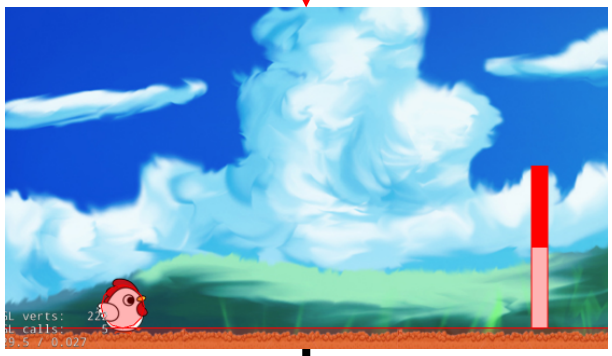
Cocos2d-xで作る物理演算ゲーム

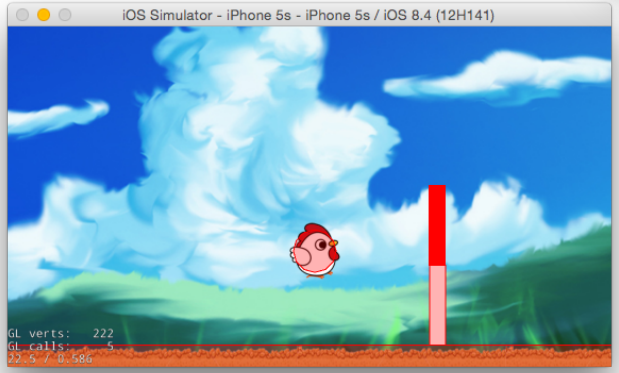
軌跡の点線を入れてみる編

= 2016年3月12日 =

前回のあらすじ

静止したら
自動的に初期化





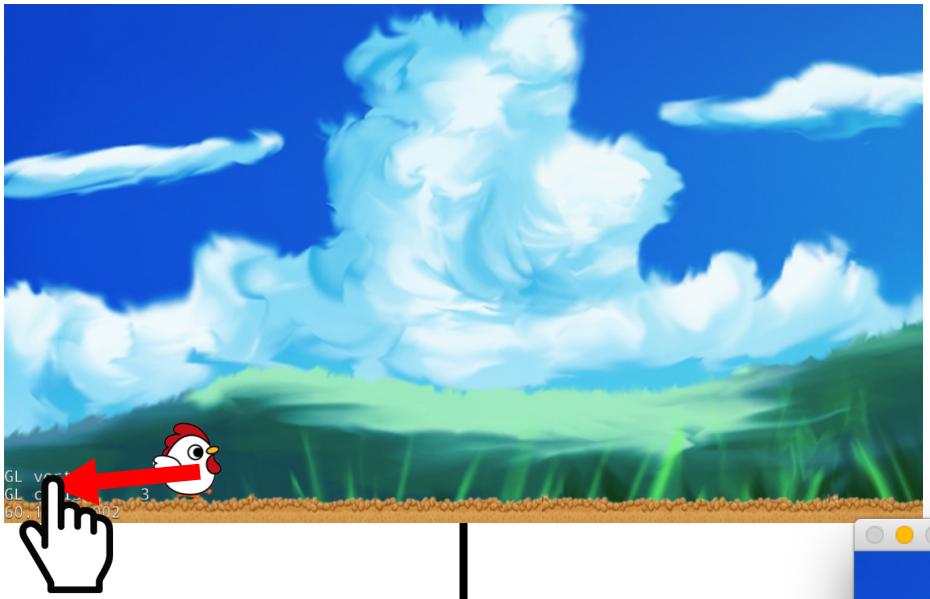
ここまでのソースプログラムはここ

http://monolizm.com/sab/src/AngryChicken16.zip

GETだぜ！

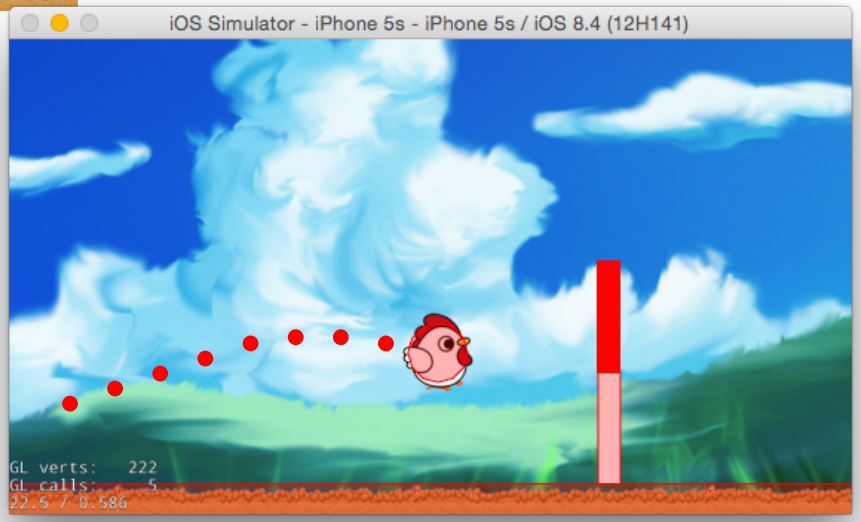
今回は軌跡の点線を
いれてみる。

今回はコレ



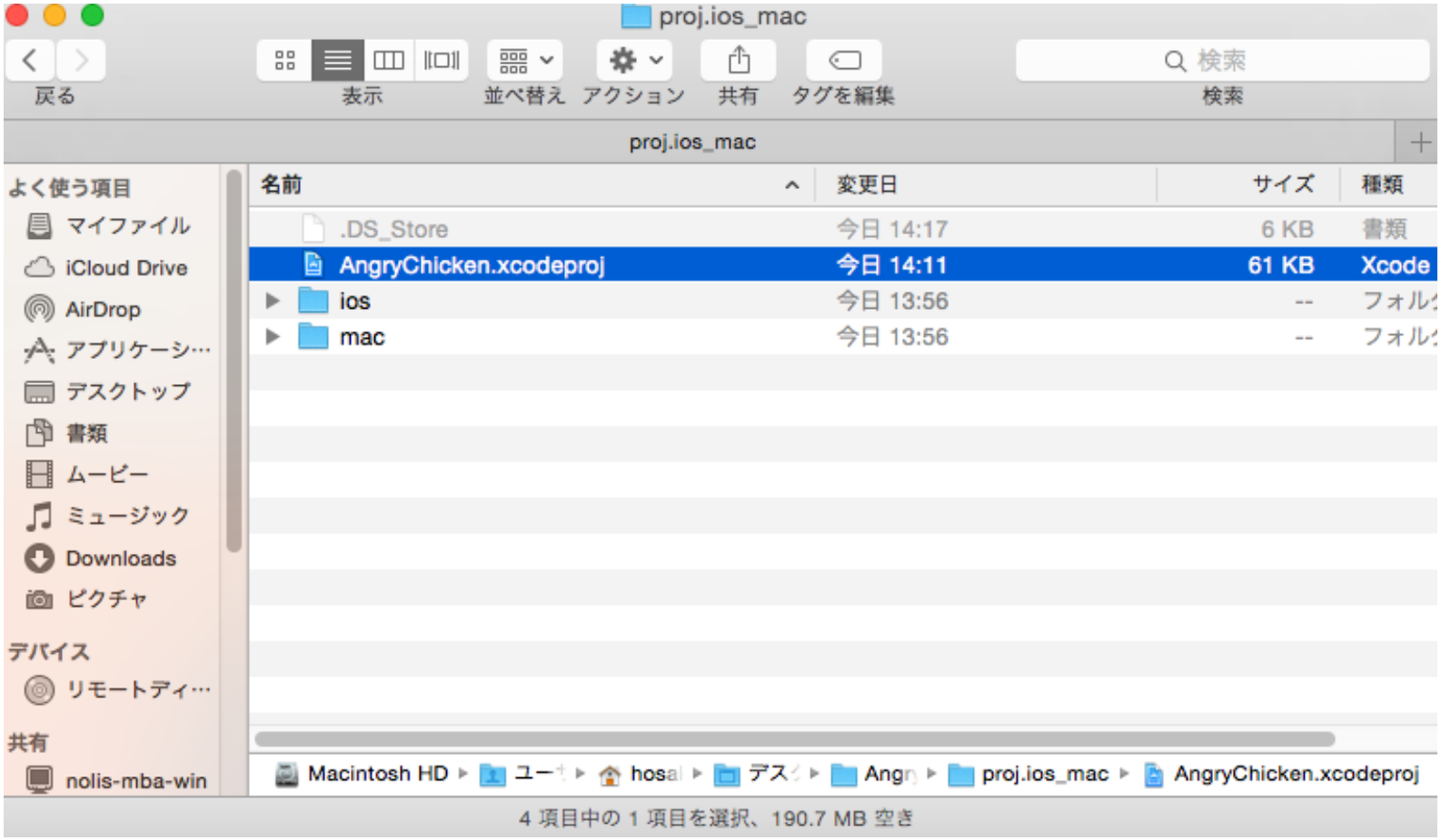
チキンを引っ張り･･･

飛ばすと･･･軌跡が！！！



まずは起動しよう

AngryChickenをxcodeで実行。



まずは必要な処理を考えてみる

①点を打つ
②チキンの座標をトレース

①点を打つ

何はともあれ「画面上の任意の位置に複数の点を打つ」にはどのようにすればいいだろうか？

おおよそ最近のゲーム開発の基礎的ライブラリには基本図形を描画するための便利な機構があるはずである。

cocos2d-xのクラスで便利なクラスがやり方がないか探す前に以下のクラスを引っ張り線で使っていたことを思いだす。

DrawNodeクラス

点、線、三角形、四角形、それ以上の多角形などを描画することができる！

サンプルコードを見てみる

ネットで調べてみたところこんな感じで点を画面上の任意の位置に表示できる模様。

```
auto draw = DrawNode::create();
draw->drawDot(Vec2(100,60), 10.0f, Color4F(1.0f, .6f, .5f, 1.0f));
// drawDotは引数として位置, 半径, 色を指定する。
```

ただしもう少し突っ込むと複数の点を打ちたい場合はどうするかというと。。。

```
auto draw = DrawNode::create();
draw->drawDot(Vec2(100,60), 10.0f, Color4F(1.0f, .6f, .5f, 1.0f));
