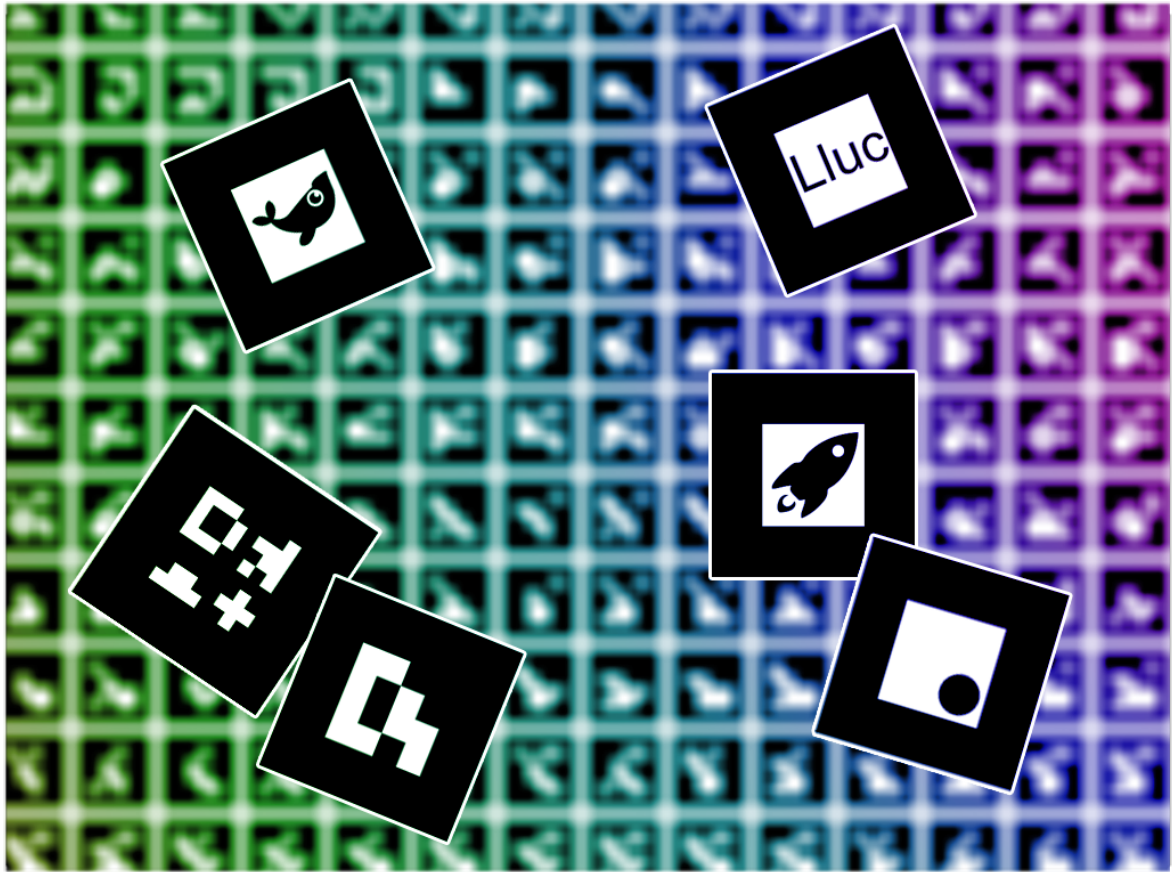


Annex 4: Material de l'aplicació

Del treball “La realitat augmentada”



Lluc Bové Canals

Tutor: Eudald Díaz Duran • Escola Escorial de Vic • 2n de batxillerat



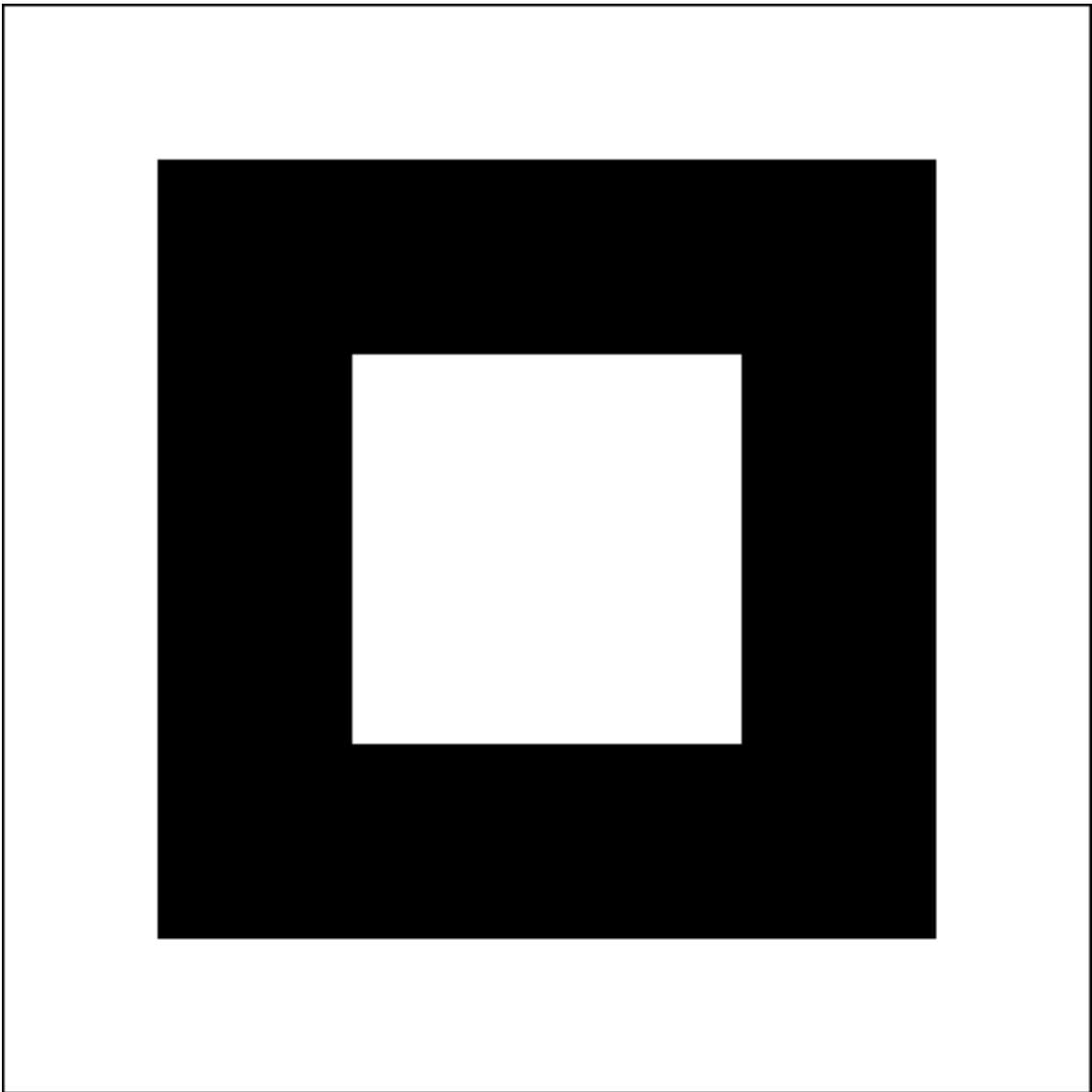
1 Disseny	4
1.1 marcadors	4
<i>1.1.1 Patró de marcadors</i>	<i>4</i>
<i>1.1.2 Marcador “coet”</i>	<i>5</i>
<i>1.1.3 Marcador “vaca”</i>	<i>5</i>
<i>1.1.4 Marcador “cub 1”</i>	<i>6</i>
<i>1.1.5 Marcador “cub 2”</i>	<i>6</i>
<i>1.1.6 Marcador “Escorial”</i>	<i>7</i>
<i>1.1.7 Marcador “Lluc”</i>	<i>7</i>
<i>1.1.8 Marcador “Helicòpter”</i>	<i>8</i>
<i>1.1.9 Marcador “orca”</i>	<i>8</i>
1.2 Models tridimensionals	9
<i>1.2.1 model “escorial”</i>	<i>9</i>
<i>1.2.2 Model “helicòpter”</i>	<i>9</i>
<i>1.2.3 Model “Lluc”</i>	<i>10</i>
<i>1.2.4 Model “orca”</i>	<i>10</i>
1.3 Interfície	11
<i>1.3.1 Mapa HTML</i>	<i>11</i>
<i>1.3.2 Fons de pantalla</i>	<i>12</i>
2. Programació	12
2.1 Codi múltiple marcador	12
<i>2.1.1 Importació de biblioteques i altres elements</i>	<i>12</i>
<i>2.1.3 Importació de marcadors</i>	<i>13</i>
<i>2.1.4 Detecció de marcadors</i>	<i>13</i>
<i>2.1.5 Configurar els models tridimensionals en general</i>	<i>14</i>
<i>2.1.6 Configurar els materials</i>	<i>14</i>
<i>2.1.7 Processament del model “Terra”</i>	<i>15</i>

