

# Visibilidad

---

## Ejercicios de clase

# Zbuffer

- Se desea visualizar una escena (Figura 1) que está compuesta por tres polígonos paralelos al plano XY en un raster de 4x4 píxeles, para obtener la imagen que se muestra en la Figura 2. Si se aplica el algoritmo de Z-Buffer para resolver la visibilidad de la escena, especifica el estado del frame-buffer y del Z-Buffer después de dibujar  $P_1$ ,  $P_2$  y  $P_3$ .

Figura 1

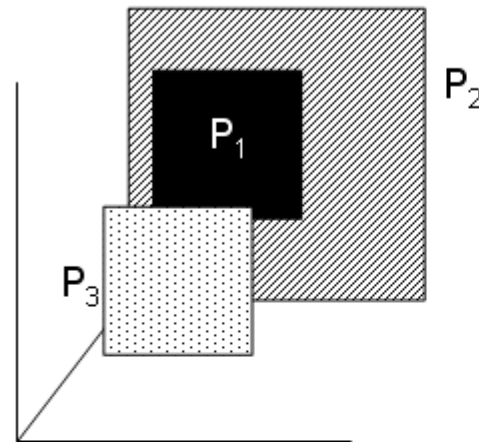
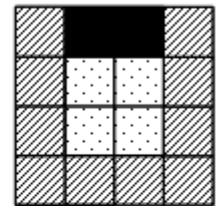
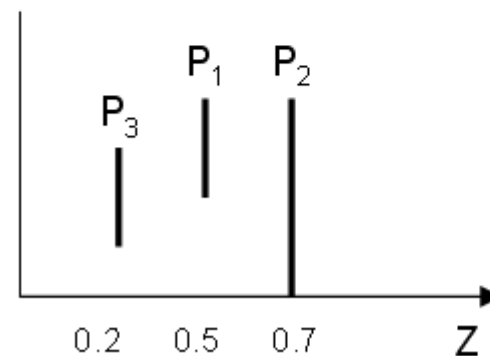
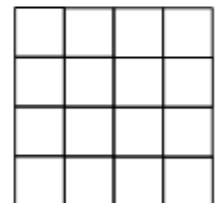


Figura 2



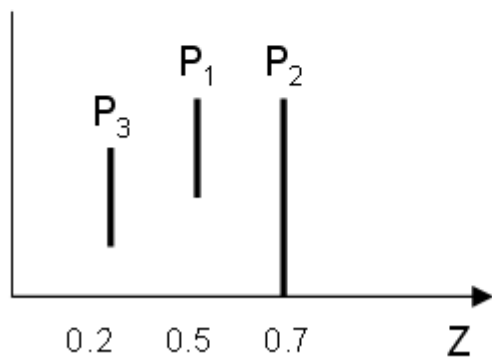
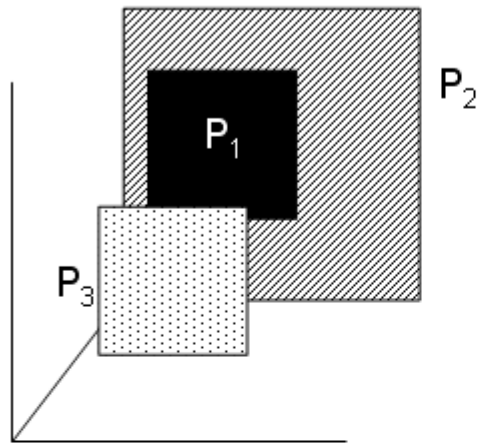
Raster (4x4)



Posiciones de los polígonos

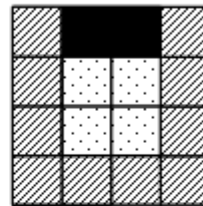
# ZBuffer

Figura 1

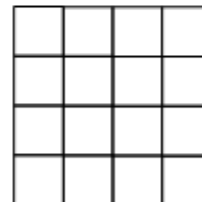


Posiciones de los polígonos

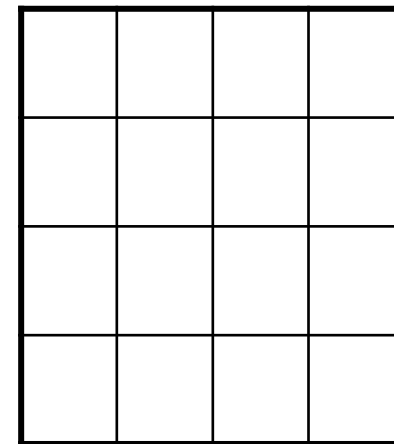
Figura 2



Raster (4x4)



## Inicialización Frame buffer

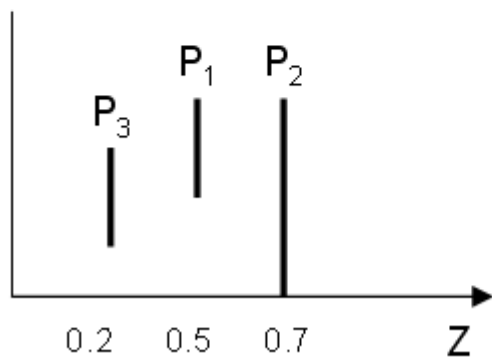
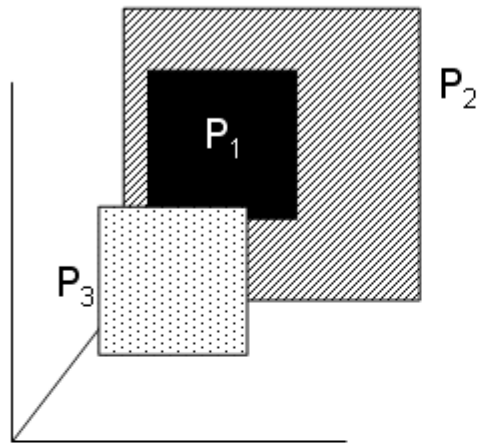


Z-buffer

1	1	1	1
1	1	1	1
1	1	1	1
1	1	1	1

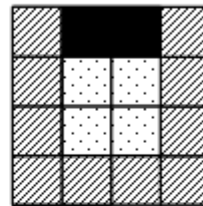
# ZBuffer

Figura 1

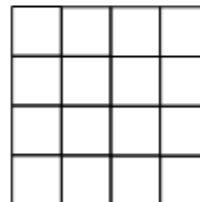


Posiciones de los polígonos

Figura 2

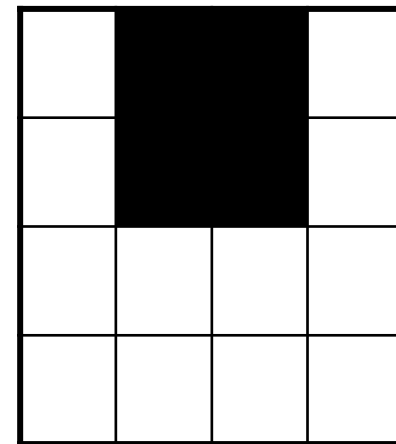


Raster (4x4)



Después de P<sub>1</sub>

Frame buffer

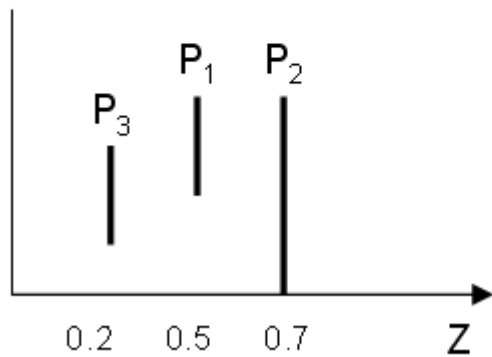
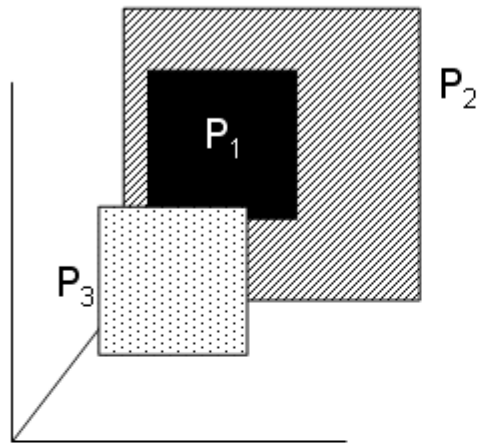


Z-buffer

1	0.5	0.5	1
1	0.5	0.5	1
1	1	1	1
1	1	1	1

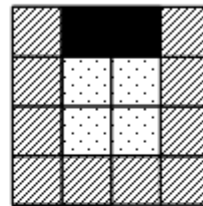
# ZBuffer

Figura 1

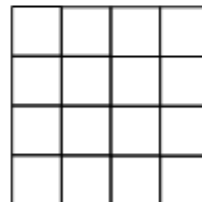


Posiciones de los polígonos

Figura 2

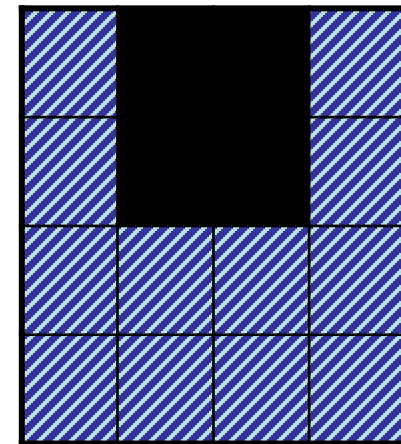


Raster (4x4)



Después de P2

Frame buffer

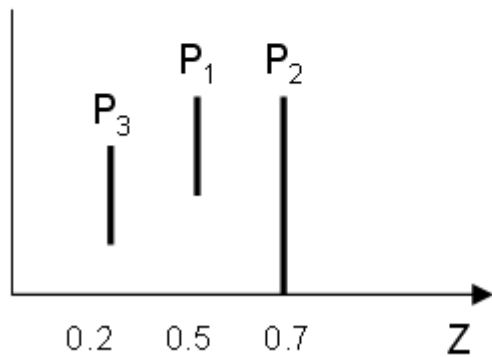
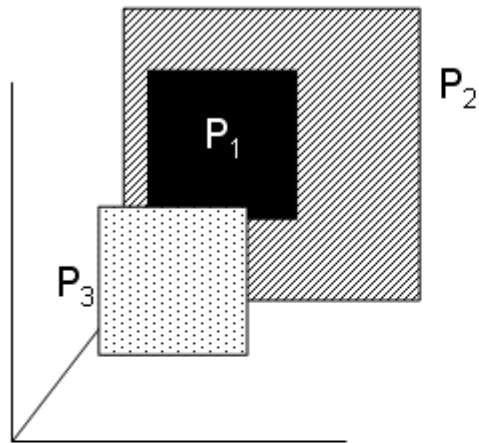


Z-buffer

0.7	0.5	0.5	0.7
0.7	0.5	0.5	0.7
0.7	0.7	0.7	0.7
0.7	0.7	0.7	0.7

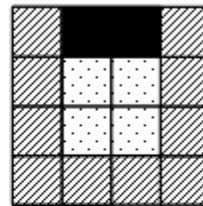
# ZBuffer

Figura 1

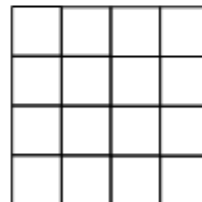


Posiciones de los polígonos

Figura 2

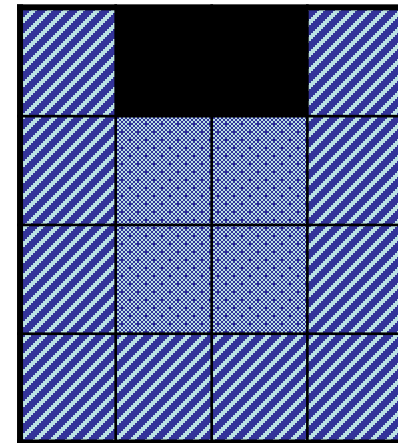


Raster (4x4)



Después de P3

Frame buffer

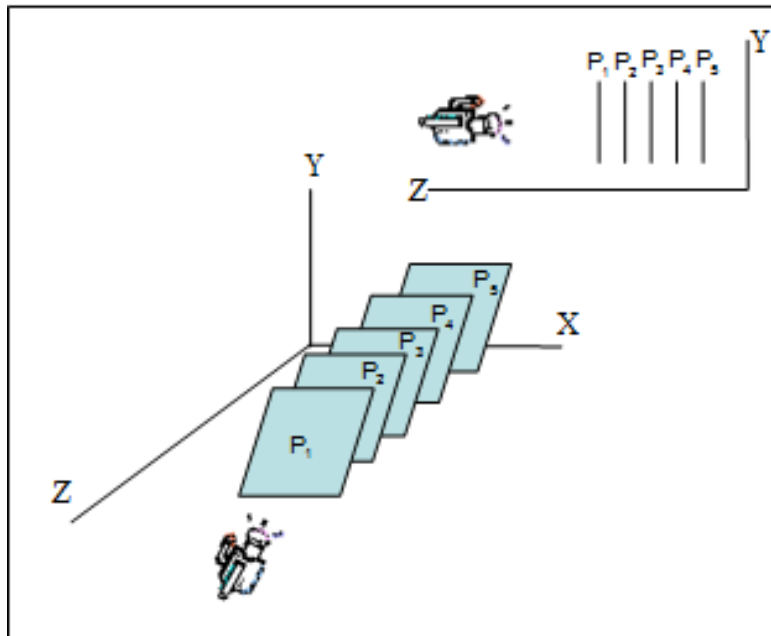


Z-buffer

0.7	0.5	0.5	0.7
0.7	0.2	0.2	0.7
0.7	0.2	0.2	0.7
0.7	0.7	0.7	0.7

# Z-Buffer

10) Supón una cámara ortográfica situada perpendicularmente a los polígonos de una escena, tal y como muestra la siguiente figura, y un algoritmo de Z-Buffer como el siguiente:



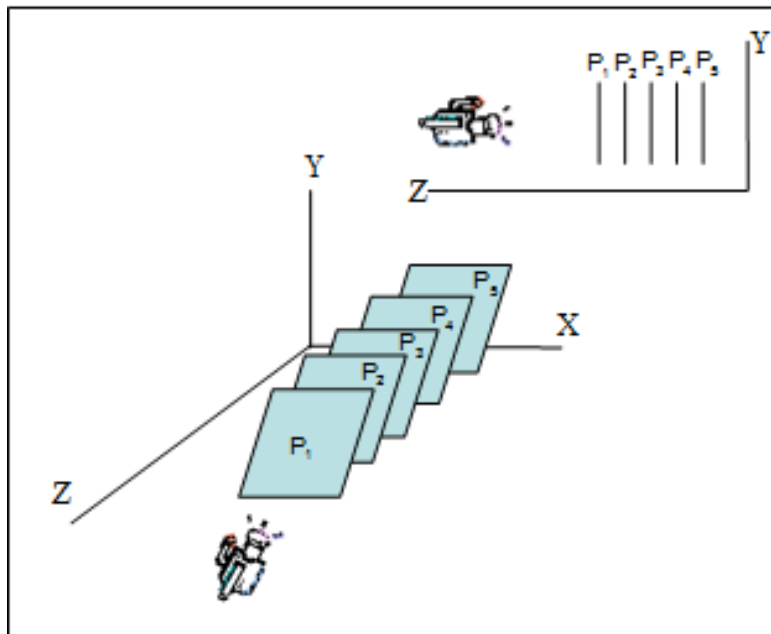
Visibilidad

```

Inicializar imagen a color del fondo
Inicializar zbuffer a la z máxima
para cada polígono p
  para cada pixel en la proyección de p
    z:=z(x,y)
    si z más cercana que zbuffer(x,y)
      zbuffer(x,y) := z
      calcular color (x,y)
      imagen(x,y)=color
    fin si
  fin para
fin para
  
```

# Z-Buffer

- Indica cuándo se produce el caso mejor al dibujar la escena mediante dicho algoritmo. Justifica tu respuesta en función del número de operaciones ejecutadas. Indica así mismo cuál sería el caso peor (es decir, en qué situación el algoritmo tiene que hacer el máximo trabajo para obtener la imagen final).



Visibilidad

```

Inicializar imagen a color del fondo
Inicializar zbuffer a la z máxima
para cada polígono p
    para cada pixel en la proyección de p
        z:=z(x,y)
        si z más cercana que zbuffer(x,y)
            zbuffer(x,y) := z
            calcular color (x,y)
            imagen(x,y)=color
        fin si
    fin para
fin para
  
```



# Z-Buffer

---

- ▶ Para calcular tanto el caso mejor como el caso peor en la ejecución de cualquier algoritmo, hay que fijarse si existe alguna parte del código que pueda no ejecutarse.
- ▶ En el algoritmo propuesto, podemos ver que la ejecución del cuerpo de la instrucción “si” depende de que se cumpla su condición.
- ▶ La operación más costosa de ese cuerpo es “calcular color(x, y)”, por lo que podemos aproximar el coste de la ejecución del algoritmo en función del número de veces que se ejecuta.
- ▶ Sea N el número de píxeles ocupados por la proyección de cualquier polígono.
  - ▶ Caso mejor: Se visualiza P<sub>1</sub>, y después se visualizan los demás polígonos, en cualquier orden. Número de veces que se ejecuta “calcular color”: N
  - ▶ Caso peor: Se visualizan los polígonos en el siguiente orden: P<sub>5</sub>, P<sub>4</sub>, P<sub>3</sub>, P<sub>2</sub>, P<sub>1</sub>. Número de veces que se ejecuta “calcular color”: 5N