

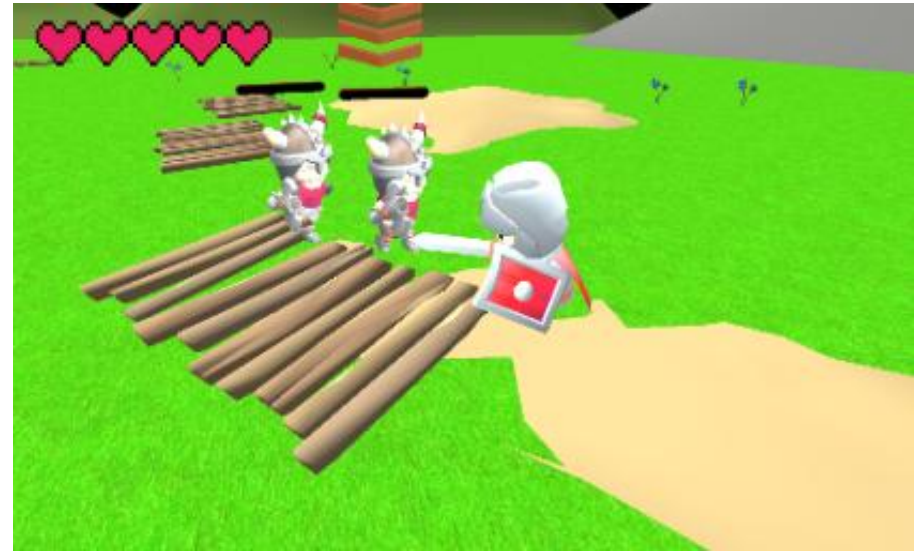


使用言語：C++言語、HLSL言語

使用ライブラリ：DXライブラリ

制作人数：1人

制作期間：2025/4～2025/7



初めて個人で3Dアクションゲームを制作しました。

戦闘のアクションを際立たせるためにロックオンやジャスト回避などを組み込みました。

カメラ制御にも注力し、ロックオン時やボス登場時の演出用のカメラ制御などを行いました。

工夫した点<ロックオンカメラ>

ロックオン中はプレイヤーと対象の
敵を画面内に入れるために、
2体の距離に応じて
カメラの注視点の座標や相対座標
の大きさを設定するようにしました。

```
//ロックオン対象と追従対象の離れている距離
VECTOR distance = VSub(rockPos_, followPos_);

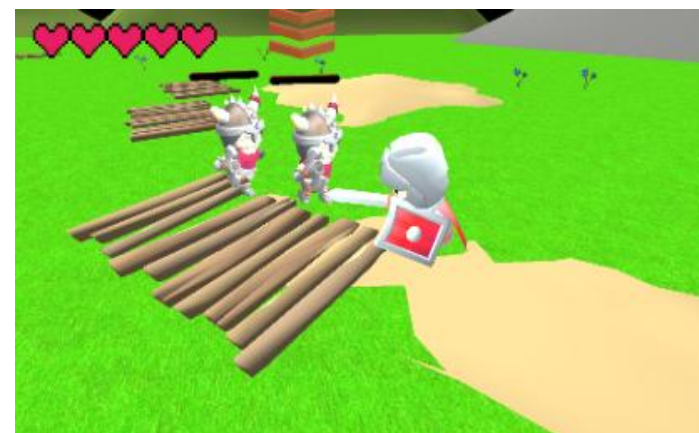
//離れる距離を数値化
float disMag = Utility::MagnitudeF(distance);
//最低限の値を下回っていたら
if (disMag <= ROCK_DISTANCE_MIN) {
    //最低限の値を入れる
    disMag = ROCK_DISTANCE_MIN;
}

//カメラ位置調整(カメラは後方位置に。Y方向は距離に応じて高さを変える。)
VECTOR relative = { 0.0f, disMag * ROCK_MAGNIFICATION_Y, -disMag };
//カメラの回転情報をもとに相対座標を回転させる
VECTOR relativeCPos = rot_.PosAxis(relative);

//注視点の更新
//ロックオン中の注視点は追従対象とロックオン対象の中間地点にある。
//focusPos_ = VAdd(followPos, VScale(distance, 0.5f));
goalFocusPos_ = VAdd(followPos, VScale(distance, 0.5f));
focusPos_ = Utility::Lerp(focusPos_, goalFocusPos_, 0.2f);

//カメラ位置の更新
auto gPos = VAdd(focusPos_, relativeCPos);
pos_ = Utility::Lerp(pos_, gPos, NO_LERP);
//pos_ = VAdd(focusPos_, relativeCPos);

//ある程度の高さは保つ
if (pos_.y < UNDERLIMIT_Y) pos_.y = UNDERLIMIT_Y;
```