2.1.8 Processament del model “vaca”	18
2.1.9 Processament dels cubs	18
2.1.10 Codi font total	18
2.2 Codi d’un sol marcador	25
2.2.1 Importació de biblioteques i altres elements	25
2.2.2 Definició de variables privades	25
2.2.3 Funció reconeixement del marcador	26
2.2.4 Funció que afegeix el model tridimensional	26
2.2.5 Funció gir	26
2.2.6 Codi font total	26

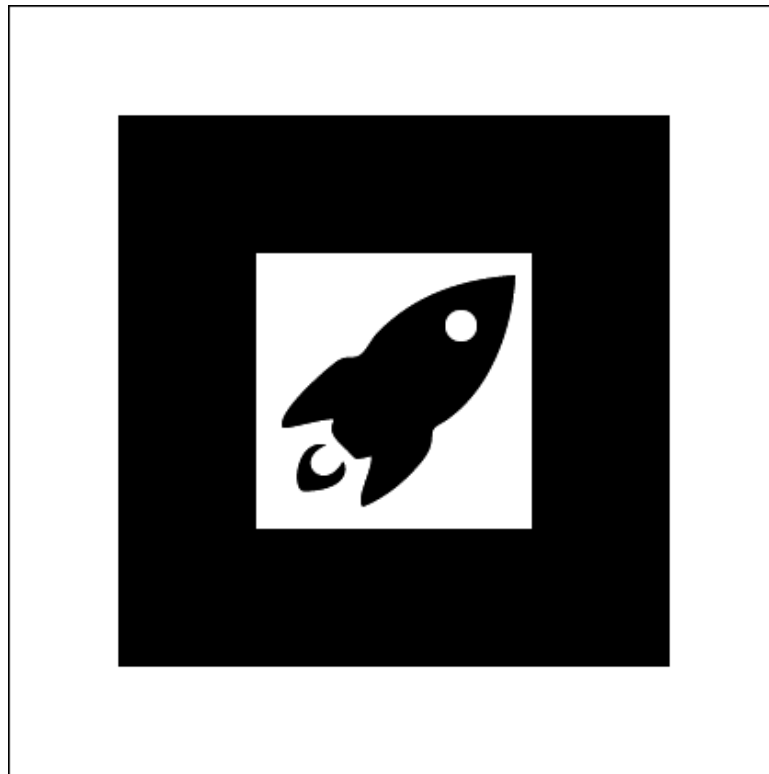
1 Disseny

1.1 marcadors

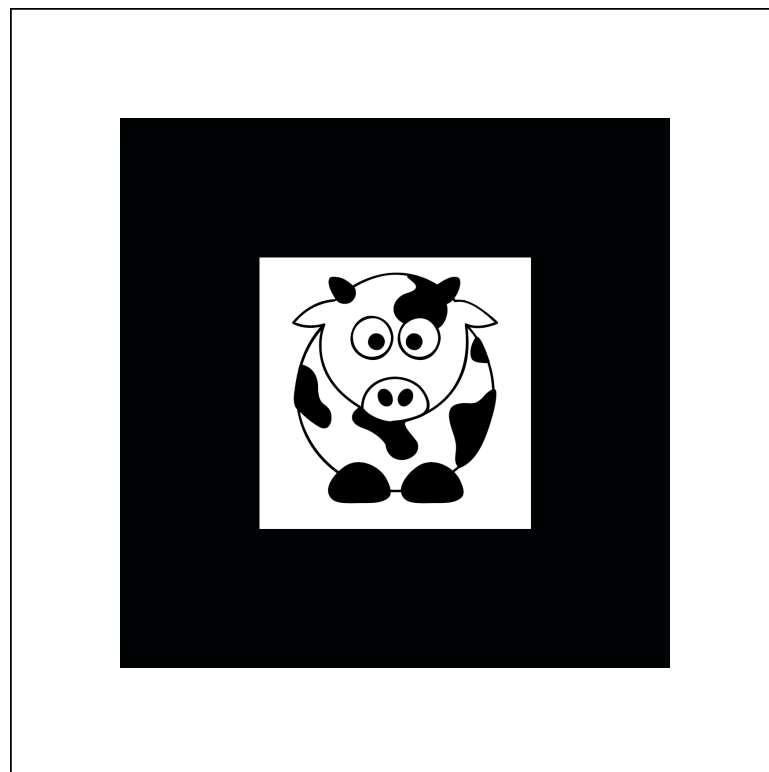
1.1.1 Patró de marcadors



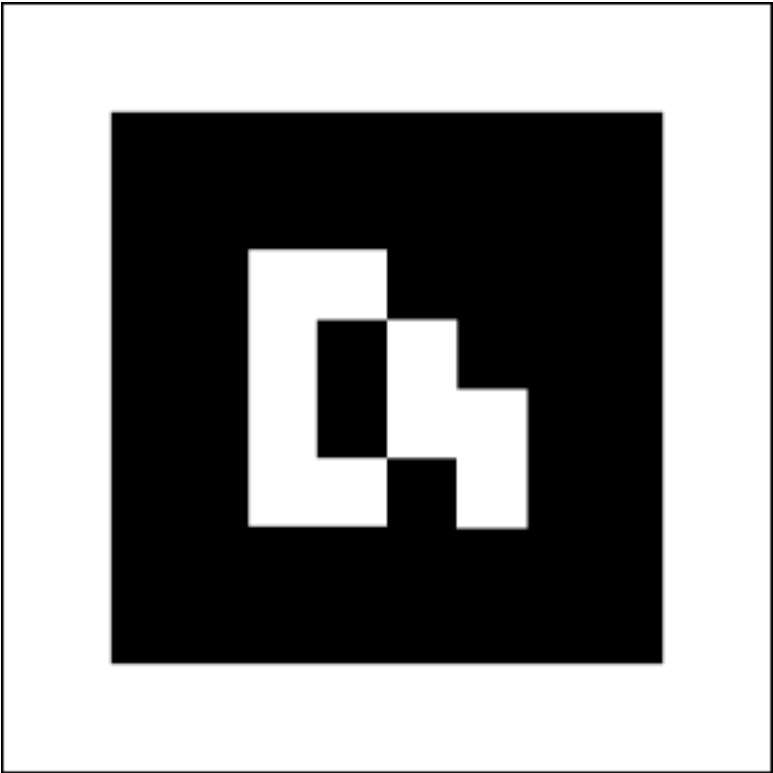
1.1.2 Marcador “coet”



