

就職作品 Break Targets

京都大学 工学部 工業化学科 岡田成広史

自己紹介

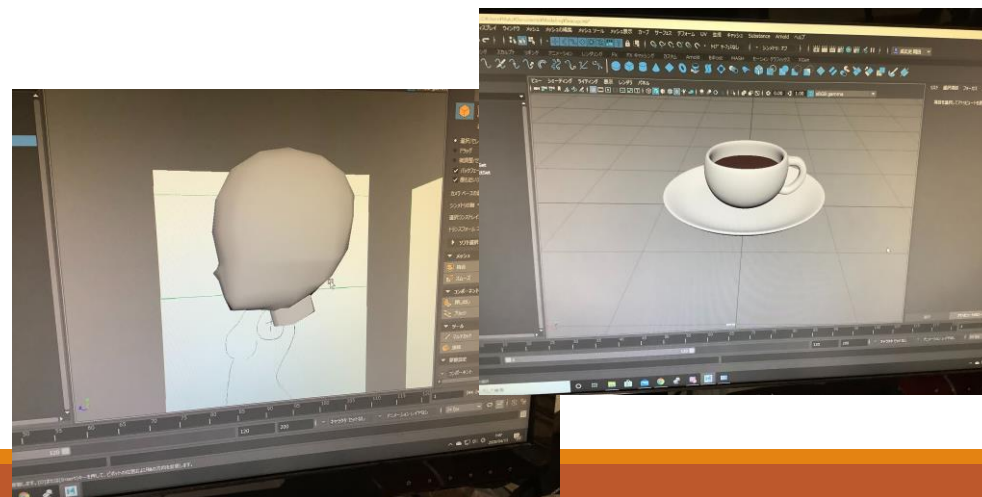
学校・学部名：京都大学工学部工業化学科

名前：岡田成広史（おかだ まさとし）

趣味：プログラミング、ギター、イラスト制作

特技：独学（プログラミングも独学で身に着けました）

好きな分野：基礎的な技術、主にグラフィックス関連の技術に興味があり、シェーダーの勉強をしたり最近ではモデリングに挑戦しています。



ゲーム概要

射的を行うTPSゲーム(プレイ動画: <https://youtu.be/orsCCMyu-8Y>)

使用API・ライブラリ

- Direct3D12
- XAudio2
- Windows.h
- ImGuiライブラリ(デバッグ用)
- Nlohmann/json(JSONパースライブラリ)

使用ツール

- Visual Studio Community 2019
- PIX on Windows(デバッグ用)



目的

DirectXを用いたフルスクラッチ開発

コンポーネント志向で動く簡易ゲームエンジンの実装

コンポーネント志向の実現

シーンオブジェクトから子オブジェクトに向かって再帰的に更新

- SceneObject::Update(float)の返り値で次に遷移するシーンを決める(Program1参照)

Object::Update(float)でそれぞれの持っているコンポーネントを更新(図1参照)

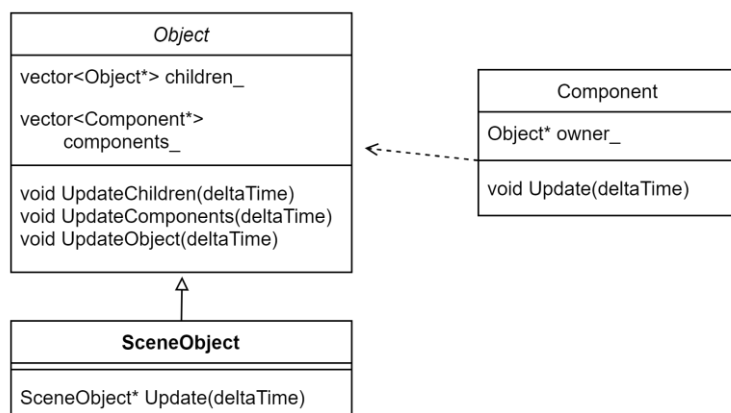


図1. オブジェクト関係のクラス図

```
nextScene_ = sceneObject_ -> Update(deltaTime);
if (sceneObject_ != nextScene_) {
    //シーンが遷移しているなら今のシーンは解放する
    delete sceneObject_;
    sceneObject_ = nextScene_;
}
```

Program1. Game. cpp内

コンポーネントの例

ColliderComponent :衝突判定用基底クラス

- AABBColliderComponent
- CapsuleColliderComponent
- OBBColliderComponent
- LineColliderComponent

MeshComponent :3Dモデル表示・アニメーション用

AudioComponent:音楽再生用コンポーネント

E.t.c.

ColliderComponent

衝突形状を囲むAABBで判定

衝突していたら形状によってオーバーライドされた関数で判定

衝突解決とObjectのOnCollision()を呼ぶ(Program2参照)

```
if (self->BroadIntersect(other->GetWorldAABB())) {  
    self->Intersect(other);  
}  
-----  
for (auto& ccomp : colliders_) {  
    auto& cis = ccomp->GetCollisionInfos();  
    for (auto ci : cis) {  
        auto a = ci.collider[0];  
        auto b = ci.collider[1];  
        a->GetOwner().UpdateOnCollision(ci);  
        ...  
    }  
}
```

Program2. PhysicsSystem.cpp内

MeshComponent

PMDファイル、または頂点配列を読み込み
DirectX12により表示

VMDファイルを読み込みフレーム補間した
モーションを再生

モーション遷移は遷移線と遷移フラグによって
管理可能

モーション間のつなぎ目の補間が可能
(Program3参照)

