# Object: Code Snippets

## OrxFrame

## OrxFX

## OrxFXPointer

## OrxObject

### OrxObject_SetPosition

```
orxVECTOR tilePos;
orxVector_Set(&tilePos, orx2F(80.0f) * x, orx2F(160.0f), orxFLOAT_0);
orxObject_SetPosition(tile, &tilePos);
```

```
orxVECTOR pos;
orxObject_GetPosition(player, &pos);
pos.fX = -pos.fX;
orxObject_SetPosition(player, &pos);
```

### orxObject_SetRotation

Rotation is set in radians. Zero rad vector is equivalent to (1, 0) vector in screen coordinates. In other words it is a horizontal line pointing from left to right.

Positive rotation is set in clockwise direction. If vector origin was in the center of the screen, then 1 rad would point to the left right corner of the screen.

By default object starts with zero rotation angle. Rotation counterclockwise will put the angle into negative range. Rotation clockwise will put the angle into positive range of values.

### orxObject_SetAngularVelocity

Set angular velocity changes object rotation value over time. Setting positive value make object rotate clockwise. Negative value sets counterclockwise direction.

TBD: What's the unit of measure?

### orxObject_GetWorldRotation and orxObject_GetRotation

As object rotates continuously clockwise its angle will continue to increase. Thus after a first full circle its angle will be set to 2*Pi. After the second full circle it will be set to 4*Pi. The same is true for confer-clockwise rotation. The 1st full circle will be set to -2*Pi and so on.

Thus you may have to normalize the rotation value of the object.

### orxObject_CreateNeighborList and orxObject_DeleteNeighborList

Use it to obtain objects within the specified bounding box.

```
    orxOBJECt *obj; // comes from mouse click event or in some other way
    orxVECTOR pos, size;
    orxFLOAT range = 250.;
    orxOBOX box;

    orxObject_GetWorldPosition(obj, &pos);
    orxVector_Set(&size, range, range, 0.);
    orxOBox_2DSet(&box, &pos, &orxVECTOR_0, &size, 0.);

    orxBANK *neighbors = orxObject_CreateNeighborList(box);
    void* cell = orxNULL;
    while ((cell = orxBank_GetNext(neighbors, cell))) {
        orxOBJECT **n = cell;
        orxLOG("object name: %s.", orxObject_GetName(*n));
    }
    orxObject_DeleteNeighborList(neighbors);
```

### object traversing

Object Traversing has its own page due to somewhat large discussion on the topic.

## OrxSpawner

## OrxStructure

## OrxTimeLine