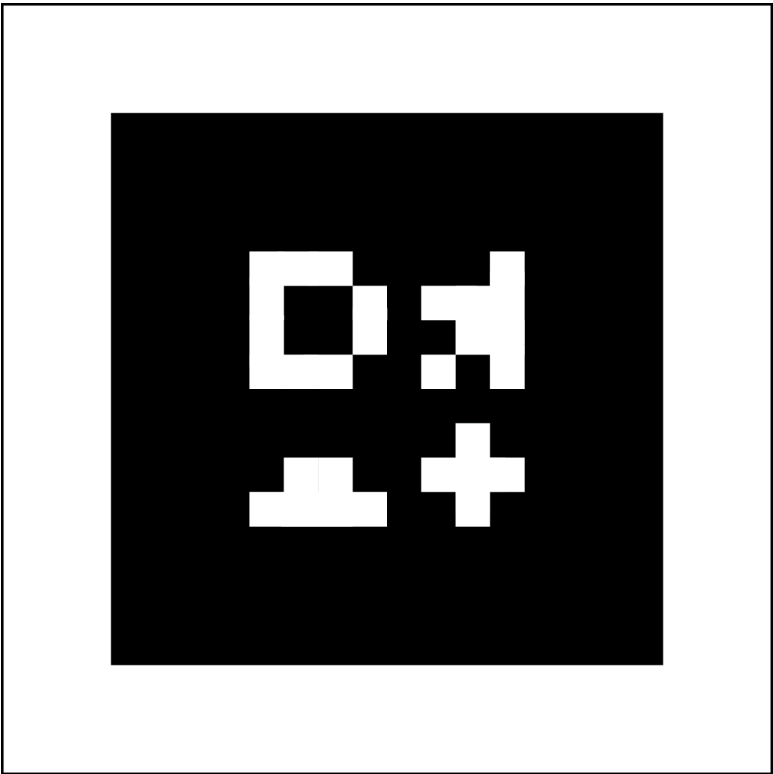
1.1.3 Marcador “vaca”



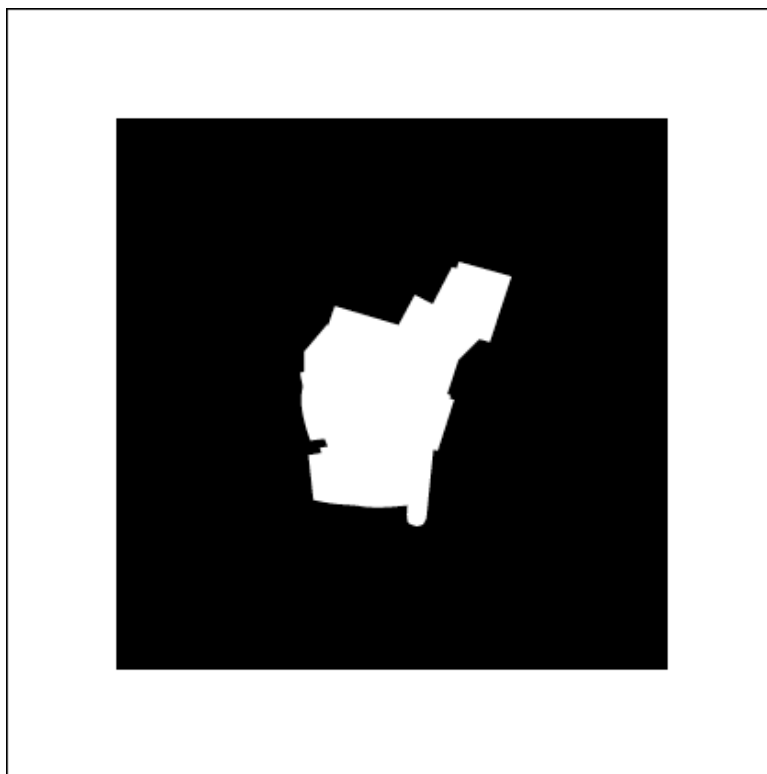
1.1.4 Marcador “cub 1”



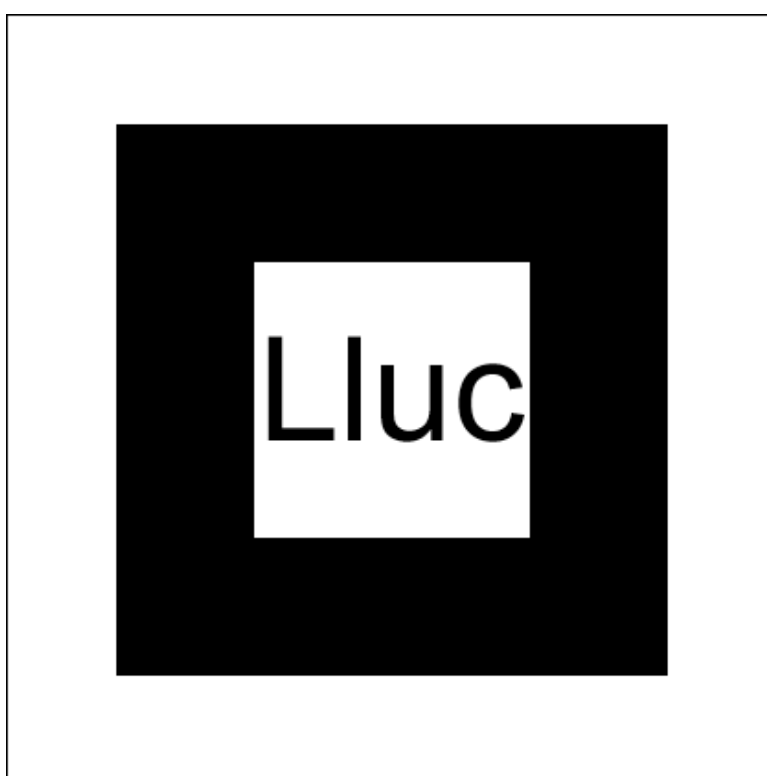
1.1.5 Marcador “cub 2”



