

Algoritmo de Warnock



Jaime Andres Vargas Arevalo

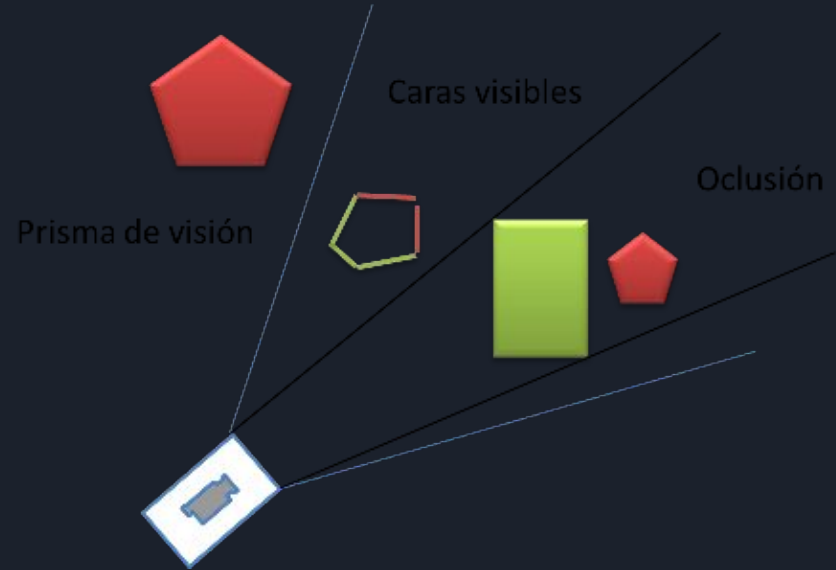


Que se quiere hacer

- Entender e implementar el algoritmo warnock
- Utilización del algoritmo “divide y vencerás”
- Ilustrar el algoritmo con rectángulos random en sus 3 vistas(arriba, abajo ,lado derecho)

Estado del arte

- Métodos de renderización y ray-tracing.
- Algoritmos HSR (Hidden Surface removal).
- Algoritmos como:
 - Pintor
 - BSP(Binary Space Partitioning)
 - Z-Buffer
 - Scan line
 - ray-tracing para superficies poliédricas y cuádricas

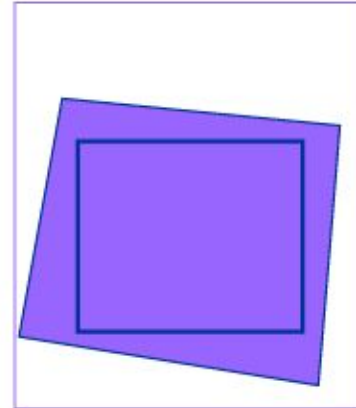
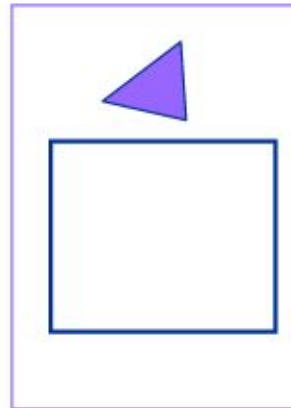
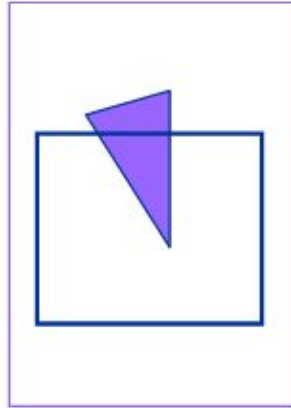
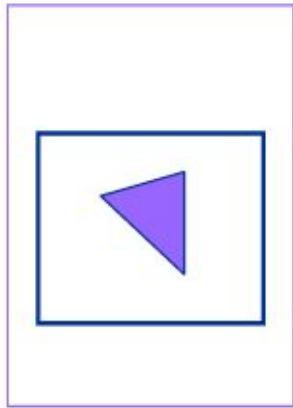




Algoritmo de Warnock

- Algoritmo “divide y vencerás”.
- Divide en 4 áreas iguales hasta llegar a casos bases
- Da una mejor visibilidad por cada cuadrante que llegó a subdividirse
- Se considera que es un algoritmo que tiene precisión de imagen
- Se utiliza en polígonos convexos.

