

プログラミング基礎 #10

入力と条件分岐2

担当：向井 智彦

本日の内容

- 目標：論理演算と条件分岐について理解を深める
- 講義内容：
 - 「論理演算」「条件分岐」のおさらい
 - インタラクションのための条件分岐
- 演習：インタラクティブCGプログラミング

第4回のまとめ

- bool型
 - true と false のいずれかのみ
- 比較演算
 - 数値の等しさ・大小関係をbool型で評価
- 論理演算
 - bool型変数に対する論理和・論理積
- 条件分岐
 - bool値に応じて異なる処理を行う仕組み

ブール(**bool**)型

- 真理値 とか 真偽値, ブーリアン型とも
 - 真: **true** あるいは 偽: **false**
 - true の否定 は false, false の否定 は true
 - 数学「命題・論理」と関係

```
bool a, b;  
a = true;  
b = !a; // 否定演算  
c = a * 2 + b * 2; // あり得ない演算  
cout << a << ", " << b << ", " << c << endl;
```

比較演算：演算結果はbool型

※以下, a と b は任意の型 (int, char, double, bool, ...)

- 一致「 $a == b$ 」(代入「 $=$ 」と混同しないよう注意)
 - a と b が同じ値であれば true, 違うなら false
- 非一致「 $a != b$ 」
 - a と b が異なる値であれば true, 違うなら false
- 大なり「 $a > b$ 」, 以上「 $a >= b$ 」
 - a が b より大きい値なら true, 小さいなら false
- 未満「 $a < b$ 」, 以下「 $a <= b$ 」
 - a が b より小さい値なら true, 大きいなら false

真偽値の論理演算

※ a と b はどちらもbool型

- 否定「 !a 」 「 !b 」
 - true の否定は false, false の否定は true
- 論理和「 a || b 」
 - a と b のいずれか true なら, 結果は true
 - a と b の両方が false の時, 結果は false
- 論理積「 a && b 」
 - a と b の両方が true なら, 結果は true
 - a と b のどちらかが false なら, 結果は false

条件分岐 if & else

- 真偽値true/falseに応じて異なる処理を行う、
処理分岐のための制御構文

```
int main() {  
    int 条件式: 真偽値: 比較演算 & 論理演算  
    if (x > y) { 条件式がtrueの時に実行される文  
        cout << x << " is larger than " << y << endl;  
    }  
    else { 条件式がfalseのときに実行される文  
        cout << x << " is smaller than " << y << endl;  
    }  
}
```

インタラクティブグラフィックス

- 「ボタンを押したら...」
- 「画面をタッチしたら...」
- 「カーソルが動いたら...」
- 「話しかけたら...」
- 「動いたら...」
- 「時間が経ったら...」



- 形が変わる
- 色が変わる
- 物が現れる／消える
- 回転する
- 移動する
- 大きさが変わる

すべて if 文を用いて実現

「もし、キーボードのQキーが 押されていたら」

```
if (key == 'q' || key == 'Q')  
{  
    exit(0);  
}
```

- キーの一致判定のための比較演算 ==
- 大文字Q・小文字q関係なく処理するための論理和演算 || （qあるいはQが押された？）
- 一致した場合の条件分岐 if

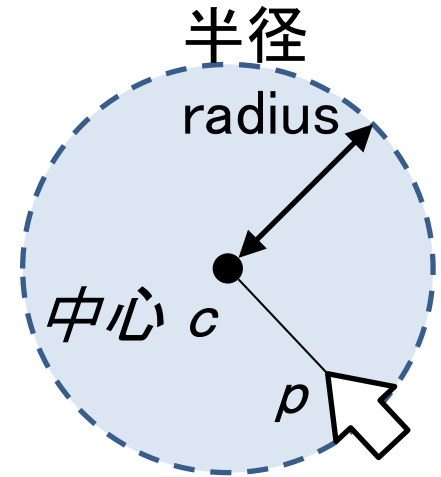
「もし、画面中央より右上に マウスカーソルがあったら...」

```
if (mousePosX > 0 && mousePos > 0)
{
    . . . . .
}
```