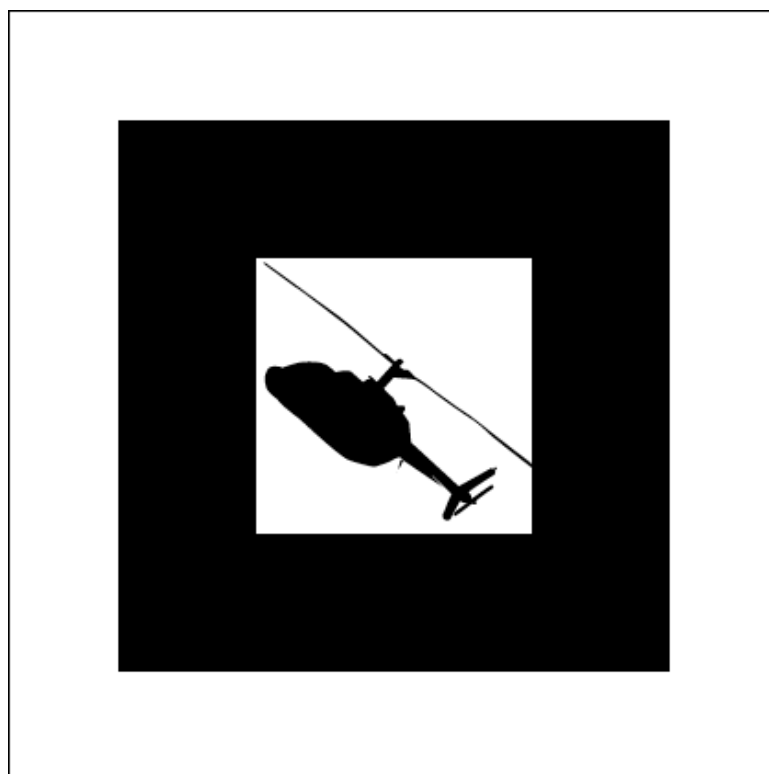
1.1.6 Marcador “Escorial”



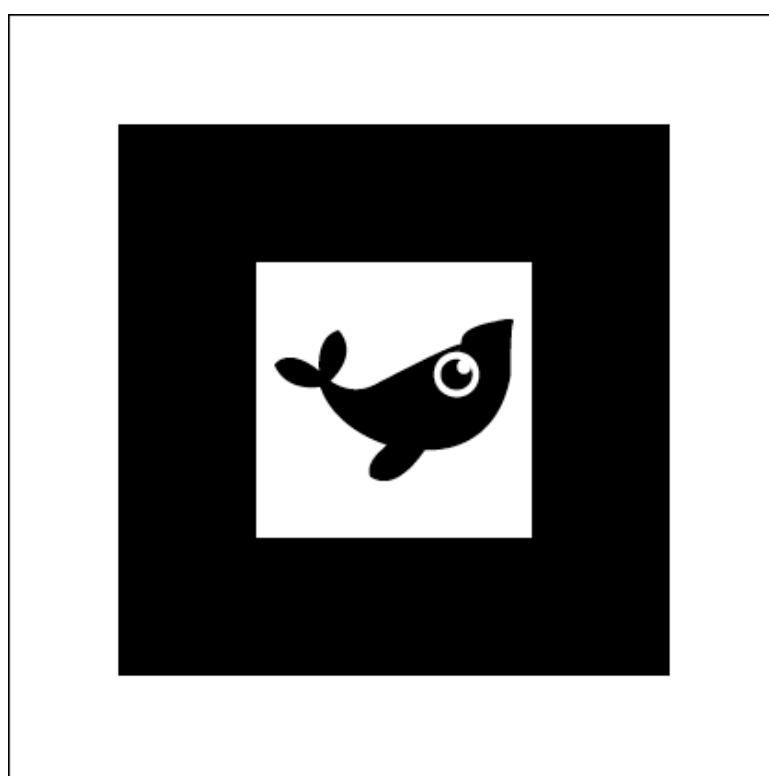
1.1.7 Marcador “Lluc”



1.1.8 Marcador “Helicòpter”

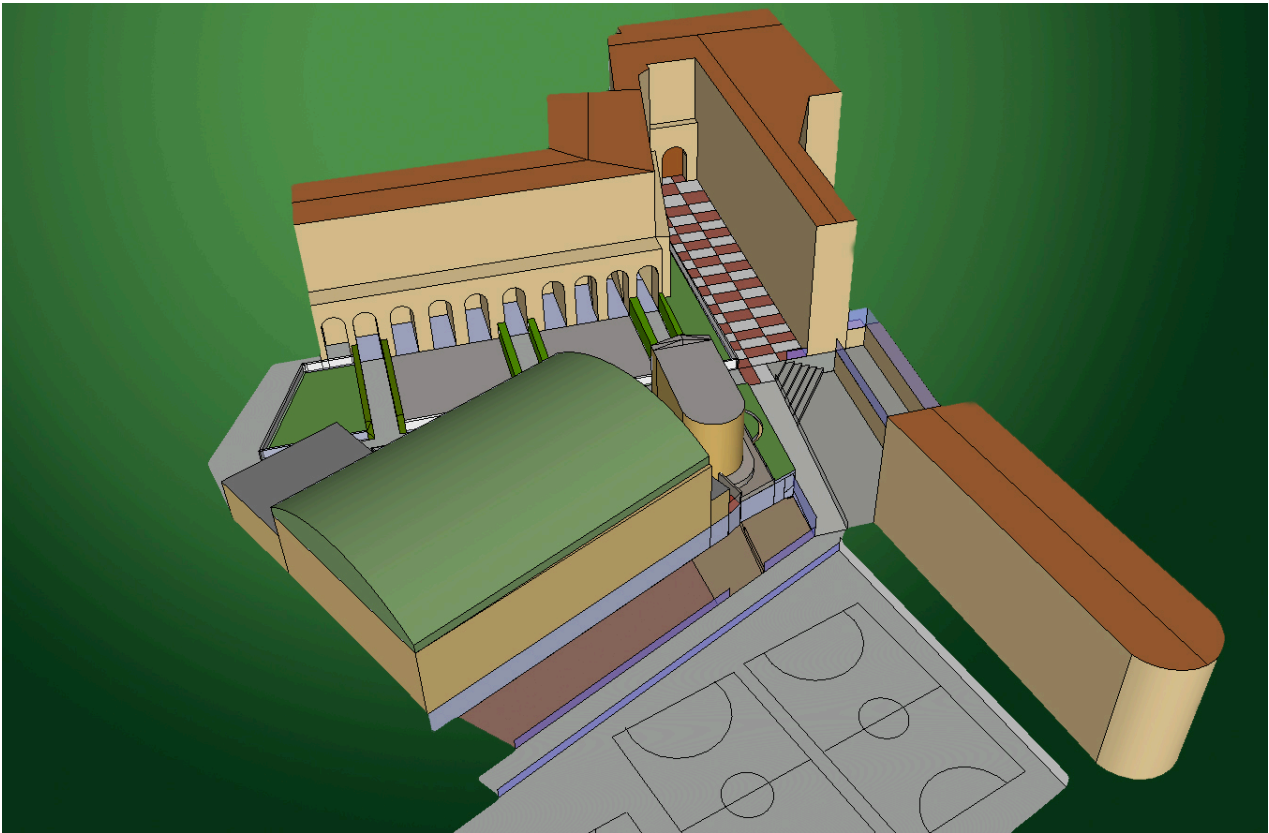


1.1.9 Marcador “orca”



1.2 Models tridimensionals

1.2.1 model “escorial”