draw->drawDot(Vec2(200,20), 10.0f, Color4F(1.0f, .6f, .5f, 1.0f));
draw->drawDot(Vec2(300,50), 10.0f, Color4F(1.0f, .6f, .5f, 1.0f));
```

それは簡単。複数drawDotを呼び出してあげればよい。

②チキンのトレース

チキンの座標を一定間隔で取得すればいいことになる。チキンの座標の取得はいままでやってきた方法で・・・

```
auto* charSprite = (Sprite*)this->getChildByTag(CHAR_OBJTAG);
//charSprite->getPosition()で取得できる。
```

後は一定間隔でこの位置に点を描画すればよい。

というわけで今回は3フレーム単位で点を打つことにします。

ちょっとティーブレイク
フレームとは？

ゲーム画面も実はパラパラ漫画と同じ。つまり1秒間に60回コマを高速で表示しているのです。このコマを転じて「フレーム」といいます。（＊TVやハードウェアによって1秒間のフレーム数は異なってきます）

1秒間、何フレームで描画されているか？という速度を表す単位としてFPSが使われます。例えば1秒間に60フレームなら60fpsといいます。

これはゲーム開発において重要な概念。覚えましょう！

でこうなる。軌跡を打つ用のDrawNodeはあらかじめつくって描画ツリーへ追加（_commonInit）

```
void MainGame::_commonInit()
{
～～～
～～～
～～～
    //描画ツリーへ追加しとく 0228
    auto draw = DrawNode::create();
    draw->setTag(LOCUS_OBJTAG);
    this->addChild(draw);
～～～
～～～
～～～
～～～
}
```

というわけでこうなる。

```
void MainGame::update(float delta)
{
    auto* draw = dynamic_cast<DrawNode*>(this->getChildByTag(PULL_LINE_OBJTAG));
    if ( draw != nullptr )
    {//存在した場合＝ひっぱり中なのでひっぱりラインを更新
                        ～～省略～～
    } else
    { // 0228
        if ( this->_touchEndedFlag == true )
        {// 飛ばし済み
            this->_cFrame++;                    // カウントアップ
            if ( this->_cFrame > 3 )            // 3フレーム経過
            {
                this->_cFrame = 0;              // カウントリセット
                // 点線を一定間隔で打つ
                auto* draw = dynamic_cast<DrawNode*>(this->getChildByTag(LOCUS_OBJTAG));
                if ( draw != nullptr )
                {//正常に取得できたなら点を描画
                    auto* charSprite = (Sprite*)this->getChildByTag(CHAR_OBJTAG);
                    draw->drawDot(charSprite->getPosition(), 10, Color4F(1.f,0.f,0.f,1.f));
                }
            }

            // 動作が止まっているか判定する
            auto* charSprite = (Sprite*)this->getChildByTag(CHAR_OBJTAG);
            auto velocity = charSprite->getPhysicsBody()->getVelocity();
            if ( velocity.length() < 1 )
            {// リセットする。
                _commonInit();
            }
        } else
        {// 飛ばし前

        }
    }


}
```

