

# コンピュータアニメーション特論 レポート

## 第3回 ピッキング

学生番号: 12345678 氏名: 九工大 太郎

20xx 年 x 月 x 日

レポートの書き方の注意：(この部分は、提出レポートからは削除すること)

- 以下の様式中の「※ レポート課題」の部分を、自分が作成したプログラムに置き換える。
- 変数定義やインデントを適切に行うこと。動作しないプログラムや見にくいプログラムは、減点となる。
- 様式で指定されている箇所以外に変更を加えた場合は、どの関数を追加変更したのかが分かるように、関数定義を含めて変更内容を枠内に記述する。

### 1 ピッキングの実現

ピッキングを実現するように、以下の通り、元のプログラムの処理の一部に変更を加えた。

#### 1.1 スクリーン座標系でのピッキング

##### 1.1.1 ワールド座標系からスクリーン座標系への投影変換

```
void UpdateObjectProjection ()
{
    // 省略

    for ( int i=0; i<num_objects; i++ )
    {
        // i番目のオブジェクトの情報を取得
        ObjectInfo * obj = &objects[ i ];

        // ※レポート課題 (ここに自分が作成したプログラムを記述する)
    }
}
```

##### 1.1.2 マウス位置に最も近いオブジェクトの探索

```
int PickObjectScreen( int mouse_x, int mouse_y )
{
    // 省略

    // 全オブジェクトの画面上の位置を更新
    UpdateObjectProjection ();

    // ※レポート課題 (ここに自分が作成したプログラムを記述する)
}
```

## 1.2 ワールド座標系でのピッキング

### 1.2.1 スクリーン座標系からワールド座標系への逆投影変換

```
int PickObjectWorld( int mouse_x, int mouse_y )
{
    // 省略

    // マウス位置に対応する3次元空間の直線を求める
    double wx, wy, wz, dx, dy, dz;

    // ※レポート課題（ここに自分が作成したプログラムを記述する）

    // 省略
}
```

### 1.2.2 三角面と半直線の交差判定

```
bool CheckCross( const Point3f & tri_p0, const Point3f & tri_p1, const Point3f &
    tri_p2, const Point3f & seg_org, const Vector3f & seg_vec, Point3f & cross_point )
{
    // ※レポート課題（ここに自分が作成したプログラムを記述する）
}
```