視錘台カリングによる高速化

```
if (isTransiting) {  
    //遷移補間中  
    ...  
    rFrameNo = -transitFrameNum; //前のモーションを補間につかう  
    rQuaternion = forMotionData_[bonemot.first].quaternion;  
    rOffset = forMotionData_[bonemot.first].offset;  
    motit = mots.begin(); //今のモーションの最初のキーフレーム  
}  
-----  
//線形補間の係数を取得  
auto t = static_cast<float>(frameNo - rFrameNo)  
/ static_cast<float>(motit->frameNo - rFrameNo);  
//ベジェ曲線状の係数を求める  
t = GetYFromXOnBezier(t, motit->p1, motit->p2, 15);  
//線形補間を行う  
...
```

Program3. MeshComponent.cpp内

AudioComponent

WAVファイルを読み込み、XAudio2により再生

フェード再生、3Dオーディオ、リバーブ、
各種パスフィルター機能の実装

XAudio2の機能を隠蔽し、AudioEventという
ハンドラクラスを通して操作できる(Program4参照)

```
class AudioComponent{  
...  
    void SetLoop(bool loop) {  
        audioEvent_. SetLoop(loop);  
    }  
...  
private:  
    AudioEvent audioEvent_;  
...  
};
```

Program4. AudioComponent.h内

Direct3D12を用いた開発

Direct3D関連のコードはDx12Wrapper,ModelLoader,Renderer内に隠蔽

ファイル名、位置、回転情報から自動的にレンダリング

DirectXMathを数学ライブラリとして使うために
XMFLOATを継承したVectorを使用(Program5参照)

- XMFLOATは演算子がなくXMVECTORとの変換も不便
- 全体でXMFLOATの継承クラスを使うことで
DirectXとの値の受け渡しが楽に！

```
//XMFLOAT3
class Vector3:public XMFLOAT3
{
public:
    Vector3() = default;
    Vector3(float x, float y, float z) {
        this->x = x; this->y = y; this->z = z;
    }
    Vector3(const XMVECTOR& xmvec) {
        XMVECTOR tmp = xmvec;
        XMStoreFloat3(this, tmp);
    }
    Vector3(const XMFLOAT3& xf) {
        this->x = xf.x; this->y = xf.y; this->z = xf.z;
    }
    ...
}
```

Program5: XMFLOAT_Helper.h内

Shader開発

シャドウマップによる影の表示(BasicPixelShade.hlsl)

スフィアマップ,ノーマルマップとAOマップに対応(BasicPixelShade.hlsl)

コントラスト調整,ビネット効果のポストエフェクト(BoardPixelShader.hlsl)

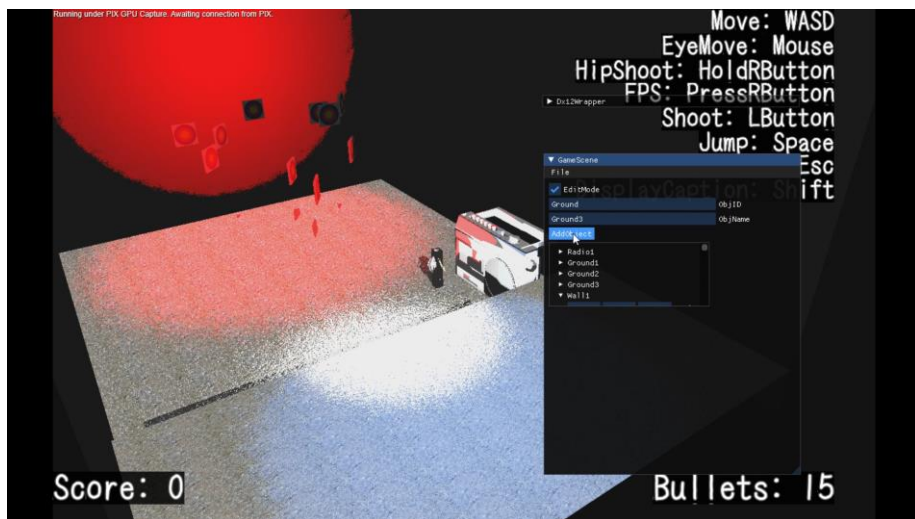
Zbufferを用いたディファードレンダリングによる複数光源のライティング計算
(ZBuffPixelShader.hlsl)

ステージエディット機能

デバッグモードのみ視覚的にステージを構築可能
(動画: <https://youtu.be/0XS1iscjbKg> 参照)

ステージ情報をJSON形式で管理(Program6参照)

- ステージセーブ、ロードが可能



画像1. デバッグビルドの画面

```
{  
  "Objects": [  
    {  
      "Name": "Radio1",  
      "ObjID": "Radio",  
      "Pos": [ 70, 0, 70 ],  
      "Rot": [ 0, 0, 0 ],  
      "Scale": [ 1000, 1000, 1000 ]  
    },  
    ...  
  ],  
}
```

Program6: JSONによるステージの管理

その他実装したもの

InputSystem.h :WinAPIから入力判定を取得

CameraComponent.h :オブジェクトをカメラとして扱う

SpriteComponent.h :スクリーン座標上に画像表示

PointLightComponent.h :オブジェクトを点光源として扱う

MoveComponent.h :オブジェクトの速度などの設定

ranking.dat :スコア上位5位の読み込み、書き込み(ResultSceneObject.cpp内)

お借りした素材

「【MMD】移動モーションv1.3」むつごろう様

<https://bowlroll.net/file/154665>

「レーザーガン.pmx」しろつき様

「モノラルラジカセ.pmx」MIN 様

「初音ミク.pmd」MMD付属モデル 作成者:あにまさ様

土・コンクリート PolyHeaven

使用フォント:「Kゴシック」作成者:ケイ様

<https://k-gothic-font.hatenablog.com/>

使用SE: 効果音ラボ 様

使用BGM: 魔王魂 様