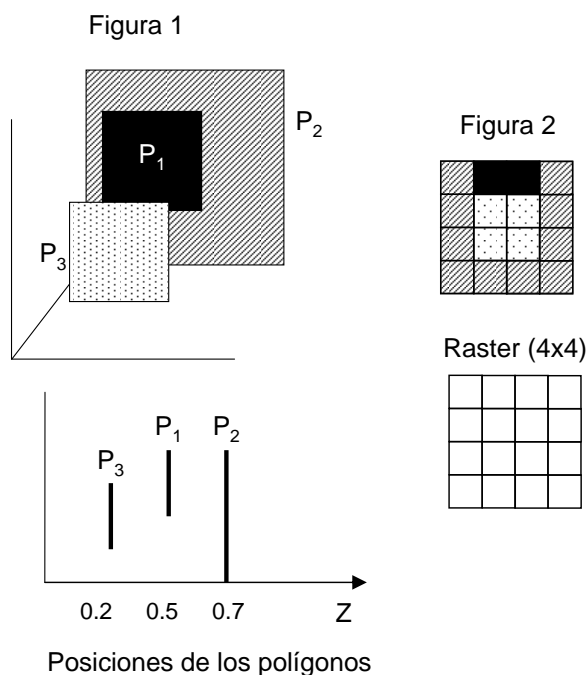


# Tema 7.Visibilidad

## Ejercicios

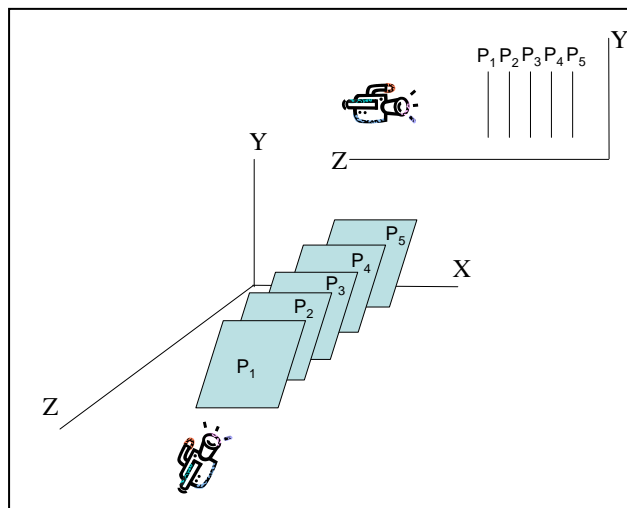
1. ¿Cuáles son las estructuras de datos que requiere el algoritmo del z-buffer y qué se almacena en ellas?
2. Respecto al algoritmo del Z-buffer, ¿Cuál de las siguientes afirmaciones es **falsa**?
  - a) El algoritmo del Z-buffer averigua cuál es el polígono visible a lo largo de una visual
  - b) El Z-buffer es una matriz donde se guarda la profundidad de cada pixel
  - c) En el algoritmo del Z-buffer los polígonos se ordenan según su profundidad en z
  - d) El algoritmo del Z-buffer trabaja con escenas con cualquier número de polígonos
3. Dados dos puntos  $P1(x1,y1,z1)$  y  $P2(x2,y2,z2)$  que se encuentran en una situación de proyección perspectiva simple y cuyas coordenadas son  $P1=(3,2,2)$  y  $P2=(6,4,4)$  ¿puede existir conflicto de visibilidad entre ellos?
  - a) Si, porque están en la misma visual ya que:  $x1/z1=x2/z2$  e  $y1/z1=y2/z2$
  - b) No, porque no están en la misma visual ya que:  $x1/z1=x2/z2$  e  $y1/z1=y2/z2$
  - c) No, porque no están en la misma visual ya que:  $x1 \neq x2$  e  $y1 \neq y2$
  - d) Si, porque están en la misma visual ya que:  $x1/y1=x2/y2$
4. ¿Cuáles de las siguientes características definen mejor el algoritmo del z-buffer para el cálculo de visibilidad?
  - a) Se aplica en el espacio de la imagen, tiene un alto consumo de memoria, funciona con escenas de cualquier complejidad, no requiere ordenación previa de los polígonos
  - b) Se aplica en el espacio de los objetos, tiene un alto consumo de memoria, funciona con escenas de cualquier complejidad, no requiere ordenación previa de los polígonos
  - c) Se aplica en el espacio de la imagen, tiene un bajo consumo de memoria, funciona con escenas de cualquier complejidad, requiere ordenación previa de los polígonos
  - d) Se aplica en el espacio de los objetos, tiene un bajo consumo de memoria, funciona con escenas de cualquier complejidad, no requiere ordenación previa de los polígonos
5. Se desea aplicar la eliminación de los polígonos vueltos de espaldas (caras traseras) previamente a un algoritmo de cálculo de visibilidad. Si el observador se encuentra en  $Z = -\infty$ , mirando en el sentido de las  $Z+$ , ¿qué condición debe cumplir un polígono para ser eliminado?
  - a) Que la normal exterior al polígono forme un ángulo  $\alpha$  con la dirección de visualización de manera que  $|\alpha| < 90^\circ$
  - b) Que la componente z de la normal exterior al polígono sea cero
  - c) Que la normal exterior al polígono forme un ángulo  $\alpha$  con la dirección de visualización de manera que  $|\alpha| > 90^\circ$
  - d) Que el producto escalar entre el vector  $(0,0,1)$  y la normal externa al polígono sea menor que cero
6. ¿Cuál de las siguientes afirmaciones es FALSA?
  - a) Con el algoritmo del z-buffer el cálculo de la visibilidad es correcto independientemente de la ordenación de los polígonos
  - b) El algoritmo del z-buffer se ejecuta durante el proceso de conversión al raster
  - c) El algoritmo del z-buffer sólo funciona correctamente si los polígonos no son penetrantes
  - d) El z-buffer es una matriz que sirve para almacenar coordenadas z

