



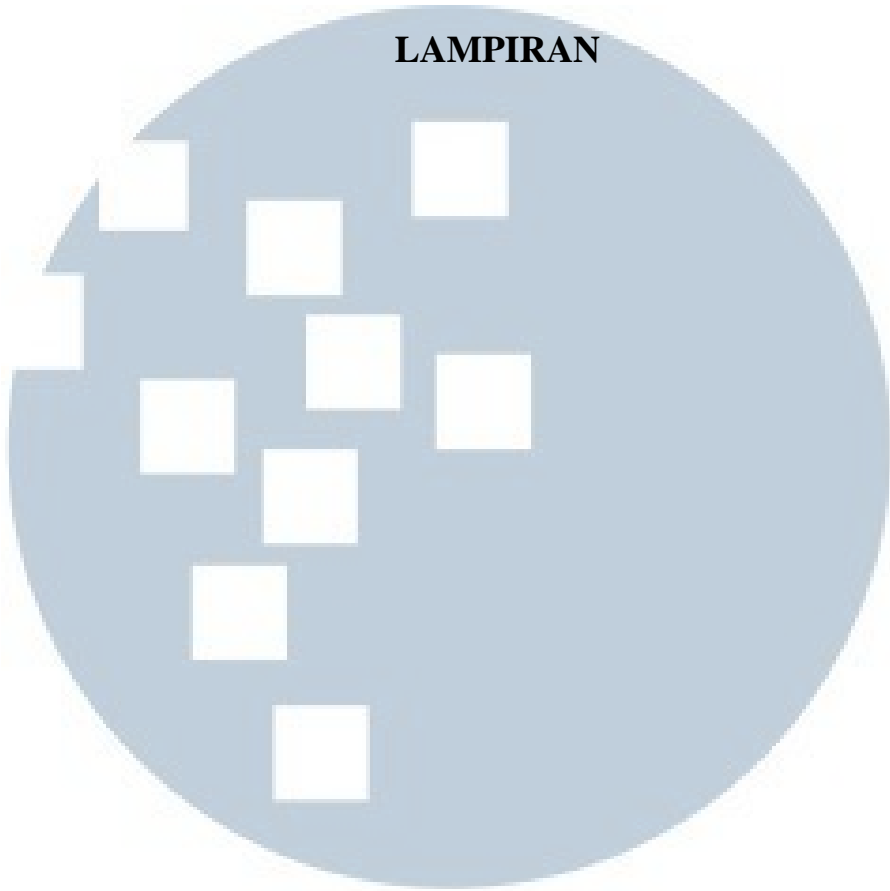
Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

LAMPIRAN



UMMN

**UNIVERSITAS
MULTIMEDIA
NUSANTARA**

```

package {
    //import stuff
    import flash.events.Event;
    import com.greensock.*;
    import org.papervision3d.lights.PointLight3D;
    import org.papervision3d.materials.shadematerials.FlatShadeMaterial;
    import org.papervision3d.materials.utils.MaterialsList;
    import org.papervision3d.objects.DisplayObject3D;
    import org.papervision3d.objects.primitives.Cube;
    import org.papervision3d.materials.BitmapFileMaterial;
    import org.papervision3d.objects.primitives.*;
    import org.papervision3d.materials.ColorMaterial;
    import flash.geom.ColorTransform;
    import flash.filters.*;
    import flash.media.SoundMixer;
    import flash.media.SoundChannel;
    import org.papervision3d.objects.parsers.Collada;
    import org.papervision3d.objects.parsers.DAE;
    import org.papervision3d.events.FileLoadEvent;

    import com.squidder.flar.FLARMarkerObj;
    import com.squidder.flar.PVFLARBaseApplication;
    import com.squidder.flar.events.FLARDetectorEvent;
    public class MultiFLARstorybook extends PVFLARBaseApplication {
        // variables that work throughout the code
        private var _cubes : Array;
        private var _lightPoint : PointLight3D;
        private var _green:Cube;
        private var cowSkin: BitmapFileMaterial;
        private var warung: MaterialsList;
        //private var drama:dramatic = new dramatic();
        //private var dramaChnl:SoundChannel = new SoundChannel();
        private var wp: DAE;
        private var piko: DAE;
        private var ikan: DAE;
        private var plastik: DAE;
        private var penyusulasi:DAE;
        private var camar: DAE;
        private var MODEL_SCALE      : Number = 30;
        private var MODEL_SCALE2     : Number = 45;
        //private var MODEL_SCALE3    : Number = 50;
        private var MODEL_SCALE4     : Number = 70;
        private var MODEL_SCALE5     : Number = 35;
        //private var model:DisplayObject3D;
        //private var moo:mooSnd = new mooSnd();
        //private var mooChnl:SoundChannel = new SoundChannel();
    }
}

