

# Grafica Computazionale

Laboratorio:  
il progetto SoftOgl

Fabio Ganovelli

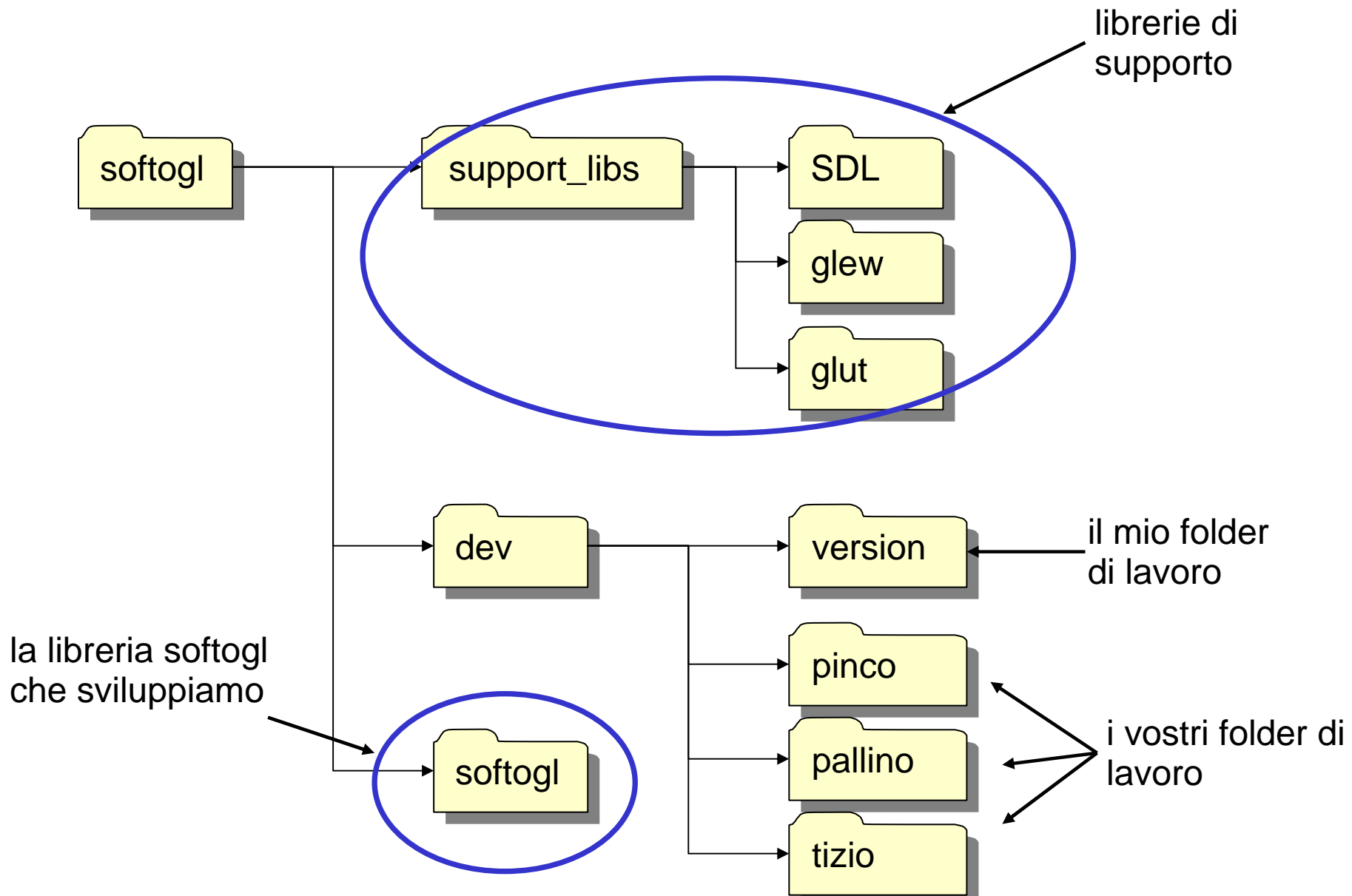
[fabio.ganovelli@isti.cnr.it](mailto:fabio.ganovelli@isti.cnr.it)

a.a. 2005-2006

## Obiettivi

- Implementare gli algoritmi di base della pipeline di rendering rasterization based
- Assemblarli in una API grafica *funzionante*