7. ¿Qué condición habrían de cumplir los polígonos de un objeto cerrado, para ser eliminados trivialmente en el proceso de ocultación, si el observador se halla en  $z = -\infty$ ?
- Que la coordenada de alguno de sus vértices tenga  $z < 0$ .
  - Que el polígono sea el más alejado del observador.
  - Que el polígono esté totalmente tapado por un objeto más cercano.
  - Que la normal exterior al polígono tenga la componente  $z$  positiva.
8. ¿Qué problema se plantea cuando se aplica el algoritmo del Z-Buffer a polígonos penetrantes (un polígono que atraviesa a otro)?
9. 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  $P_3$ .



	Frame buffer	Z-buffer																																
Inicialización	<table><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr></table>																	<table><tr><td>1</td><td>1</td><td>1</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td><td>1</td></tr></table>	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1																															
1	1	1	1																															
1	1	1	1																															
1	1	1	1																															
Después de P1	<table><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr></table>																	<table><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr></table>																
Después de P2	<table><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr></table>																	<table><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr></table>																
Después de P3	<table><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr></table>																	<table><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr></table>																

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:



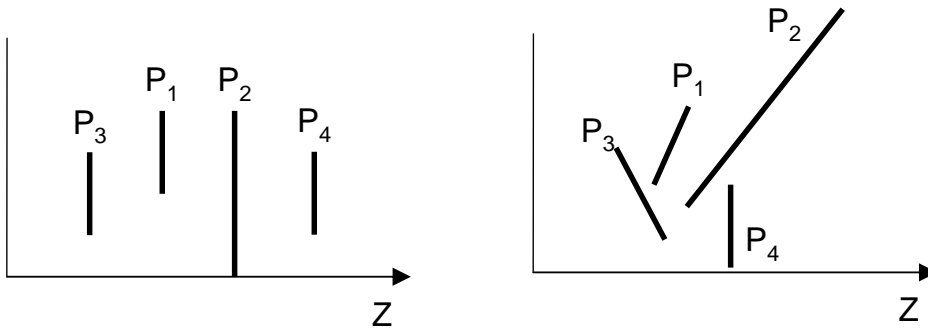
```

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

```

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).

11. ¿En cuál de las siguientes escenas funcionaría más rápido el algoritmo del pintor? Justifica tu respuesta.



12. Dados los polígonos de la figura, ¿qué problema plantea la aplicación del algoritmo del pintor para resolver la visibilidad? ¿y el algoritmo del z-buffer?