＊新たに_cFrame変数が追加し適切な位置で初期化しています。

ここで1つわかったことが。

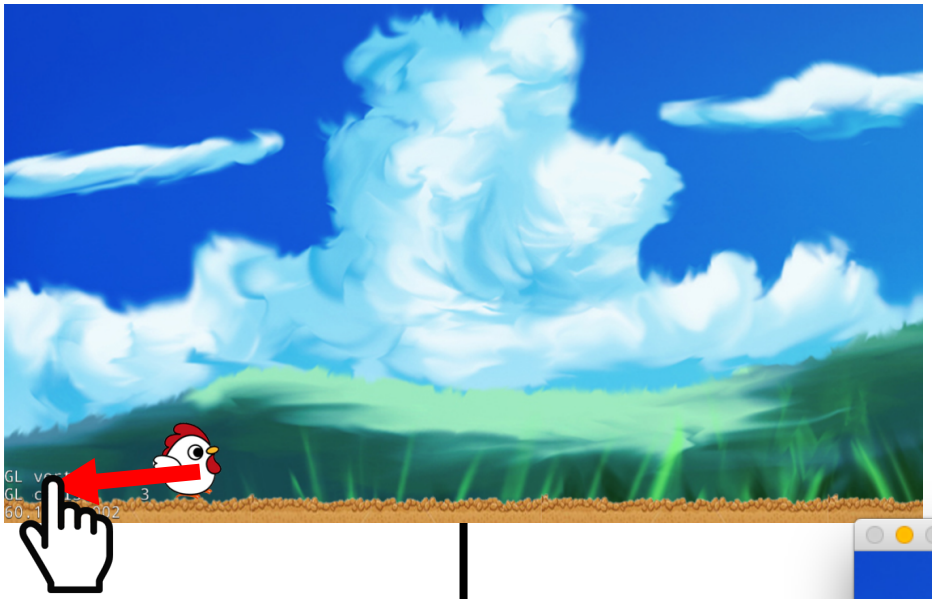
DrawNodeの特徴

描画が過去のモノも蓄積されていく挙動をします。つまり、過去描画されたモノはそのまま描画され続けます。

さらに補足をすると引っ張り線が1本しか毎回でないのは、毎フレーム
「DrawNode::clear()」を呼び出しているからです。

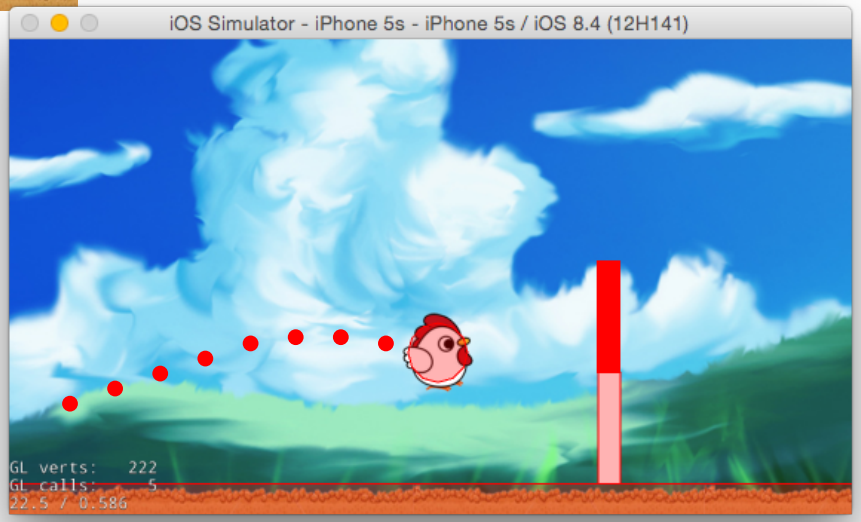
実行してみよう！

こうなる



チキンを引っ張り･･･

飛ばすと･･･軌跡が！！！



んが！！？？

点がずっと出てしまうのでこれは次の回でどうにかしましょう。

まとめ

DrawNodeは基本図形を描画することができる。
DrawNodeの基本図形描画は蓄積される。

フレームとはいわゆるパラパラ漫画の1コマとコマにあたるゲーム開発の言葉。ゲームは高速でパラパラが行われている。1秒間に60フレーム場合は60fpsという。

次回は物理演算Chipmunk
軌跡の点線を入れてみる②編

ここまでのソースプログラムはここ

http://monolizm.com/sab/src/AngryChicken17.zip

ご清聴ありがとうございました。