```

```

public function MultiFLARstorybook() {
    _cubes = new Array();

    // import the marker pattern
    _markers = new Array();
    _markers.push( new FLARMarkerObj( "assets/flar/wp.pat" , 16 , 50 , 80 )
);
    _markers.push( new FLARMarkerObj( "assets/flar/piko.pat" , 16 , 50 , 80
));
    _markers.push( new FLARMarkerObj( "assets/flar/ikan.pat" , 16 , 50 , 80
));
    _markers.push( new FLARMarkerObj( "assets/flar/plastik.pat" , 16 , 50 , 80 )
);
    _markers.push( new FLARMarkerObj( "assets/flar/penyu.pat" , 16 , 50 , 80 )
);
    _markers.push( new FLARMarkerObj( "assets/flar/camar.pat" , 16 , 50 , 80 )
);

    super( );
}

override protected function _init( event : Event ) : void {

    super._init( event );

    _lightPoint = new PointLight3D( );
    _lightPoint.y = 1000;
    _lightPoint.z = -1000;

}
//detecting the marker
override protected function _detectMarkers() : void {

_resultsArray = _flarDetector.updateMarkerPosition( _flarRaster , 80 , .5
);

    for ( var i : int = 0 ; i < _resultsArray.length ; i ++ ) {

        var subResults : Array = _resultsArray[ i ];
        for ( var j : * in subResults ) {
            _flarDetector.getTransmationMatrix( subResults[ j ] , _resultMat
);
            if ( _cubes[ i ][ j ] != null ) transformMatrix( _cubes[ i ][ j ] , _resultMat );
        }
    }
}

```

```

    }
}

override protected function _handleMarkerAdded( event : FLARDetectorEvent ) : void {
    _addCube( event.codeId , event.codeIndex );
}

override protected function _handleMarkerRemove( event : FLARDetectorEvent ) : void {
    _removeCube( event.codeId , event.codeIndex );
}

//adding your objects

private function _addCube( id:int , index:int ) : void {
//make the cow moo
//if(id==1){
//mooChnl = moo.play(0, 1);
//}

if ( _cubes[ id ] == null ) _cubes[ id ] = new Array();

if ( _cubes[ id ][ index ] == null ) {

var dispObj : DisplayObject3D = new DisplayObject3D();

//the "green screen" effect, masks out the colour green
this.viewport.filters = [
new ColorMatrixFilter([
1, 0, 0, 0, 0,
0, 1, 0, 0, 0,
0, 0, 1, 0, 0,
1, -1, 1, 1, 0
])
];

if(id==1){

