

Cocos2d-xで作る物理演算ゲーム

敵の処理編

= 2016年11月16日 =

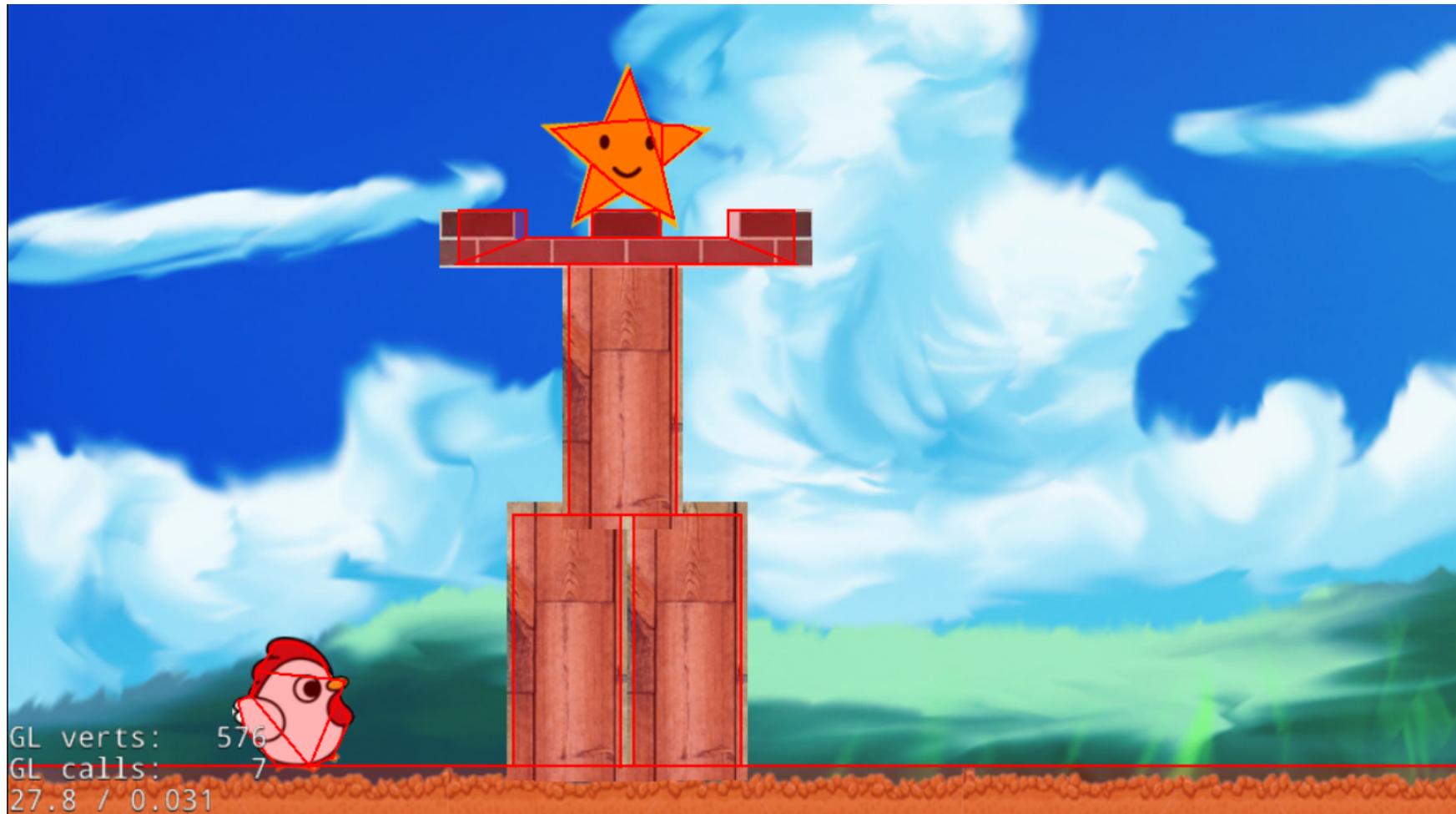
ここまでのソースプログラムはここ

<http://monolizm.com/sab/src/AngryChicken26.zip>

GETだぜ！

今回から
敵の処理の実装
を行っていきます。

今回は画面的には変わりません



敵の処理で気になること

敵を倒すということ。条件。

つまり、一定以上の高さから落ちると死ぬとか、
建造物が倒れて挟まれて死ぬとか。

>これをどのように表現するのか？

の前に、ちょっと1点の改良を

今はステージリセットが・・・
「自キャラの動きが止まったら」
なので、建造物とかが動いている
ときでもリセットされてしまう。

==気持ち悪いのでこれを治します。

本来は・・・

敵・建造物・自キャラの動きが止まったら
リセットされる。

* 厳密にいうと止まったら数秒のディレイをい
れるべきだけど、今は無視

なので、その処理実装

前のソース

```
void MainGame::update(float delta)
{
    auto* draw = dynamic_cast<DrawNode*>(this->getChildByTag(PULL_LINE_OBJTAG));
    ~~~~~
    ~~~~~

    // 自キャラ
    auto* charSprite = (Sprite*)this->getChildByTag(CHAR_OBJTAG);
    auto velocity = charSprite->getPhysicsBody()->getVelocity();
    if ( velocity.length() >= 1 )
    { // リセットする。
        _commonInit();

        // 0617
        this->_swapTHDrawer->swap();
    }
    } else
    { // 飛ばし前
    }
}
}
```

関数化しちゃおう

```
void MainGame::update(float delta)
{
    auto* draw = dynamic_cast<DrawNode*>(this->getChildByTag(PULL_LINE_OBJTAG));
    ~~~~~
    ~~~~~

    if ( _isAllObjectStop() == true )
    { // リセットする。
        _commonInit();

        // 0617
        this->_swapTHDrawer->swap();
    }
    } else
    { // 飛ばし前
    }
}
}
```

事前の準備

```
#define CHAR_OBJTAG 100
#define BLOCKS_OBJTAG 101
#define PULL_LINE_OBJTAG 200
#define LOCUS_OBJTAG 300 // 0228
#define LOCUS_HISTORY_OBJTAG 310 // 0525
```

```
void MainGame::_createBlocks()
{
    Node* node = CSLoader::getInstance()->createNode("MainScene.csb");
    node->setTag(BLOCKS_OBJTAG);
    this->addChild(node);
}
```

```
class MainGame : public cocos2d::Layer
{
public:
    ~~~~~
    ~~~~~
    void _commonInit();
    void _createBlocks();
    bool _isAllObjectStop();

    // 軌跡履歴用
    // DrawNode* _historyDraw;
    ~~~~~
}
```