Relaciones entre polígonos y elementos de área





Pseudocódigo

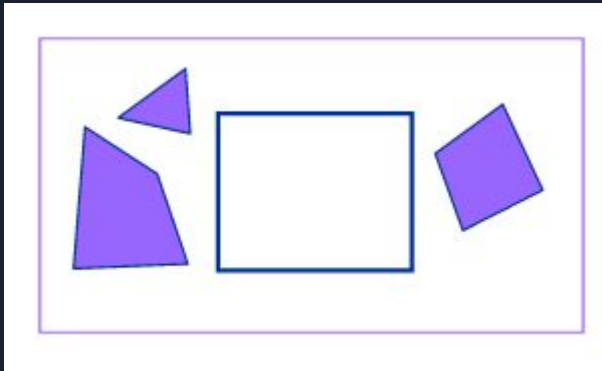
```
Warnock(PolyList PL, ViewPort VP)
If ( PL simple in VP) then
    Draw PL in VP
else
    Split VP vertically and horizontally into VP1,VP2,VP3,VP4
    Warnock(PL in VP1, VP1)
    Warnock(PL in VP2, VP2)
    Warnock(PL in VP3, VP3)
    Warnock(PL in VP4, VP4)
end
```

$O(np)$

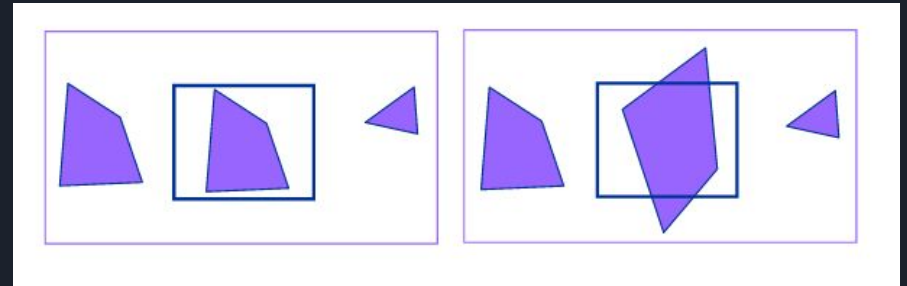
n: #poligonos
p: #pixeles

Casos bases

1. Disjuntos

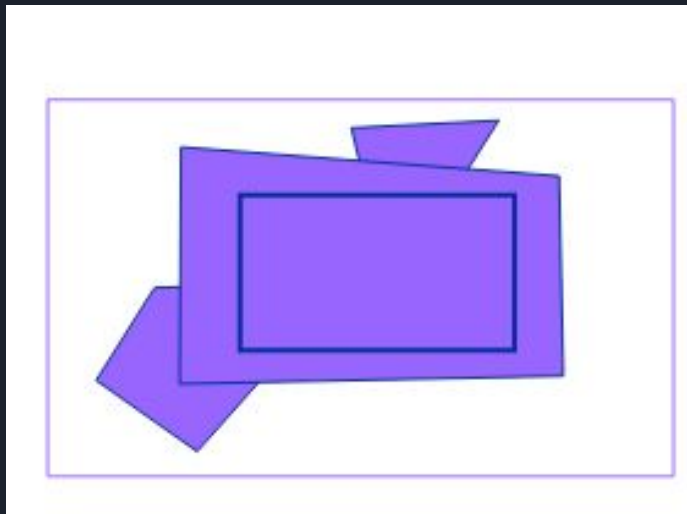


2. contenido o intersección

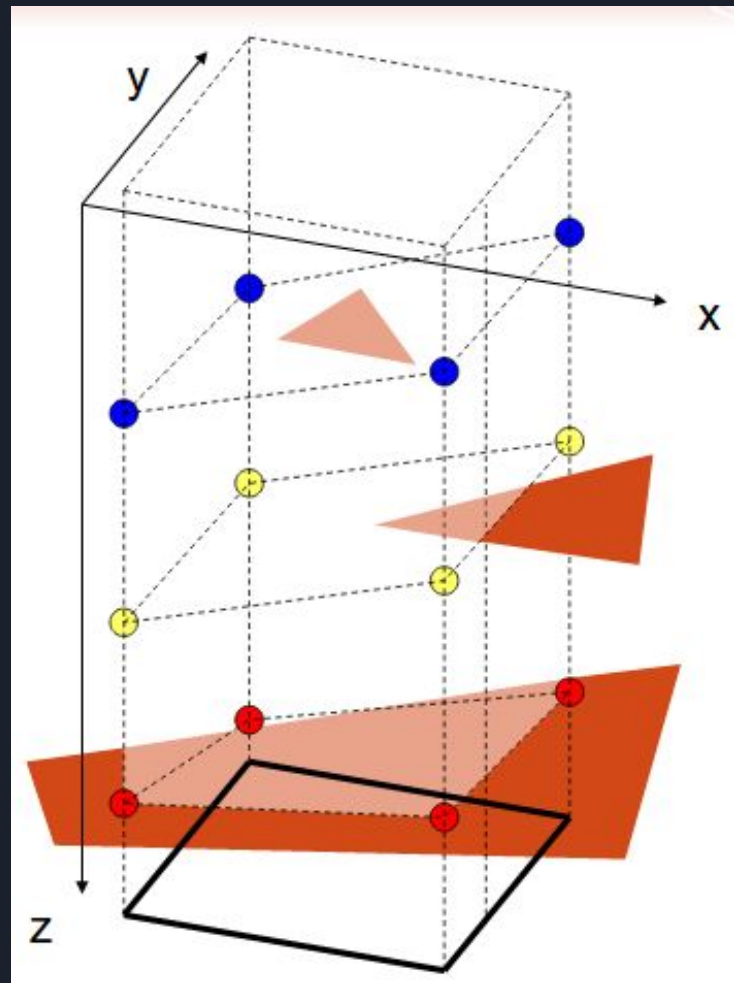




3. Dentro

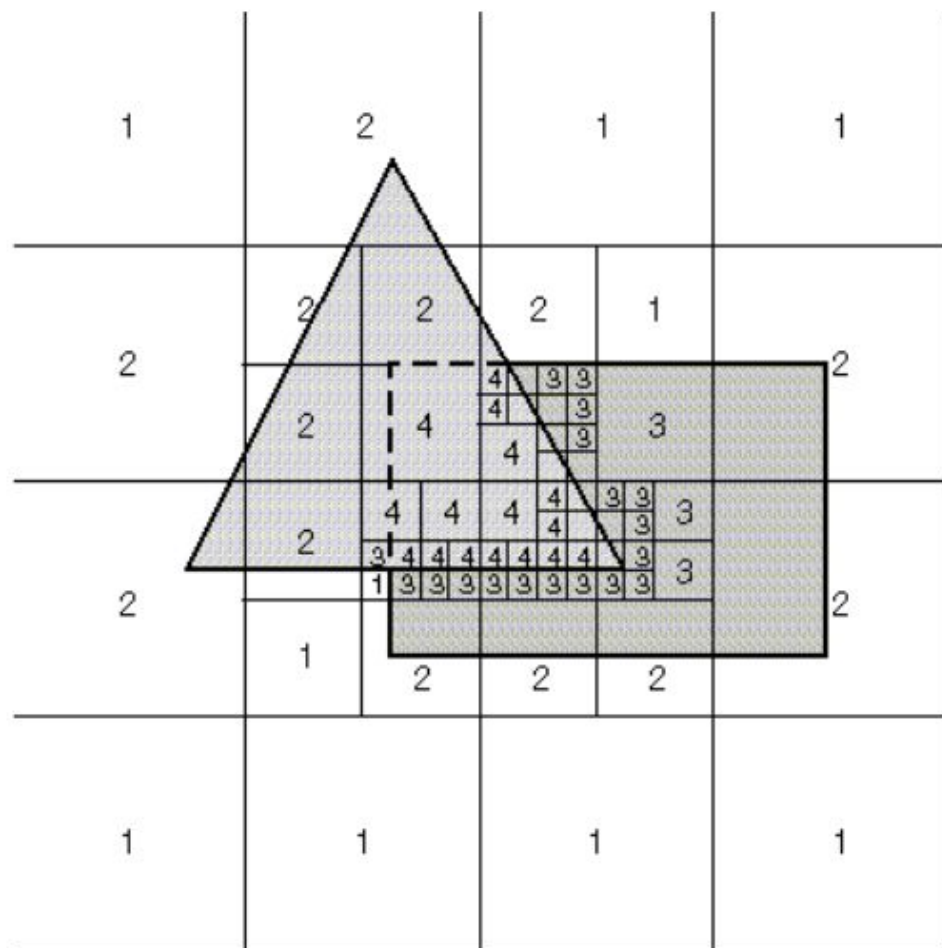
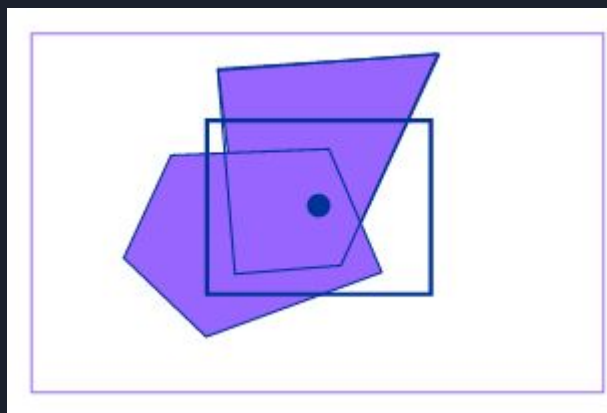


