

KAIST

# CS 482 Lab Session

## Interactive Computer Graphics

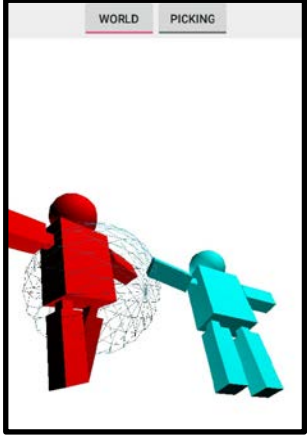
2016.10.26

VISUAL  
COMPUTING Lab

KAIST

## Goals

- Color picking with Scene graph
  - Implement user manipulation code



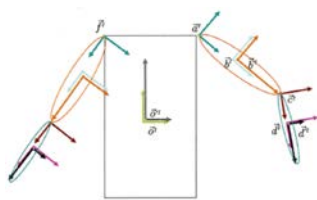
2

VISUAL  
COMPUTING Lab

## Preliminaries

KAIST

- You should understand
  - How to work with a SceneGraph
  - How to make a color-coded scene
  - How to draw scene on the FrameBuffer in OpenGL
  - How to pick color on the FrameBuffer



3

VISUAL  
COMPUTING Lab

## Pick OBJ with Color

KAIST

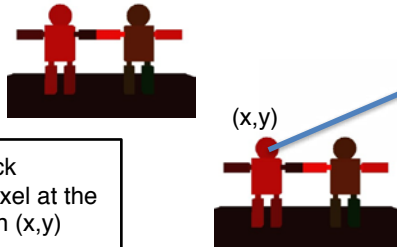
- Color mapping with ID
  - Make some color encoding method utilizing the ID of object: colorTold()
  - Store the color-ID pair
- Draw color-coded scene
  - Draw each object with coded color generated from object's ID
  - Color-coded scene is drawn on the FrameBuffer
- Pick object from the scene
  - Get the color on the position (x,y)
  - Find the object which is drawn with the picked color.

4

VISUAL  
COMPUTING Lab

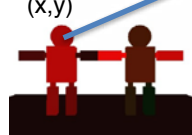
## Pick OBJ with Color

Draw  
color-coded scene  
on the FrameBuffer



RGB: (24,2,2)

Pick  
Color of pixel at the  
position (x,y)




Pick  
Object from the  
picked color


(24,2,2) → id: 4 → Head\_Joint Node

IMPORTANT:

Draw  
The scene which is  
show to user



5



## Scene graph

- Please remind the structure of SceneGraph

$$\vec{o}' = \vec{w}'O$$

$$\vec{o}'' = \vec{o}'O'$$

$$\vec{a}' = \vec{o}'A$$

$$\vec{b}' = \vec{a}'B$$

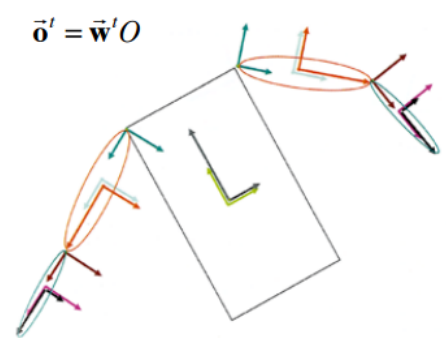
$$\vec{b}'' = \vec{b}'B'$$

$$\vec{c}' = \vec{b}'C$$


$$\vec{d}' = \vec{c}'D$$

$$\vec{d}'' = \vec{d}'D'$$


$$\vec{f}' = \vec{o}'F$$



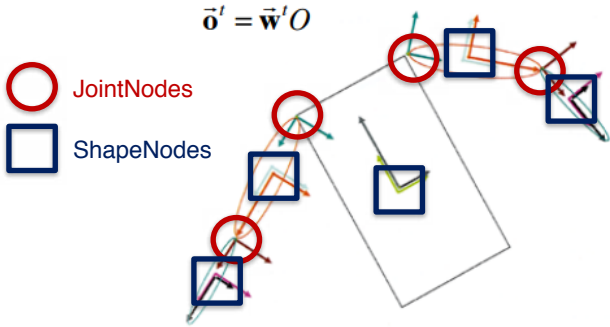
6



# Scene graph



- Also, you should think about the different role of “JointNode” and “ShapeNode”