1.2.2 Model “helicòpter”



1.2.3 Model “Lluc”

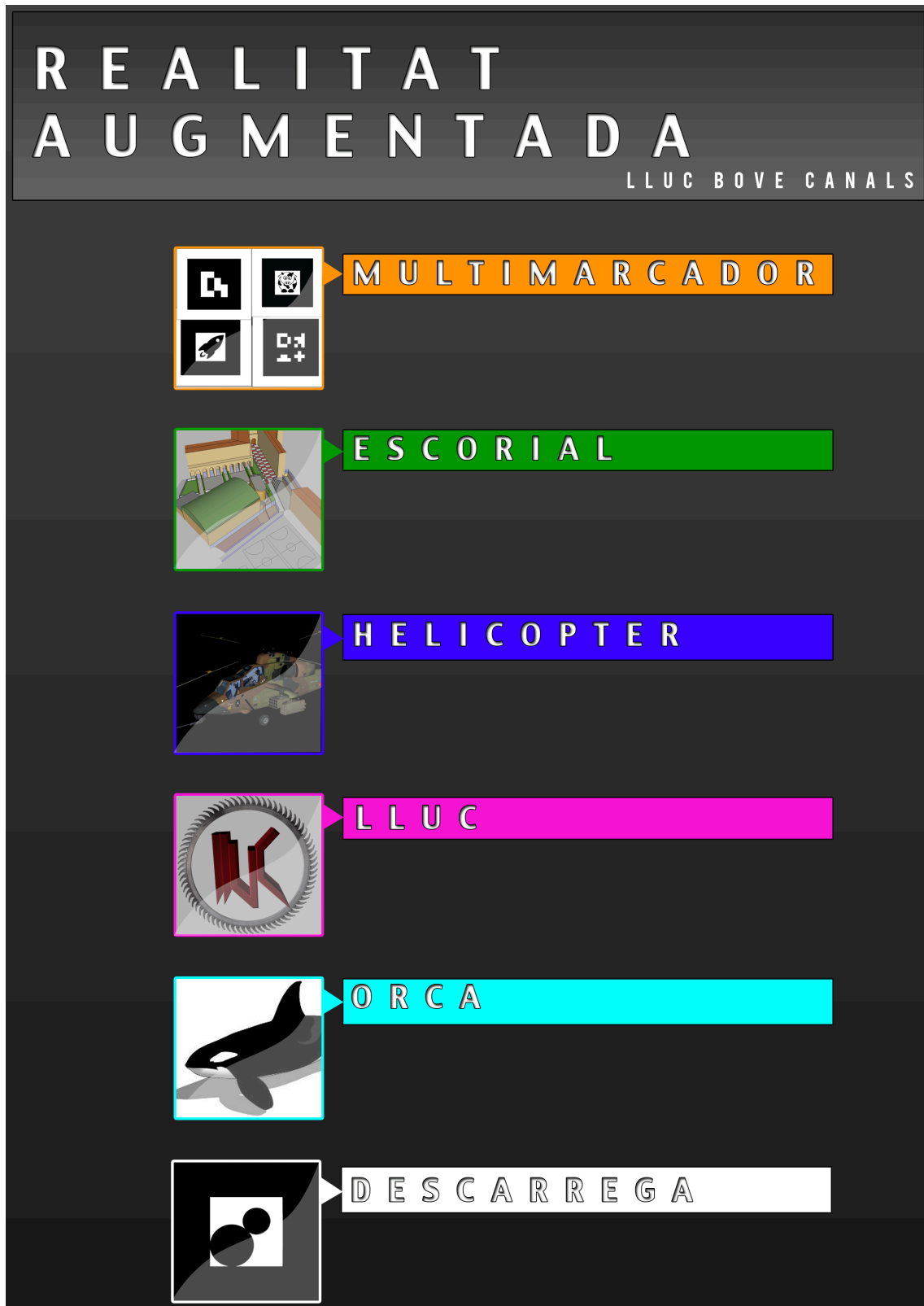


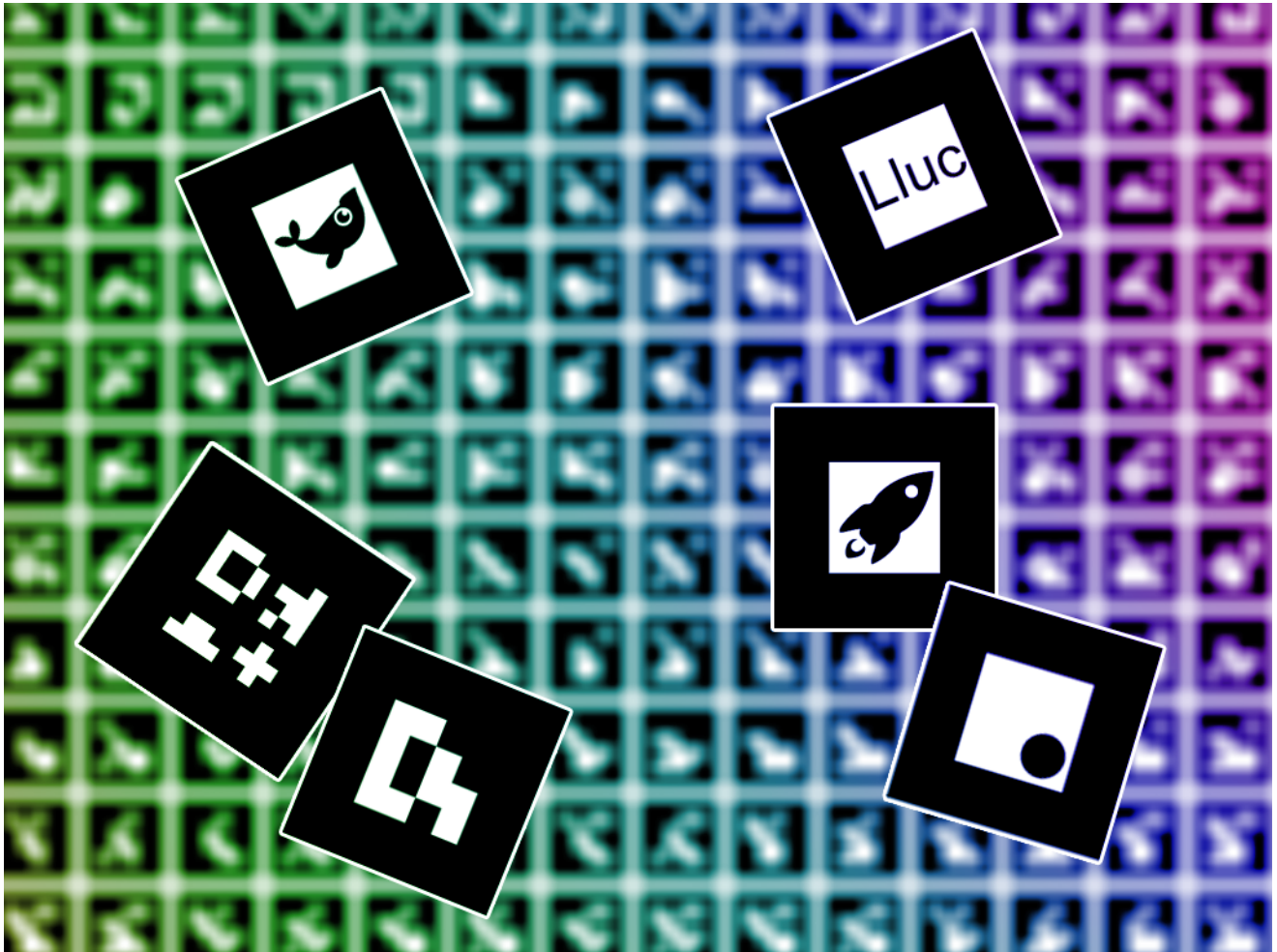
1.2.4 Model “orca”