4. Eje Z

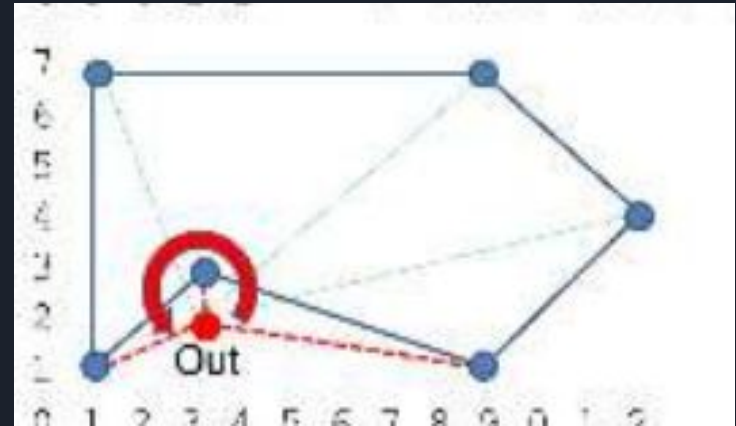
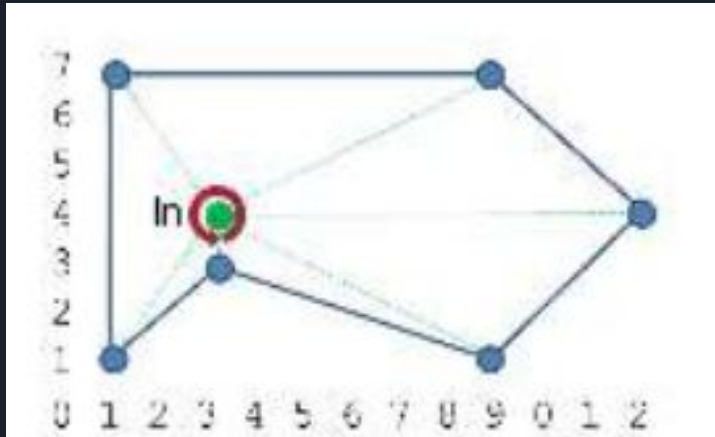