# Il progetto: struttura dei folder



# Files di softogl: softogl/softogl/softogl.h

```
namespace softogl{
    // ***** softogl API calls *****
    // ***** functions provided to the user *****
    namespace impl{
        // ***** function and data structure to implement the functions
        // ***** declared in softogl::
        // ***** 3D ENTITIES *****
        // STILL EMPTY
        // ***** 2D ENTITIES *****
        .....
        // ***** RASTERIZATION *****
        .....
        // DDA segment rasterization
        void DDASegmentRasterization(Segment2d s,int size=1);

        // Bresenham segment rasterization
        void BresenhamSegmentRasterization(Segment2d s,int size = 1);

        // Polygon Filling
        void PolygonFilling(Polygon2d p);

    }// end impl

    namespace sdk{
        // ***** utility functions for debugging
        ..
        void Line(float start_x, float star_y, float end_x, float end_y, float width = 1);
        ...
    }// end sdk
};
```

# Files di softogl: softogl/softogl/softogl.cpp

```
namespace softogl{
    namespace impl{
        // DDA segment rasterization
        // void DDASegmentRasterization(Segment2d s,int size=1){
        // }

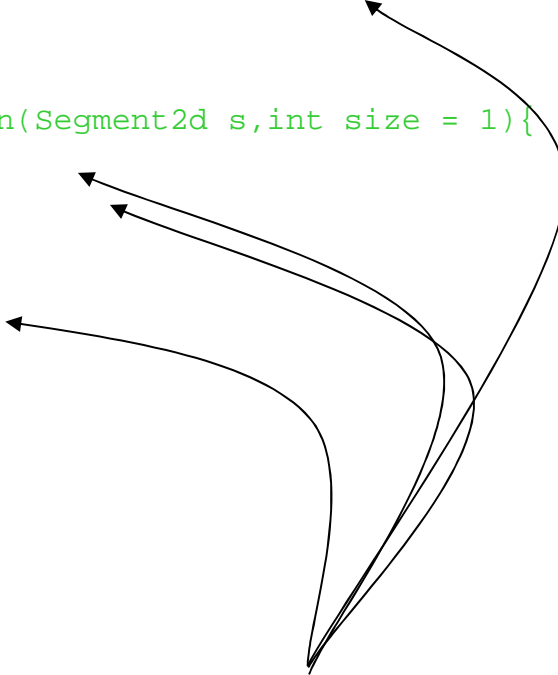
        // Breseham segment rasterization
        // void BresenhamSegmentRasterization(Segment2d s,int size = 1){
        // }

        // Polygon Filling
        // void PolygonFilling(Polygon2d p){
        // }

        // }// end impl

    namespace sdk{

} // end sdk
};
```

A diagram consisting of several black arrows originates from a single point on the right side of the slide. Four arrows point to the function definitions: one to the DDA function, one to the Bresenham function, one to the Polygon Filling function, and one to the closing brace of the 'impl' namespace. A fifth arrow points to the closing brace of the 'sdk' namespace.

Implementazione delle funzioni dichiarate in softogl.h.