1.3 Interfície

1.3.1 Mapa HTML





1.3.2 Fons de pantalla

2. Programació

2.1 Codi múltiple marcador

2.1.1 Importació de biblioteques i altres elements

```
package {  
    //importar biblioteques  
    import flash.events.Event;  
    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 com.greensock.*;
import com.squidder.flar.FLARMarkerObj;
import com.squidder.flar.PVFLARBaseApplication;
import com.squidder.flar.events.FLARDetectorEvent;
import org.papervision3d.objects.parsers.Collada;

public class MultiFLARExample extends PVFLARBaseApplication {

```

2.1.2 Classes privades

```

private var _cubes : Array;
private var _lightPoint : PointLight3D;
private var _green:Cube;
private var cowSkin: BitmapFileMaterial;
private var cowMat: MaterialsList;
private var cow: Collada;

public function MultiFLARExample() {

    _cubes = new Array();

```

2.1.3 Importació de marcadors

```

    _markers = new Array();
    _markers.push( new FLARMarkerObj( "assets/flar/
coet.pat" , 16 , 50 , 80 ) );
    _markers.push( new FLARMarkerObj( "assets/flar/
cow.pat" , 16 , 50 , 80 ) );
    _markers.push( new FLARMarkerObj( "assets/flar/
wizard.pat" , 16 , 50 , 80 ) );
    _markers.push( new FLARMarkerObj( "assets/flar/
multi.pat" , 16 , 50 , 80 ) );

    super( );

```

2.1.4 Detecció de marcadors

```

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 );
    }
}

```

2.1.5 Configurar els models tridimensionals en general

```

private function _addCube( id:int , index:int ) : void {

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

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

```

2.1.6 Configurar els materials

```

    var fmat : FlatShadeMaterial = new
FlatShadeMaterial( _lightPoint , 0xff22aa , 0x75104e );
        var fmat2 : FlatShadeMaterial = new
FlatShadeMaterial( _lightPoint , 0x00ff00 , 0x113311 );
        var fmat3 : FlatShadeMaterial = new
FlatShadeMaterial( _lightPoint , 0x0000ff , 0x111133 );
        var fmat4 : FlatShadeMaterial = new
FlatShadeMaterial( _lightPoint , 0x777777 , 0x111111 );
        var dispObj : DisplayObject3D = new
DisplayObject3D();
        var earth : BitmapFileMaterial = new
BitmapFileMaterial("assets/map.jpg");
        var star : ColorMaterial = new
ColorMaterial(0xFFFFFFFF);

```

```

        var Top : BitmapFileMaterial = new
BitmapFileMaterial("assets/top.png");
        var Bottom : BitmapFileMaterial = new
BitmapFileMaterial("assets/bottom.png");
        var Left : BitmapFileMaterial = new
BitmapFileMaterial("assets/left.png");
        var Right : BitmapFileMaterial = new
BitmapFileMaterial("assets/right.png");

        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
])
];

```

2.1.7 Processament del model “Terra”

```

if (id==0){

    //creant el cub que fa de forat
        var hole:Cube = dispObj.addChild(new Cube(new
MaterialsList({all:new BitmapFileMaterial( "assets/hole.jpg" ),
bottom: new BitmapFileMaterial( "assets/k.jpg" )}), 80,
80,80,1,1,1, Cube.ALL,Cube.TOP)) as Cube;

    //creant la terra
        this._green = dispObj.addChild(new
Cube(new MaterialsList({all: new ColorMaterial(0x00ff00)}), 80,
80,80, 1, 1, 1, Cube.TOP)) as Cube;
        hole.rotationX = this._green.rotationX =90;
        hole.z = this._green.z = -40;

    //animant la terra

        var Earth:Sphere = new Sphere(earth, 1);
        Earth.z=-40;
        TweenMax.to(Earth, 4,{scaleX:40, scaleY:40,
scaleZ:40, z:"200", delay:4});
        dispObj.addChild(Earth);

    //creant i animant les estrelles
        var star1:Sphere = new Sphere(star, 1);

```



```

star1.z=-40;
TweenMax.to(star1, 4,{scaleX:1, scaleY:1, scaleZ:1, x:"84",
y:"164", z:"65", delay:4});
dispObj.addChild(star1);

var star2:Sphere = new Sphere(star, 1);
star2.z=-40;
TweenMax.to(star2, 4,{scaleX:3, scaleY:3, scaleZ:3, x:"32",
y:"64", z:"246", delay:4});
dispObj.addChild(star2);

var star3:Sphere = new Sphere(star, 1);
star3.z=-40;
TweenMax.to(star3, 4,{scaleX:2, scaleY:2, scaleZ:2, x:"78",
y:"98", z:"163", delay:4});
dispObj.addChild(star3);

var star4:Sphere = new Sphere(star, 1);
star4.z=-40;
TweenMax.to(star4, 4,{scaleX:4, scaleY:4, scaleZ:4, x:"164",
y:"157", z:"120", delay:4});
dispObj.addChild(star4);

var star5:Sphere = new Sphere(star, 1);
star5.z=-40;
TweenMax.to(star5, 4,{scaleX:2, scaleY:2, scaleZ:2, x:"-164",
y:"-157", z:"148", delay:4});
dispObj.addChild(star5);

var star6:Sphere = new Sphere(star, 1);
star6.z=-40;
TweenMax.to(star6, 4,{scaleX:3, scaleY:3, scaleZ:3, x:"-36",
y:"-156", z:"46", delay:4});
dispObj.addChild(star6);

var star7:Sphere = new Sphere(star, 1);
star7.z=-40;
TweenMax.to(star7, 4,{scaleX:5, scaleY:5, scaleZ:5, x:"-16",
y:"-84", z:"40", delay:4});
dispObj.addChild(star7);

var star8:Sphere = new Sphere(star, 1);
star8.z=-40;
TweenMax.to(star8, 4,{scaleX:5, scaleY:5, scaleZ:5, x:"-84",
y:"30", z:"59", delay:4});
dispObj.addChild(star8);

var star9:Sphere = new Sphere(star, 1);
star9.z=-40;
TweenMax.to(star9, 4,{scaleX:4, scaleY:4, scaleZ:4, x:"-134",
y:"84", z:"87", delay:4});
dispObj.addChild(star9);