関数の中身

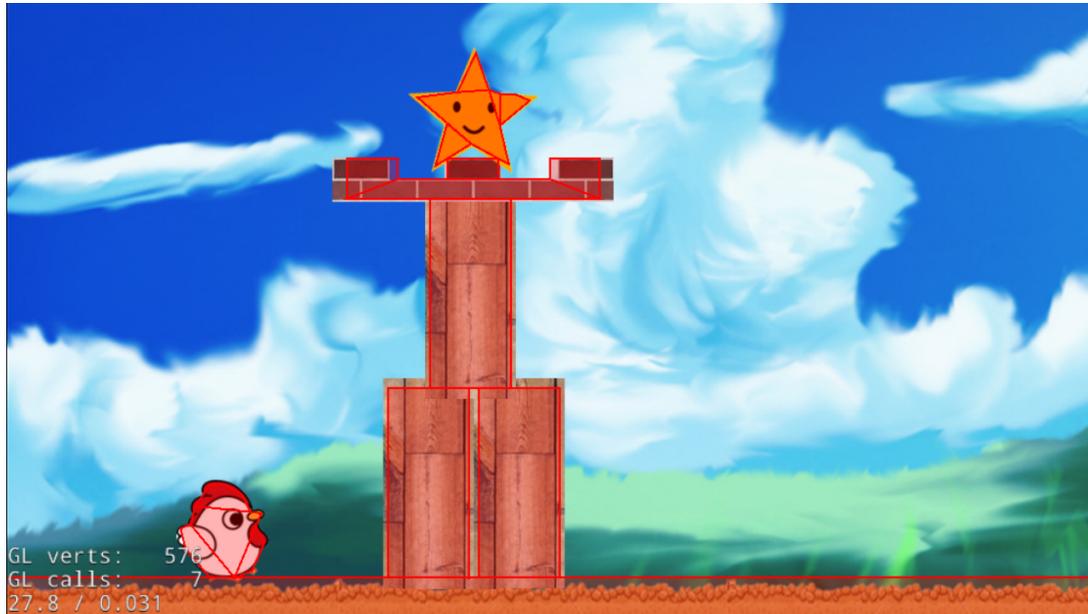
```
bool MainGame::_isAllObjectStop()
{
    // 全てのオブジェクトが静止しているかかくにんする。

    // 自キャラ
    auto* charSprite = (Sprite*)this->getChildByTag(CHAR_OBJTAG);
    auto velocity = charSprite->getPhysicsBody()->getVelocity();
    if ( velocity.length() >= 1 )
    {
        return false;
    }

    // 建物
    auto rootBlocks = this->getChildByTag(BLOCKS_OBJTAG);
    Vector<Node*> children = rootBlocks->getChildren();
    Vector<Node*>::iterator it = children.begin(); // イテレータのインスタンス化
    while( it != children.end() ) // 末尾要素まで
    {
        auto pb = (*it)->getPhysicsBody();
        if ( pb != nullptr )
        {
            auto velocity = (*it)->getPhysicsBody()->getVelocity();
            if ( velocity.length() >= 1 )
            {
                return false;
            }
        }
        ++it;
    }

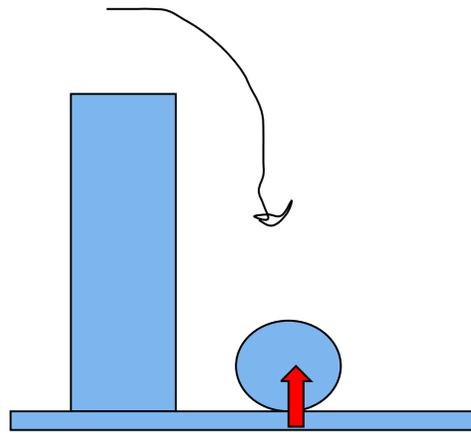
    return true;
}
```

とりあえずOKなはず。

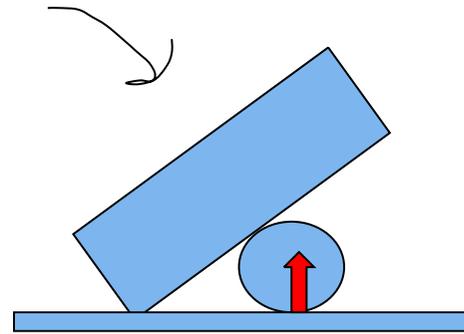


* 画面外に出ると永久に落下するので一生リセットされません。それはまたいずれ修正

さて本題。敵の衝突をどうやってやるか？



高いところから落ちて、
地面の衝突した力



何かがぶつかって、押され
る力(挟まる?)

敵キャラに一定の衝撃があったら……。
言い換えると、敵キャラ自身に「もともと加わっている力とは、反対方向への力が加わったら」となるのでは？

もう少し思案・・・

やり方としては・・・

- ・もともと敵キャラにかかっている力を取得する。
- ・新たにかかった力を取得する。
- ・2つの力のベクトルが反対？で、かつ一定の強さ？だったら死ぬ

こんな感じと予想！

いざ実装！・・・は次回へ・・・

次回は物理演算Chipmunk
敵の処理編

ここまでのソースプログラムはここ

<http://monolizm.com/sab/src/AngryChicken29.zip>

ご清聴ありがとうございました。