Al momento contiene tutte le funzioni che servono a implementare gli algoritmi di rasterizzazione

## Files di sviluppo:

softogl/dev/**tizio**/rasterization.cpp

```
namespace softogl{
    namespace impl{

        // DDA segment rasterization
        void DDASegmentRasterization(Segment2d s,int size=1){
        }

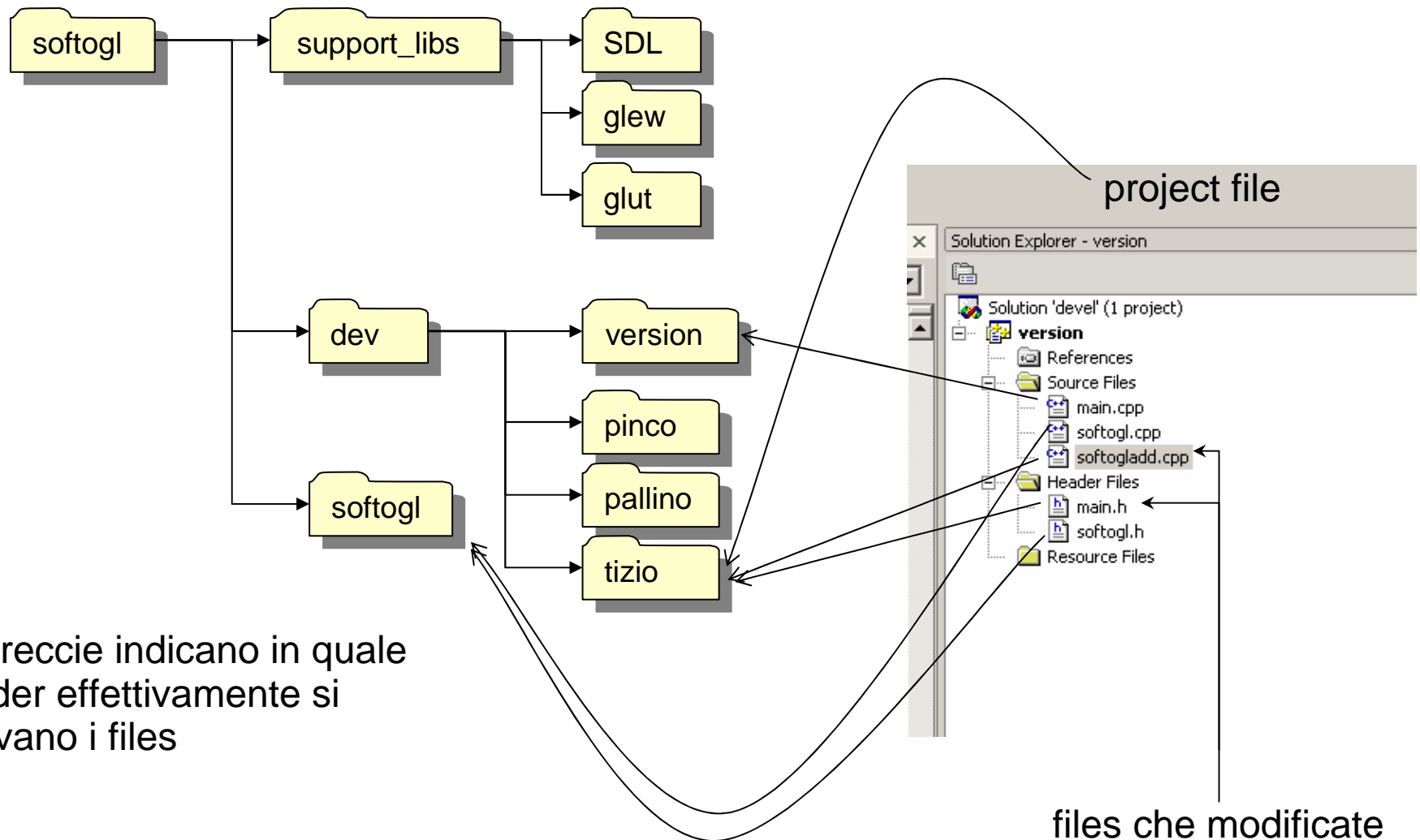
        // Breseham segment rasterization
        void BresenhamSegmentRasterization(Segment2d s,int size = 1){
        }

        // Polygon Filling
        void PolygonFilling(Polygon2d p){
        }

    };
```

Ognuno, NEL PROPRIO FOLDER, ha un file softogladd.cpp che contiene l'implementazione delle funzioni dichiarate in softogl/softogl/softogl.h ma non ancora implementate in softogl/softogl/softogl.cpp

# Struttura del progetto c++ di tizio: i files



# Struttura del progetto c++ di tizio: i path

## Include paths:

```
../../support_libs;  
../tizio
```

## Library paths:

```
../../support_libs/glut/lib;  
../../support_libs/glew/lib;  
../../support_libs/sdl/lib
```

## Librerie da linkare:

```
sdl.lib  
sdlmain.lib  
glew32.lib  
glut32.lib
```



# esempio minimo

softogl/tizio/main.h

```
#pragma once
#include "../softogl/softogl.h"
```

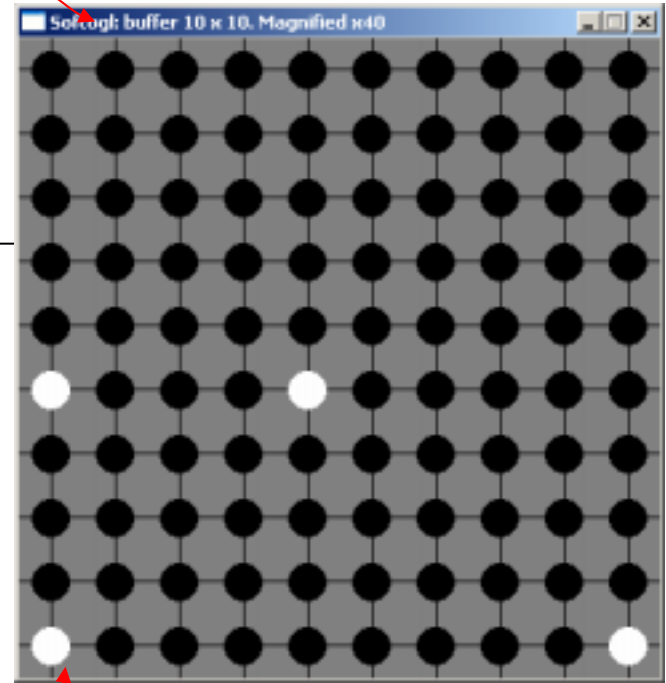
```
bool init() {
    softogl::impl::CreateWindow(10,10,40);
    return true;
}
```

```
void display(){
```

```
    /*****
    softogl::impl::ClearWindow();
```

```
    softogl::impl::Pixel(4,4) = softogl::impl::ON;
    softogl::impl::Pixel(0,4) = softogl::impl::ON;
    softogl::impl::Pixel(0,0) = softogl::impl::ON;
    softogl::impl::Pixel(9,0) = softogl::impl::ON;
```

```
}
```

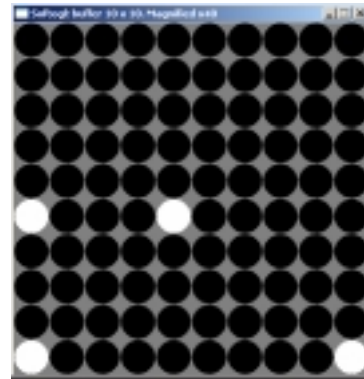
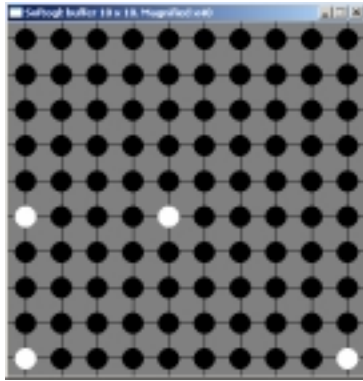