○ JointNodes  
□ ShapeNodes

$$\vec{o}' = \vec{w}'O$$

$$\vec{o}'' = \vec{o}'O'$$

$$\vec{a}' = \vec{o}'A$$

$$\vec{b}' = \vec{a}'B$$


$$\vec{b}'' = \vec{b}'B'$$

$$\vec{c}' = \vec{b}'C$$


$$\vec{d}' = \vec{c}'D$$

$$\vec{d}'' = \vec{d}'D'$$

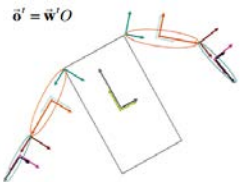
$$\vec{f}' = \vec{o}'F$$

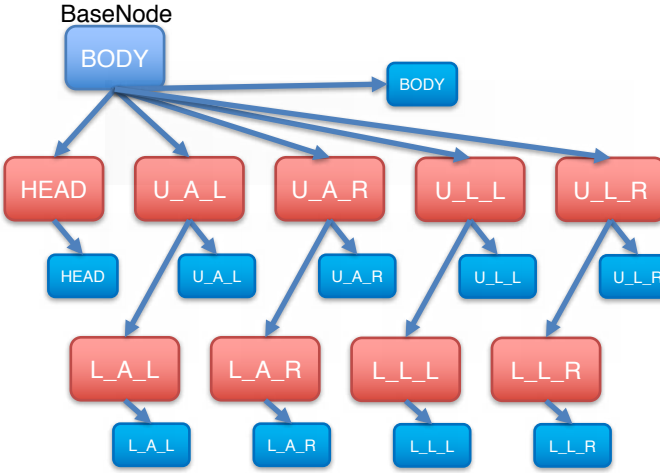



# Scene graph



- Structure of MyRobot



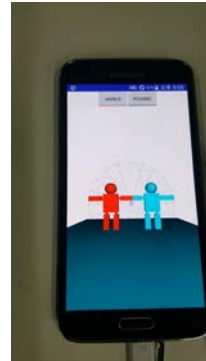




## Practice: result

KAIST

- Implement Color picker
  - Fill out the **Picker** class
  - Fill out the **MyGLRenderer** class
  - Replace the **blue robot** to the your own object (which is implemented at previous lab session)
- Motion events are already implemented
- World mode: is rendered with DEFAULT\_SHADER
- Picking mode: is rendered with PICKING\_SHADER (Color-coded scene)



9

VISUAL COMPUTING Lab

## Practice: Task 1

KAIST

- Fill out the **Picker** class on the skeleton code

```

private void addToMap(int id, SgRbtNode node) {
    //Task: add the pair of "id" and "node" into "idToRbtNodeMap".
}

private SgRbtNode find(int id) {
    //Task: Find the JointNode utilizing the id from the idToRbtNodeMap
    return null;
}

@Override
public boolean visit(SgShapeNode node) {
    boolean result = true;
    idCounter++;
    for(int i = nodeStack.size() - 1; i >= 0; --i) {
        SgRbtNode asRbtNode = (SgRbtNode)nodeStack.get(i);
        if(asRbtNode != null) {
            addToMap(idCounter, asRbtNode);
            break;
        }
    }
    float[] idColor = idToColor(idCounter);
    //Task: Draw the color-coded scene utilizing the generated idColor and drawer
    return result;
}

public SgRbtNode getRbtNode(int id) {
    //Task: Read the color at position (x,y) from the FrameBuffer is OpenGL.
    //After that, utilizing the color, find and return the JointNode which rendered at position (x,y)
    ByteBuffer query_color;
    int id = -1;
    return find(id);
}

```

idToColor & colorToId methods are already implemented, but you should understand how to work those methods

The number is represented recommended fill-out order

10

VISUAL COMPUTING Lab

# Practice: Task 2



- Fill out the **MyGLRenderer** class on the skeleton code

```

public void drawStuff(boolean picking){
    float RigForm eyeRbt = RbtAccuVisitor.getPartEyeRbt(q_world, q_currentCameraNode);
    float RigForm invEyeRbt = RigForm.inv(eyeRbt);

    if(!picking) {
        float mat11 = mat11b1111; mat11b11; mat11b11; mat11b11;
        ...
    }
    else {
        int x = (int)(x*viewportWidth / touchedW);
        //Task: utilizing PickingShader, pick object utilizing the Picker.
        //
        //Note: you should set the picked object to "g_currentPickedNode"
        //if you do that, picking will work.
        //Hint: you should use following: Picker class, q_world, q_currentPickedNode, glFlush method
        //
        ////////////////////////////////////////////////////
        if (q_currentPickedNode != q_currentNode) {
            q_currentPickedNode = null; // set to null
            mArcBall.setScale(1f);
            ...
        }
    }
}

public void picking(float x, float y){
    touchedW = x;
    touchedH = y;
    pickingNode = true;

    //Task: draw solid colored scene utilizing the PickingShader which is declared on initShader() method
    //Clear color, drawStuff with PickingShader, set back the clear color
}
    
```

11



# Practice: Task3



- **MyGLRenderer**: Replace the “g\_robotNode2” to the your own object which is implemented at previous lab session

```

public void initState(){
    q_world = new SgBotNode();
    q_world.setName("world");

    s_skyNode = new SgBotNode(new RigForm(defaultCameraPos));
    s_skyNode.setName("sky");

    s_groundNode = new SgBotNode();
    //utilize SgCamera/SceneNode(Sphere q_geometry q_float[] color float[] translate float[] multByMag float[] scale)
    SgSphere/SphereNode and = new SgSphere/SphereNode(new Sphere(), new float[]{0.3f, 0.9f, 0.9f}, new float[]{0, -0.0f}, new float[]{90, 0.0f}, new float[]{6.6f, 0});
    and.setName("sun");

    s_groundNode.addChild(sun);
    s_groundNode.setName("ground");

    s_robotNode = robotFactory.constructRobot(new float[]{10, 0, 0});
    s_robotNode.setName("robot");
    s_robotNode2 = robotFactory.constructRobot(new float[]{10, 11, 11});
    s_robotNode2.setName("robot2");
    s_world.addChild(s_skyNode);
    s_world.addChild(s_groundNode);
    s_world.addChild(s_robotNode);
    s_world.addChild(s_robotNode2);

    q_currentCameraNode = s_skyNode;
}
    
```

12

If your object do not work as a MyRobot object, you should modify the center position of nodes on your own object.

Also you should refer the pages about SceneGraph structure.



## Practice: Ref

KAIST

- Please check the **MyGLSurfaceView**
  - queueEvent
    - queueEvent is provided class in openGL ES
    - It is run on the rendering thread.
    - it is need for Event handling with your renderer

```
//queueEvent is run on the rendering thread.  
//it is need for communicate with your renderer  
queueEvent(new Runnable() {  
    @Override  
    public void run() {  
        mRenderer.picking(mPreviousX,mPreviousY);  
    }  
});
```

13

 VISUAL  
COMPUTING Lab