```

```

wp = new DAE(true,null,true);
wp.load("assets/dae/wp.DAE");
wp.scaleX = wp.scaleY = wp.scaleZ = MODEL_SCALE;
//model.z = 5;
//Moves Model 'Up' a Line Perpendicular to Marker
wp.rotationX = 0; //Rotates Model Around 2D X-Axis of Marker
wp.rotationY = 180; //Rotates Model Around 2D Y-Axis of Marker
wp.rotationZ = 0;
wp.moveDown(80);

dispObj.addChild(wp);
}
else if(id==0){

//Create the new Collada Object with materialList
piko = new DAE(true,null,true);
piko.load("assets/dae/piko.DAE");
piko.scaleX = piko.scaleY = piko.scaleZ = MODEL_SCALE;
//model.z = 5;
//Moves Model 'Up' a Line Perpendicular to Marker
piko.rotationX = 0; //Rotates Model Around 2D X-Axis of
Marker
piko.rotationY = 90; //Rotates Model Around 2D Y-Axis of
Marker
piko.rotationZ = 0;
piko.moveDown(50);

dispObj.addChild(piko);
}
else if(id==2){

//Create the new Collada Object with materialList
ikan = new DAE(true,null,true);
ikan.load("assets/dae/ikan.DAE");
ikan.scaleX = ikan.scaleY = ikan.scaleZ = MODEL_SCALE2;
//model.z = 5;
//Moves Model 'Up' a Line Perpendicular to Marker
ikan.rotationX = 180;
//Rotates Model Around 2D X-Axis of Marker
ikan.rotationY = 0;

```

```

//Rotates Model Around 2D Y-Axis of Marker
ikan.rotationZ = 90;
ikan.moveLeft(20);

dispObj.addChild(ikan);
}

else if(id==3){

//Create the new Collada Object with materialList
plastik = new DAE(true,null,true);
plastik.load("assets/dae/plastik.DAE");
plastik.scaleX = plastik.scaleY = plastik.scaleZ = MODEL_SCALE4;
//model.z = 5;
//Moves Model 'Up' a Line Perpendicular to Marker
plastik.rotationX = 180;
//Rotates Model Around 2D X-Axis of Marker
plastik.rotationY = 0;
//Rotates Model Around 2D Y-Axis of Marker
plastik.rotationZ = 90;
//plastik.moveUp(80);

dispObj.addChild(plastik);

}

else if(id==4){

//Create the new Collada Object with materialList

penyusolusi = new DAE(true,null,true);
penyusolusi.load("assets/dae/penyusolusi.DAE");
penyusolusi.scaleX = penyusolusi.scaleY = penyusolusi.scaleZ =
MODEL_SCALE4;
//model.z = 5;
//Moves Model 'Up' a Line Perpendicular to Marker
penyusolusi.rotationX = 180;
//Rotates Model Around 2D X-Axis of Marker
penyusolusi.rotationY = 0;
//Rotates Model Around 2D Y-Axis of Marker
penyusolusi.rotationZ = 180;
//plastik.moveUp(80);

dispObj.addChild(penyusolusi);
}

```

```

    }
else if(id==5){
    camar = new DAE(true,null,true);
    camar.load("assets/dae/camar.DAE");
    camar.scaleX = camar.scaleY = camar.scaleZ = MODEL_SCALE5;
    //model.z = 5;
    //Moves Model 'Up' a Line Perpendicular to Marker
    camar.rotationX = 180;
    //Rotates Model Around 2D X-Axis of Marker
    camar.rotationY = 0;
    //Rotates Model Around 2D Y-Axis of Marker
    camar.rotationZ = 90;
    camar.moveUp(-100);

    dispObj.addChild(camar);
}
_baseNode.addChild( dispObj );

_cubes[ id ][ index ] = dispObj;

```

```

}
_baseNode.addChild( _cubes[ id ][ index ] );
}
//private function modelLoaded(e:FileLoadEvent):void
//{
//}
//The remove cube function. Gets activated once a marker is removed
private function _removeCube( id:int , index:int ) : void {

    if ( _cubes[ id ] == null ) _cubes[ id ] = new Array();
    if ( _cubes[ id ][ index ] != null ) {
        _baseNode.removeChild( _cubes[ id ][ index ] );
    }
}

```



```
}  
}  
//Script ini diadaptasi dari  
(http://active.tutsplus.com/tutorials/3d/beginner%E2%80%99s-guide-to-augmented-reality-part-2/)  
//oleh Jhonathan Reid untuk tutorial "Augmented Reality for Beginer part 2" di website  
Tut+  
//Diadaptasi oleh Rendy Kurniawan & Agatha Maisie, Mei- Juni 2011
```



UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA