

## 3D Action Game 制作体験

### 1. Mapを表示させます。

(1) ソリューションエクスプローラで、CSceneGame.hをダブルクリックします。

(2) `/*1`に、次のプログラムを入力します。

```
/*1
//マップ
CMap mMap;
```

(3) ソリューションエクスプローラで、CSceneGame.cppをダブルクリックします。

(4) CSceneGame::Renderメソッドの`/*1`に次のプログラムを入力して実行します。

```
/*1
//マップの描画
mMap.Render();
```

(5) マップが表示されます。

### 2. プレイヤーを登場させます。

(1) CSceneGame.cppを開きます。

(2) CSceneGame::Initメソッドの`/*2`に次のプログラムを入力して実行します。

```
/*2
//プレイヤーの生成
new CXPlayer(&mModel, CVector3(-10.0f, 5.0f, -65.0f));
```

(3) プレイヤーは表示されますが、落ちていきます。

### 3. プレイヤーにコライダを設定し、地下へ落ちないようにします。

(1) ソリューションエクスプローラで、CXPlayer.cppをダブルクリックします。

(2) CXPlayer::Initメソッドの`/*3`に次のプログラムを入力して実行します。

```
/*3
//カプセルコライダ
mColliders.push_back(
    new CCapsuleCollider(this, &mpCombinedMatrix[1],
        CVector3(0.0f, 10.0f, 0.0f), CVector3(0.0f, 160.0f, 0.0f), 0.7f));
```

(3) プレイヤーは地上で止まります。

### 4. カメラをJキーで左へ、Lキーで右へ回転させます。

(1) CXPlayer.cppを開きます。

(2) CXPlayer::Updateメソッドの`/*4`に次のプログラムを入力して実行します。

```
/*4
//Jキーでカメラ回転
if (CKey::Push('J')) {
    mCameraRotation.y += CAMERA_ROTATION_ADD;
}
//Lキーでカメラ回転
if (CKey::Push('L')) {
    mCameraRotation.y -= CAMERA_ROTATION_ADD;
}
```

(3) JキーとLキーを押すと、プレイヤーを中心にカメラが回転します。

5. Wキーでプレイヤーを前進させます。

(1) CXPlayer.cppを開きます。

(2) CXPlayer::Updateメソッドの/\*5に次のプログラムを入力して実行します。

```
/*5
//位置を移動させる
mPosition = mPosition + CVector3(0, 0, PLAYER_VELOCITY)*rot;
//歩きモーション (1番、繰り返し、60フレーム)
ChangeAnimation(1, true, 60);
//ステータスを歩く
mState = EWALK;
```

(3) Wキーを押すと、プレイヤーが前進します。

6. Sキーでプレイヤーを後退させます。

(1) CXPlayer.cppを開きます。

(2) CXPlayer::Updateメソッドの/\*6に次のプログラムを入力して実行します。

```
/*6
//位置を移動させる
mPosition = mPosition + CVector3(0, 0, -PLAYER_VELOCITY)*rot;
//歩きモーション (1番、繰り返し、60フレーム)
ChangeAnimation(1, true, 60);
//ステータスを歩く
mState = EWALK;
```

(3) Sキーを押すと、プレイヤーが後退します。

7. Aキーで左へ、Dキーで右へプレイヤーを回転させます。

(1) CXPlayer.cppを開きます。

(2) CXPlayer::Updateメソッドの/\*7に次のプログラムを入力して実行します。

```
/*7
//Dキーを押すと右へ回転し
if (CKey::Push('D')) {
    //右へ回転
    mRotation.y -= PLAYER_ROTATION_ADD;
    mRotation.y = (int)mRotation.y % 360;
}
//Aキーを押すと左へ回転する
if (CKey::Push('A')) {
    //左へ回転
    mRotation.y += PLAYER_ROTATION_ADD;
    mRotation.y = (int)mRotation.y % 360;
}
```

(3) AキーとDキーを押すと、プレイヤーが回転します。

8. スペースキーで、プレイヤーが攻撃します。

(1) CXPlayer.cppを開きます。

(2) CXPlayer::Updateメソッドの/\*8に次のプログラムを入力して実行します。

```

    /*8
    //攻撃1モーション（3番、繰り返さない、30フレーム）にする
    ChangeAnimation(3, false, PLAYER_ATTACK_SPEED);
    mState = EATTACK;

```

- (3) スペースキーを押すと、プレイヤーが剣を振ります。

#### 9. 敵を出現させます。

- (1) CSceneGame.cppを開きます。
- (2) CSceneGame::Initメソッドの/\*9に次のプログラムを入力して実行します。

```

    /*9
    //敵の生成
    new CXEnemyKnight(&mModelKnight, CVector3(0.0f, 0.0f, -50.0f));

```

- (3) 敵が出現します。

#### 10. 敵がプレイヤーを見つけると、追いかけてくるようにします。

- (1) ソリューションエクスプローラで、CXEnemyKnight.cppをダブルクリックします。
- (2) CXEnemyKnight::Initメソッドの/\*10に次のプログラムを入力して実行します。

```

    /*10
    //索敵
    mColliders.push_back(
        new CSphereCollider(CCollider::ENONE, this, &mpCombinedMatrix[1],
            CVector3(0.0f, 0.0f, 1.0f), 20.0f));

```

- (3) プレイヤーが近くにくると、追いかけて始めます。

#### 11. 敵に、プレイヤーが攻撃範囲に入ると攻撃させます。

- (1) CXEnemyKnight.cppを開きます。
- (2) CXEnemyKnight::Updateメソッドの/\*11に次のプログラムを入力して実行します。

```

    /*11
    //攻撃アニメーション
    ChangeAnimation(7, false, ENEMY_ATTACK_SPEED);
    //ステータスを攻撃中
    mState = EATTACK;

```

- (3) プレイヤーが攻撃範囲に入ると、剣を振ります。

#### 12. 敵を増やします。

- (1) CSceneGame.cppを開きます。
- (2) CSceneGame::Initメソッドの/\*12に次のプログラムを入力して実行します。

```

    /*12
    //敵の生成
    new CXEnemyKnight(&mModelKnight, CVector3(-10.0f, 0.0f, -40.0f));

```

- (3) 敵が増えます。

#### 13. ゲームバランス調整

- (1) ソリューションエクスプローラで、Define.hをダブルクリックします。
- (2) #defineの数値を変えると、動きが変わります。