Caso especial

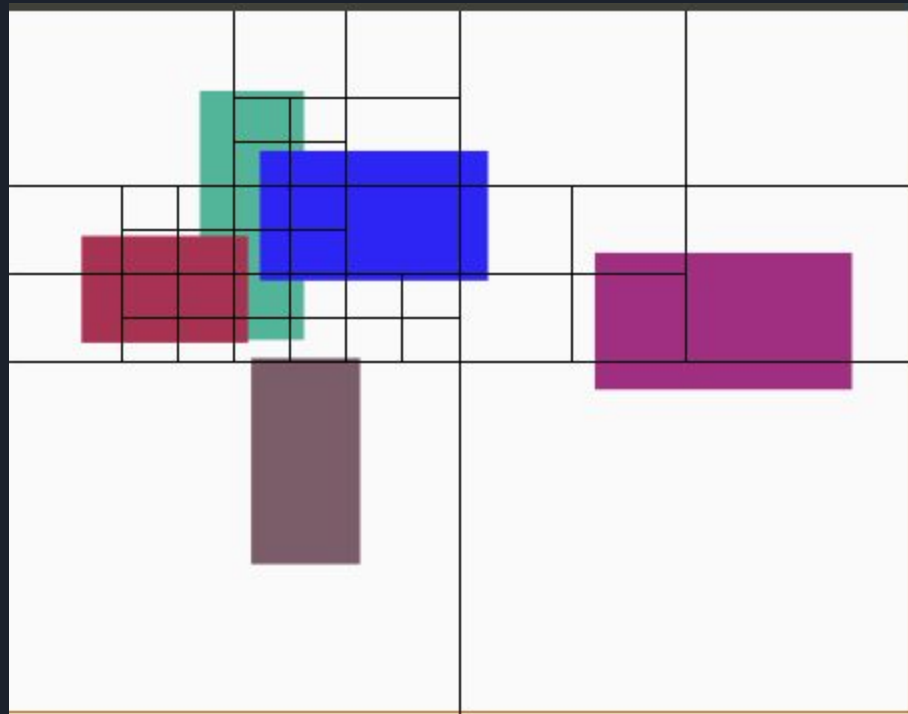
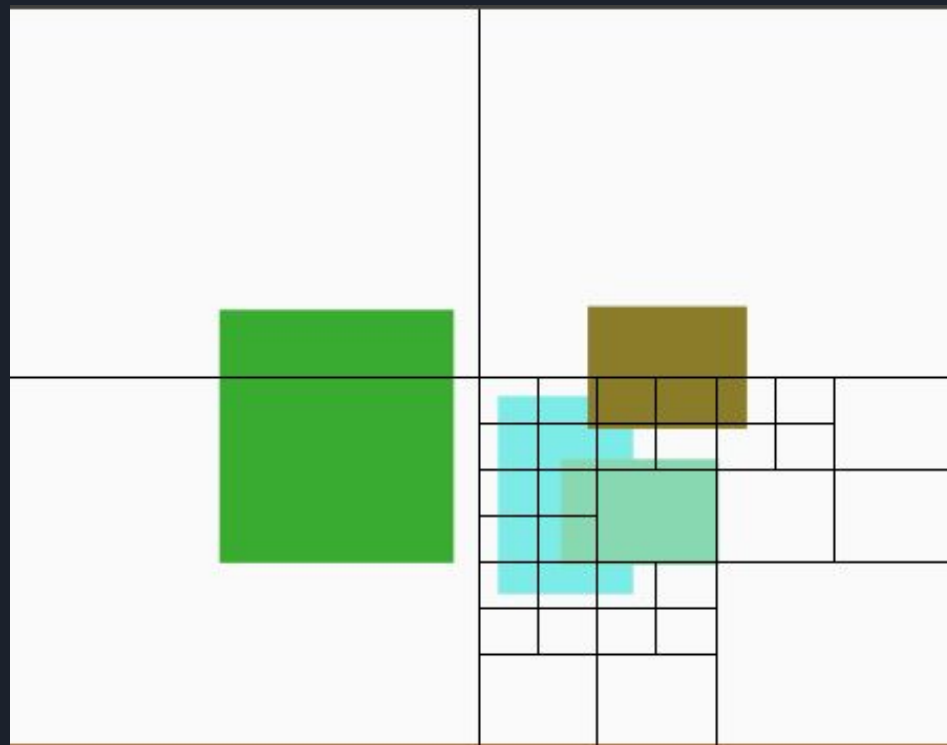