工夫した点<イベントシーン>

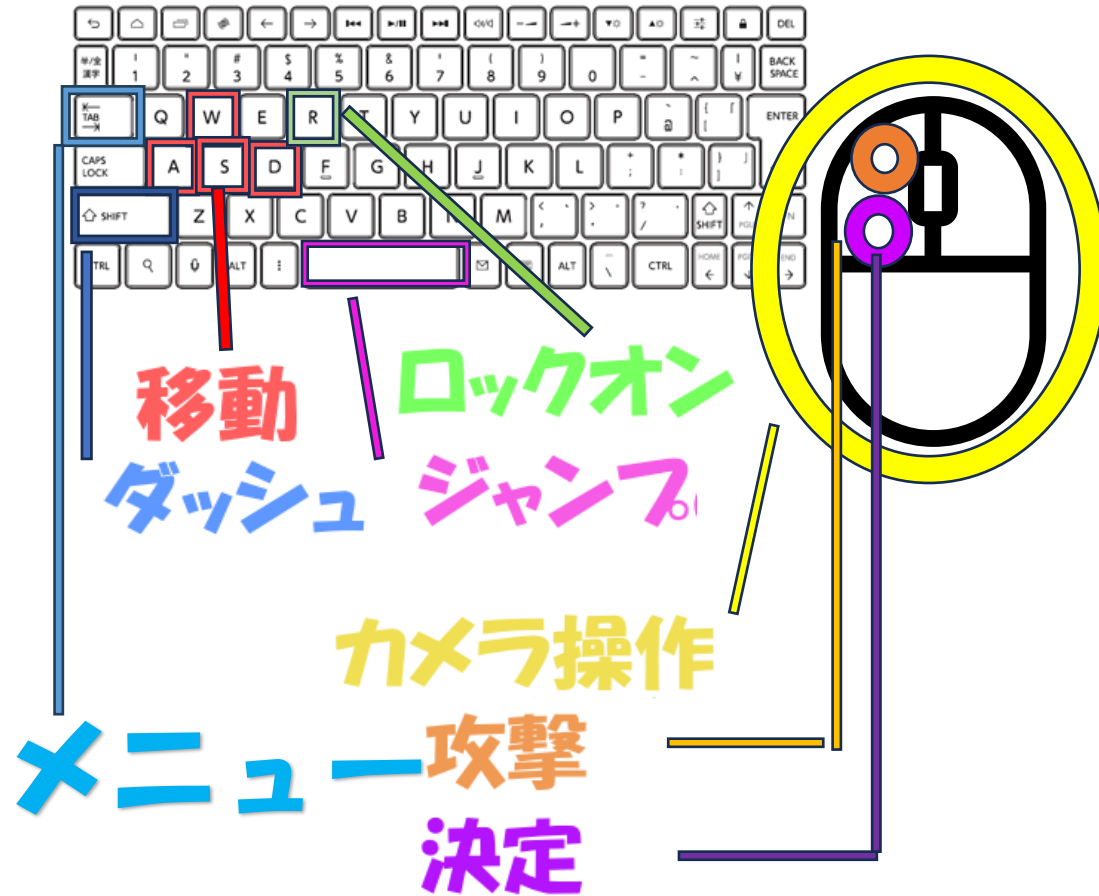
ボス登場時に「強敵が来る!」という演出を加えるために「ポストエフェクト」や「画面揺れ」「カメラワーク」などを活用しました。
関数ポインタと状態管理を活用し実行しました。

```
void DirectionUpdate(void); //演出アップデート  
bool DirectionPostEffect(void); //ポストエフェクト  
bool DirectionShakeScreen(void); //画面揺れ(流れ)  
void DoShake(void); //画面揺れ(実施)  
bool DirectionCameraMove(void); //カメラ移動
```

```
void Game::DirectionUpdate(void)  
{  
    //危険のポストエフェクト→画面揺れ→カメラ  
    if ((this->*direcUpdate_()) {  
        //次の演出に  
        direcState_ = static_cast<BOSS_DIRECTION>(static_cast<int>(direcState_) + 1);  
        direcCnt_ = 0;  
        //もし終了したら  
        if (direcState_ == BOSS_DIRECTION::END) {
```



<操作方法>



<回避行動> ロックオン+移動(左右後)+ジャンプ
敵の攻撃に合わせることでジャスト回避が可能

<プレイをする前に>

・PCのディスプレイの**拡大率を100%に**設定してください。
(ノートPCではデフォルトが125%のことが多いので注意して下さい)

・キャラクターに頂点シェーダを使用しています。
もしキャラクターが実演動画のように映らない場合、
グラフィックボードの設定をしてもらう場合があります