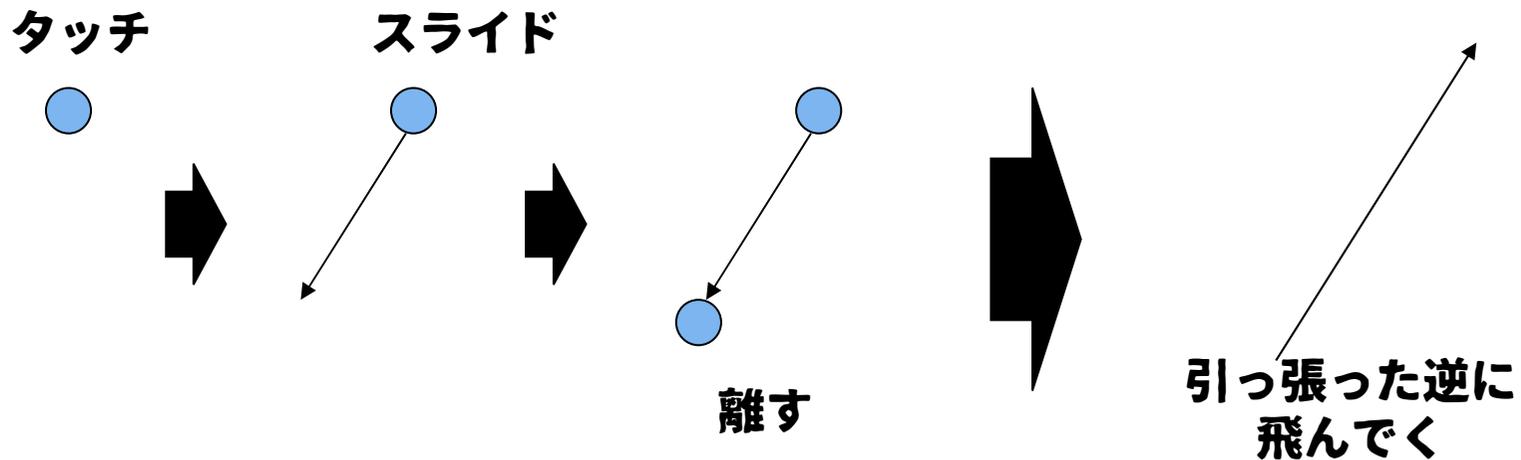
// ここまで処理が来たということはチキンがタッチされなかったということ。
// falseを返すと以降のタッチイベントは呼ばれない
return false;
```

上記でメソッドでtrueを返すということは、moved, endedがよばれることとなる。つまりmoved,ended,cancelledにはチキンがタップされたときにしか呼ばれないという状態になるのだ。

**③スライドして引っ張ると
チキンが飛ぶ！**

少しだけ掘り下げて表現

タッチした箇所（始点）、離れた箇所（終点）
その2つを結ぶ線がいわゆるチキンが飛ぶ方向なのである



HelloWorld::_onTouchEndedへ以下を追加

```
// チキン画像の座標を取得する。
auto* charSprite = (Sprite*)this->getChildByTag(Char_ObjTag);
auto charPos = charSprite->getPosition();

// タッチ終了の座標を取得する。
auto endPos = touch->getLocation();

// 2つの座標からベクトルを求める。(x4は係数)
auto force = Vect(charPos.x - endPos.x, charPos.y - endPos.y) * 4;

// チキンに力を加える
charSprite->getPhysicsBody()->applyImpulse(force);
```

***ベクトルがわからない方は下を読んどきましょう。**

http://www.geocities.co.jp/Technopolis/1505/vx_index.htm

実行してみよう！



チキンにタッチ
して・・・

スライド



飛ぶ！

離す！

***線は出ません。**

**次回は物理演算Chipmunk
スクロールして～編**

ここまでのソースプログラムはここ

引っ張り線描画とリファクタリング済み

<http://monolizm.com/sab/src/AngryChicken.zip>

ご清聴ありがとうございました。