Winding number algorithm



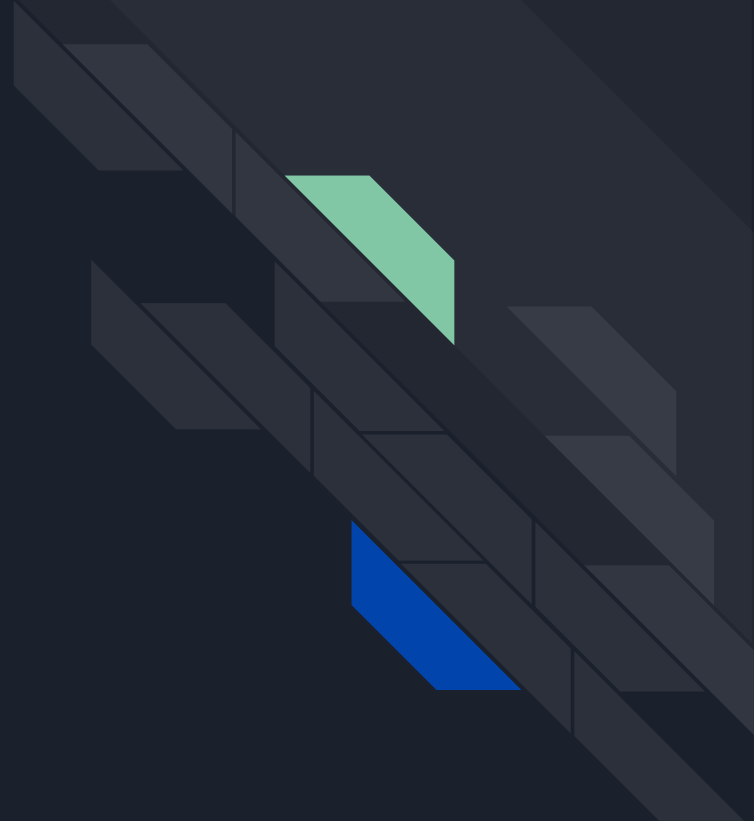
Si la suma final es 2π (360 grados),
entonces el punto pt está dentro del
polígono P

```
warnockAlgorithm( x1+(y1-x1)/2,y1,y2,x2+(y2-x2)/2);  
warnockAlgorithm( x1,x1+(y1-x1)/2,y2,x2+(y2-x2)/2);  
warnockAlgorithm(x1,x1+(y1-x1)/2,x2+(y2-x2)/2,x2);  
warnockAlgorithm(x1+(y1-x1)/2,y1,x2+(y2-x2)/2,x2);
```



Demo

https://github.com/jaavargasar/jaavargasar.github.io/tree/master/warnock_v3





Conclusiones

- Implementación e ilustración del algoritmo warnock utilizando rectángulos superpuestos.
- Poder ver las vistas de la escena para entender de mejor manera como funciona el algoritmo warnock.
- Futuro:
 - Implementación con P3D (mejor visualización tridimensional de la escena)
 - Poder utilizar cualquier polígono convexo
 - Optimización de la implementación



Referencias

- University of Waterloo. Computer Graphics Lab. Warnocks' Algorithm, desde <http://medialab.di.unipi.it/web/IUM/Waterloo/node68.html>
- herpes. Algoritmo de warnock, desde <http://www16.wikipedes.eu/03500965/AlgoritmoDeWarnock>
- Computación gráfica. Eduardo Fernandez, desde <https://www.fing.edu.uy/inco/cursos/compgraf/Clases/2012/13-Superficies%20Visibles.pdf>
- [Competitive Programming 3, The New Lower Bound of programming Contests.2013. Steven and Felix Halim.](#)
- Superficies visibles, Prof Fernandez(Universidad de la republica de Uruguay), desde https://esaulgd.files.wordpress.com/2012/10/07_superficiesvisibles_p2.pdf
- Visualizacion y realismo, Carlos Ureñas Almagro, Curso 2011-12, desde <https://lsi.ugr.es/curena/doce/vr/transpa/11-12/vr-c02-impr.pdf>
- A Hidden Surface Algorithm for Computer Generated Halftone Pictures. Warnock John, pdf tomado de <http://www.dtic.mil/docs/citations/AD0753671>



Gracias