- マウス位置判定のための比較演算 >
- 右上⇒右半分かつ上半分という 2条件の同時成立を判定するための論理積 &&
- 条件成立時の処理分岐を行う if

「もし、球体とマウスカーソルが接触していたら...」

- 球体の中心位置とマウス位置の距離によって判定
 - 球に接触していたら
 - 球の内部にあったら
 - 球の中心からの距離が、球の半径よりも小さかったら



- 距離... $\sqrt{(p_x - c_x)^2 + (p_y - c_y)^2}$

「もし、球体とマウスカーソルが 接触していたら...」

```
double distX = mousePosX - centerX;  
double distY = mousePosY - centerY;  
double distance = sqrt(distX * distX  
                        + distY * distY);  
if (distance < radius)  
{  
    .....  
}
```

演習課題BASIC

- マウス位置に応じて表示する物体を切り替えるシステムを実現
 - X, Y とともにプラスの場合は球
 - Xがプラス、Yがマイナスの場合はティーポット
 - Xがマイナス、Yがプラスの場合はドーナツ
 - X, Y とともにマイナスの場合は何も表示しない

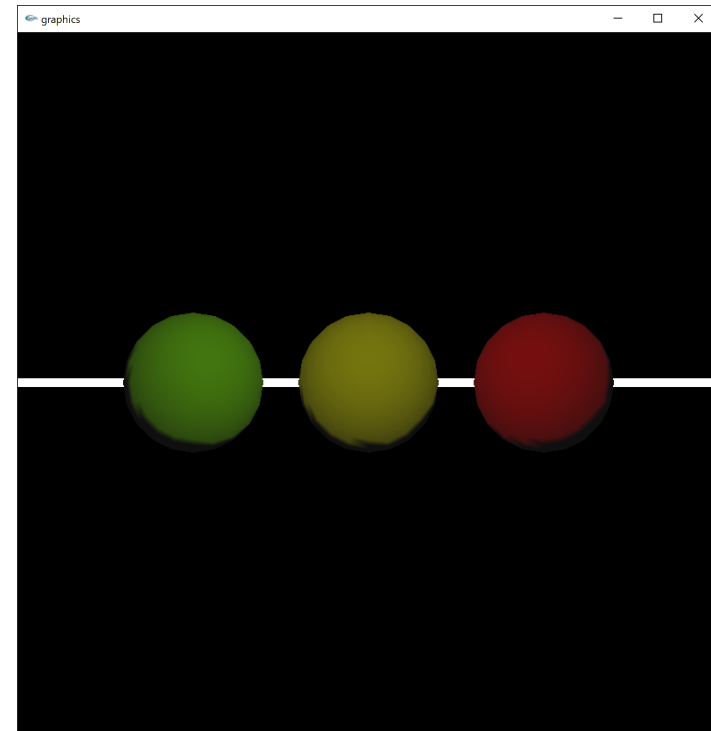
ドーナツ (Torus)	球
表示 なし	Teapot

演習課題BASIC

```
if (右上か?) {  
    glutSolidSphere(50.0, 40, 40);  
}  
else if (左上か?) {  
    glutSolidTorus(20.0, 40.0, 40, 40);  
}  
else if (右下か?) {  
    glutSolidTeapot(50.0);  
}  
// else {    // 左下の場合は何もしないので省略  
// }
```

演習課題EXTRA

- 3つの球体を横に並べる
- 左の球体から順に、暗めの緑色、暗めの黄色、暗めの赤色にする
- マウスカーソルが重なった球体の色のみ明るくする



EXTRAの補足（緑球体の描画）

```
glPushMatrix();  
glTranslated(-50, 0, 0);  
float sphere1Color[4] = { 0.1, 0.2, 0.0, 1.0 };  
glMaterialfv(GL_FRONT_AND_BACK, GL_DIFFUSE, sphere1Color);  
glMaterialfv(GL_FRONT_AND_BACK, GL_SPECULAR, sphere1Color);  
glutSolidSphere(20.0, 20, 20);  
glPopMatrix();
```

左側へ移動

暗めの黄緑

追加した行

演習課題ADVANCE

- マウス位置とキーボード入力を同時活用するインタラクティブティーポットを制作
 - お題・インタラクション方法・実装方法、いずれもすべて自由
 - あまり凝り過ぎないように
 - 良いネタは授業の最終課題のためにとっておく