```



```

var star10:Sphere = new Sphere(star, 1);
star10.z=-40;
TweenMax.to(star10, 4,{scaleX:2, scaleY:2, scaleZ:2, x:"10",
y:"18", z:"49", delay:4});
dispObj.addChild(star10);

var star11:Sphere = new Sphere(star, 1);
star11.z=-40;
TweenMax.to(star11, 4,{scaleX:5, scaleY:5, scaleZ:5, x:"-84",
y:"41", z:"94", delay:4});
dispObj.addChild(star11);

var star12:Sphere = new Sphere(star, 1);
star12.z=-40;
TweenMax.to(star12, 4,{scaleX:3, scaleY:3, scaleZ:3, x:"91",
y:"-46", z:"54", delay:4});
dispObj.addChild(star12);

var star13:Sphere = new Sphere(star, 1);
star13.z=-40;
TweenMax.to(star13, 4,{scaleX:2, scaleY:2, scaleZ:2, x:"88",
y:"-130", z:"180", delay:4});
dispObj.addChild(star13);

var star14:Sphere = new Sphere(star, 1);
star14.z=-40;
TweenMax.to(star14, 4,{scaleX:4, scaleY:4, scaleZ:4, x:"134",
y:"-13", z:"102", delay:4});
dispObj.addChild(star14);

var star15:Sphere = new Sphere(star, 1);
star15.z=-40;
TweenMax.to(star15, 4,{scaleX:1, scaleY:1, scaleZ:1, x:"-35",
y:"145", z:"61", delay:4});
dispObj.addChild(star15);

var top:Cube = new Cube( new MaterialsList( {all: Top} ) , 80 ,
0 , 80 );
top.z=0;
top.y=40;
TweenMax.to(top, 2,{rotationX:-180, delay:2});
dispObj.addChild(top);

var bottom:Cube = new Cube( new MaterialsList( {all: Bottom} ) ,
80 , 0 , 80 );
bottom.z=0;
bottom.y=-40;
TweenMax.to(bottom, 2,{rotationX:180, delay:2});
dispObj.addChild(bottom);

```

```

var left:Cube = new Cube( new MaterialsList( {all: Left} ) , 80 ,
0 , 80 );
left.z=0;
left.rotationZ=90;
left.x=-40;
TweenMax.to(left, 2,{rotationX:180, delay:2});
dispObj.addChild(left);

var right:Cube = new Cube( new MaterialsList( {all: Right} ) ,
80 , 0 , 80 );
right.z=0;
right.x=40;
right.rotationZ=90;
TweenMax.to(right, 2,{rotationX:-180, delay:2});
dispObj.addChild(right);
}

```

2.1.8 Processament del model “vaca”

```

else if(id==1){

        cowMat = new MaterialsList();

//Creant la vaca
cow = new Collada("assets/cow.dae",cowMat);
cowSkin = new BitmapFileMaterial("assets/cow.png");
cowMat.addMaterial(cowSkin,"all");

cow.rotationX = 90;
cow.scale = 0.5;
dispObj.addChild(cow);

```

2.1.9 Processament dels cubs

```

} else if (id==2) {

        var cube3 : Cube = new Cube( new
MaterialsList( {all: fmat3} ) , 40 , 40 , 40 );
        cube3.z =0;
        dispObj.addChild( cube3 );

        } else if (id==3){

        var cube4 : Cube = new Cube( new
MaterialsList( {all: fmat4} ) , 40 , 40 , 40 );
        cube4.z = 60;
        dispObj.addChild( cube4 )

```

2.1.10 Codi font total

```

package {
    //importar biblioteques
    import flash.events.Event;
    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 com.greensock.*;
    import com.squidder.flar.FLARMarkerObj;
    import com.squidder.flar.PVFLARBaseApplication;
    import com.squidder.flar.events.FLARDetectorEvent;
    import org.papervision3d.objects.parsers.Collada;

    public class MultiFLARExample extends PVFLARBaseApplication {

        // variables privades que funcionen al llarg del codi
        private var _cubes : Array;
        private var _lightPoint : PointLight3D;
        private var _green:Cube;
        private var cowSkin: BitmapFileMaterial;
        private var cowMat: MaterialsList;
        private var cow: Collada;

        public function MultiFLARExample() {

            _cubes = new Array();

            // importar el marcador
            _markers = new Array();
            _markers.push( new FLARMarkerObj( "assets/flare/
coet.pat" , 16 , 50 , 80 ) );
            _markers.push( new FLARMarkerObj( "assets/flare/
cow.pat" , 16 , 50 , 80 ) );
            _markers.push( new FLARMarkerObj( "assets/flare/
wizard.pat" , 16 , 50 , 80 ) );
            _markers.push( new FLARMarkerObj( "assets/flare/
multi.pat" , 16 , 50 , 80 ) );

            super( );
        }

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

```

```

        super._init( event );

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

    }
    //detectar el marcador
    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 );
    }

    //afegint els models

    private function _addCube( id:int , index:int ) : void {

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

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

```

```

        //configurant els materials
        var fmat : FlatShadeMaterial = new
FlatShadeMaterial( _lightPoint , 0xff22aa , 0x75104e );
        var fmat2 : FlatShadeMaterial = new
FlatShadeMaterial( _lightPoint , 0x00ff00 , 0x113311 );
        var fmat3 : FlatShadeMaterial = new
FlatShadeMaterial( _lightPoint , 0x0000ff , 0x111133 );
        var fmat4 : FlatShadeMaterial = new
FlatShadeMaterial( _lightPoint , 0x777777 , 0x111111 );
        var dispObj : DisplayObject3D = new
DisplayObject3D();
        var earth : BitmapFileMaterial = new
BitmapFileMaterial("assets/map.jpg");
        var star : ColorMaterial = new
ColorMaterial(0xFFFFF);
        var Top : BitmapFileMaterial = new
BitmapFileMaterial("assets/top.png");
        var Bottom : BitmapFileMaterial = new
BitmapFileMaterial("assets/bottom.png");
        var Left : BitmapFileMaterial = new
BitmapFileMaterial("assets/left.png");
        var Right : BitmapFileMaterial = new
BitmapFileMaterial("assets/right.png");

        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
])
];
        //condicional (if==0,1,2,3.)
        if (id==0){

            //creant el cub que fa de forat
            var hole:Cube = dispObj.addChild(new Cube(new
MaterialsList({all:new BitmapFileMaterial( "assets/hole.jpg" ),
bottom: new BitmapFileMaterial( "assets/k.jpg" })), 80,
80,80,1,1,1, Cube.ALL,Cube.TOP)) as Cube;

            //creant la terra
            this._green = dispObj.addChild(new
Cube(new MaterialsList({all: new ColorMaterial(0x00ff00)}), 80,
80,80, 1, 1, 1, Cube.TOP)) as Cube;
            hole.rotationX = this._green.rotationX =90;
            hole.z = this._green.z = -40;

            //animant la terra