# Strutture dati

```
/**/struct Point2d{  
/**/    float p[2];  
/**/    float & operator [](int i){ return p[i];}  
/**/    };  
/**/  
/**/// Classe vertice  
/**/struct Vertex2d{  
/**/    Point2d pos;  
/**/    float operator [](int i){ return pos[i];}  
/**/    Vertex2d();  
/**/    Vertex2d(float x,float y);  
/**/    };  
/**/  
/**/// Classe poligono in due dimensioni  
/**/struct Polygon2d{  
/**/// vertici del poligono v0,..,vn. GLi spigoli sono tutte le  
/**/// coppie di vertici consecutive e (vn,v0)  
/**/std::vector<Vertex2d> vertices;  
/**/};  
/**/  
/**/ // Segmento in 2 dimensioni  
/**/ struct Segment2d{  
/**/ // vertici del segmento  
/**/    Vertex2d p0,p1;  
/**/};
```

Sembra una ridondanza,  
ma in futuro ci servirà  
esprimere il fatto che un vertice  
è più di una posizione  
nello spazio

# Funzionalità

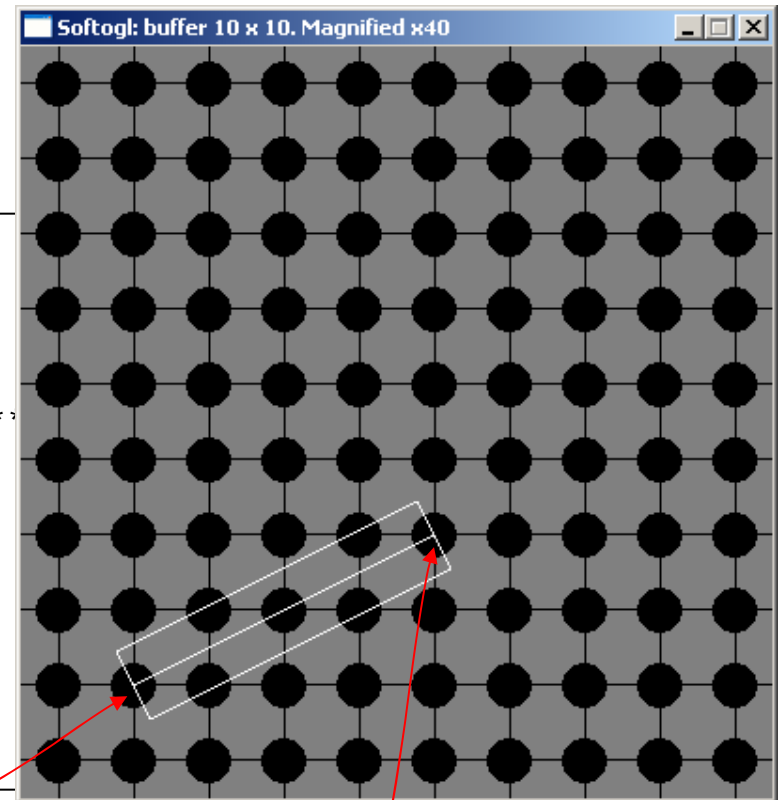


girando la rotella del mouse

# Funzionalità

softogl/tizio/main.h

```
.....  
void display(){  
  
    /*****  
    softogl::impl::ClearWindow();  
  
    softogl::sdk::Segment2d(1,1,5,3,1);  
}
```



# Altro esempio minimo

softogl/tizio/main.h

```
#pragma once
#include "../softogl/softogl.h"

bool init() {
    softogl::impl::CreateWindow(20,20,40);
    return true;
}

void display(){

    softogl::impl::ClearWindow();

    softogl::impl::Segment2d segment;
    segment.p0 = softogl::impl::Vertex2d(0,5);
    segment.p1 = softogl::impl::Vertex2d(16,9);

    softogl::impl::DDASegmentRasterization(segment,1.0);
    softogl::sdk::DrawSegment2d(0,5,16,9,1);

}
```