```

```

        var Earth:Sphere = new Sphere(earth, 1);
        Earth.z=-40;
        TweenMax.to(Earth, 4,{scaleX:40, scaleY:40,
scaleZ:40, z:"200", delay:4});
        dispObj.addChild(Earth);

        //creant i animant les estrelles
        var star1:Sphere = new Sphere(star, 1);
        star1.z=-40;
        TweenMax.to(star1, 4,{scaleX:1, scaleY:1, scaleZ:1, x:"84",
y:"164", z:"65", delay:4});
        dispObj.addChild(star1);

        var star2:Sphere = new Sphere(star, 1);
        star2.z=-40;
        TweenMax.to(star2, 4,{scaleX:3, scaleY:3, scaleZ:3, x:"32",
y:"64", z:"246", delay:4});
        dispObj.addChild(star2);

        var star3:Sphere = new Sphere(star, 1);
        star3.z=-40;
        TweenMax.to(star3, 4,{scaleX:2, scaleY:2, scaleZ:2, x:"78",
y:"98", z:"163", delay:4});
        dispObj.addChild(star3);

        var star4:Sphere = new Sphere(star, 1);
        star4.z=-40;
        TweenMax.to(star4, 4,{scaleX:4, scaleY:4, scaleZ:4, x:"164",
y:"157", z:"120", delay:4});
        dispObj.addChild(star4);

        var star5:Sphere = new Sphere(star, 1);
        star5.z=-40;
        TweenMax.to(star5, 4,{scaleX:2, scaleY:2, scaleZ:2, x:"-164",
y:"-157", z:"148", delay:4});
        dispObj.addChild(star5);

        var star6:Sphere = new Sphere(star, 1);
        star6.z=-40;
        TweenMax.to(star6, 4,{scaleX:3, scaleY:3, scaleZ:3, x:"-36",
y:"-156", z:"46", delay:4});
        dispObj.addChild(star6);

        var star7:Sphere = new Sphere(star, 1);
        star7.z=-40;
        TweenMax.to(star7, 4,{scaleX:5, scaleY:5, scaleZ:5, x:"-16",
y:"-84", z:"40", delay:4});
        dispObj.addChild(star7);

        var star8:Sphere = new Sphere(star, 1);
        star8.z=-40;

```

```

TweenMax.to(star8, 4,{scaleX:5, scaleY:5, scaleZ:5, x:"-84",
y:"30", z:"59", delay:4});
dispObj.addChild(star8);

var star9:Sphere = new Sphere(star, 1);
star9.z=-40;
TweenMax.to(star9, 4,{scaleX:4, scaleY:4, scaleZ:4, x:"-134",
y:"84", z:"87", delay:4});
dispObj.addChild(star9);

var star10:Sphere = new Sphere(star, 1);
star10.z=-40;
TweenMax.to(star10, 4,{scaleX:2, scaleY:2, scaleZ:2, x:"10",
y:"18", z:"49", delay:4});
dispObj.addChild(star10);

var star11:Sphere = new Sphere(star, 1);
star11.z=-40;
TweenMax.to(star11, 4,{scaleX:5, scaleY:5, scaleZ:5, x:"-84",
y:"41", z:"94", delay:4});
dispObj.addChild(star11);

var star12:Sphere = new Sphere(star, 1);
star12.z=-40;
TweenMax.to(star12, 4,{scaleX:3, scaleY:3, scaleZ:3, x:"91",
y:"-46", z:"54", delay:4});
dispObj.addChild(star12);

var star13:Sphere = new Sphere(star, 1);
star13.z=-40;
TweenMax.to(star13, 4,{scaleX:2, scaleY:2, scaleZ:2, x:"88",
y:"-130", z:"180", delay:4});
dispObj.addChild(star13);

var star14:Sphere = new Sphere(star, 1);
star14.z=-40;
TweenMax.to(star14, 4,{scaleX:4, scaleY:4, scaleZ:4, x:"134",
y:"-13", z:"102", delay:4});
dispObj.addChild(star14);

var star15:Sphere = new Sphere(star, 1);
star15.z=-40;
TweenMax.to(star15, 4,{scaleX:1, scaleY:1, scaleZ:1, x:"-35",
y:"145", z:"61", delay:4});
dispObj.addChild(star15);

var top:Cube = new Cube( new MaterialsList( {all: Top} ) , 80 ,
0 , 80 );
top.z=0;
top.y=40;
TweenMax.to(top, 2,{rotationX:-180, delay:2});
dispObj.addChild(top);

```

```

var bottom:Cube = new Cube( new MaterialsList( {all: Bottom} ) ,
80 , 0 , 80 );
bottom.z=0;
bottom.y=-40;
TweenMax.to(bottom, 2,{rotationX:180, delay:2});
dispObj.addChild(bottom);

var left:Cube = new Cube( new MaterialsList( {all: Left} ) , 80 ,
0 , 80 );
left.z=0;
left.rotationZ=90;
left.x=-40;
TweenMax.to(left, 2,{rotationX:180, delay:2});
dispObj.addChild(left);

var right:Cube = new Cube( new MaterialsList( {all: Right} ) ,
80 , 0 , 80 );
right.z=0;
right.x=40;
right.rotationZ=90;
TweenMax.to(right, 2,{rotationX:-180, delay:2});
dispObj.addChild(right);
    }

    else if(id==1){

        cowMat = new MaterialsList();

//Creant la vaca
cow = new Collada("assets/cow.dae",cowMat);
cowSkin = new BitmapFileMaterial("assets/cow.png");
cowMat.addMaterial(cowSkin,"all");

cow.rotationX = 90;
cow.scale = 0.5;
dispObj.addChild(cow);

//els altres dos cubs
    } else if (id==2) {

        var cube3 : Cube = new Cube( new
MaterialsList( {all: fmat3} ) , 40 , 40 , 40 );
        cube3.z =0;
        dispObj.addChild( cube3 );

    } else if (id==3){

```



```

        var cube4 : Cube = new Cube( new
MaterialsList( {all: fmat4} ) , 40 , 40 , 40 );
        cube4.z = 60;
        dispObj.addChild( cube4 );

    }
    _baseNode.addChild( dispObj );

    _cubes[ id ][ index ] = dispObj;

}

_baseNode.addChild( _cubes[ id ][ index ] );

}

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 ] );
    }

}

}
}

```

2.2 Codi d'un sol marcadore

2.2.1 Importació de biblioteques i altres elements

```

package {
//importar biblioteques i altres

import flash.events.Event;
import org.papervision3d.objects.parsers.DAE;

```

2.2.2 Definició de variables privades

```

private var _escorial:DAE;

```

2.2.3 Funció reconeixement del marcador

```
public function escorial() {  
    addEventListener(Event.INIT, _onInit);  
    init('Data/camera_para.dat', 'Data/escorial.pat');
```

2.2.4 Funció que afegeix el model tridimensional

```
    private function _onInit(e:Event):void {  
        _escorial = new DAE(true,null,true);  
        _escorial.load('model/escorial.dae');  
        _escorial.scale = 0.008;  
        _escorial.rotationX = 90;  
        _markerNode.addChild(_escorial);  
  
        addEventListener(Event.ENTER_FRAME, _update);  
    }
```

2.2.5 Funció gir

```
    private function _update(e:Event):void {  
        _escorial.rotationZ -= 0  
    }
```

2.2.6 Codi font total

```
package {  
    //importar biblioteques i altres  
  
    import flash.events.Event;  
    import org.papervision3d.objects.parsers.DAE;  
  
    //crear SWF  
    [SWF(width=640, height=480, backgroundColor=0x808080,  
frameRate=30)]  
  
    //portar la variable escorial a una PV.  
    publica class escorial extends PV3DARApp {  
  
        //variables privades  
        private var _escorial:DAE;  
  
        //funció identificar marcador  
        public function escorial() {  
            addEventListener(Event.INIT, _onInit);  
            init('Data/camera_para.dat', 'Data/escorial.pat');  
        }  
        //funció carregar models  
        private function _onInit(e:Event):void {  
            _escorial = new DAE(true,null,true);
```

```

        _escorial.load('model/escorial.dae');
        _escorial.scale = 0.008;
        _escorial.rotationX = 90;
        _markerNode.addChild(_escorial);

        addEventListener(Event.ENTER_FRAME, _update);
    }
    //funció rotació
    private function _update(e:Event):void {
        _escorial.rotationZ -= 0
    }
}

```