



IA para Videojuegos con JS

(y Phaser)



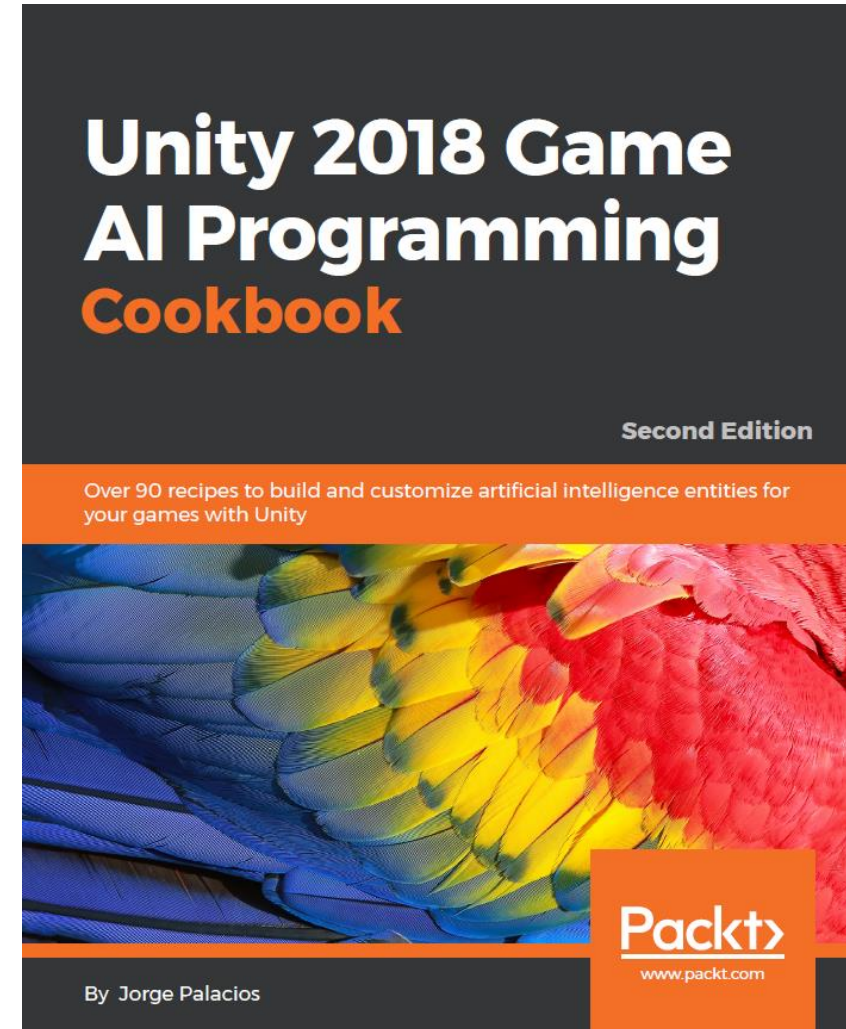
Jorge Palacios
Noviembre 2018

Un poco sobre mí

- ▶ Lic. En Computación
- ▶ Estudios de Maestría
- ▶ Autor

- ▶ Baterista
- ▶ Rocket League adicto

- ▶ Me encanta enseñar y compartir



Experiencia



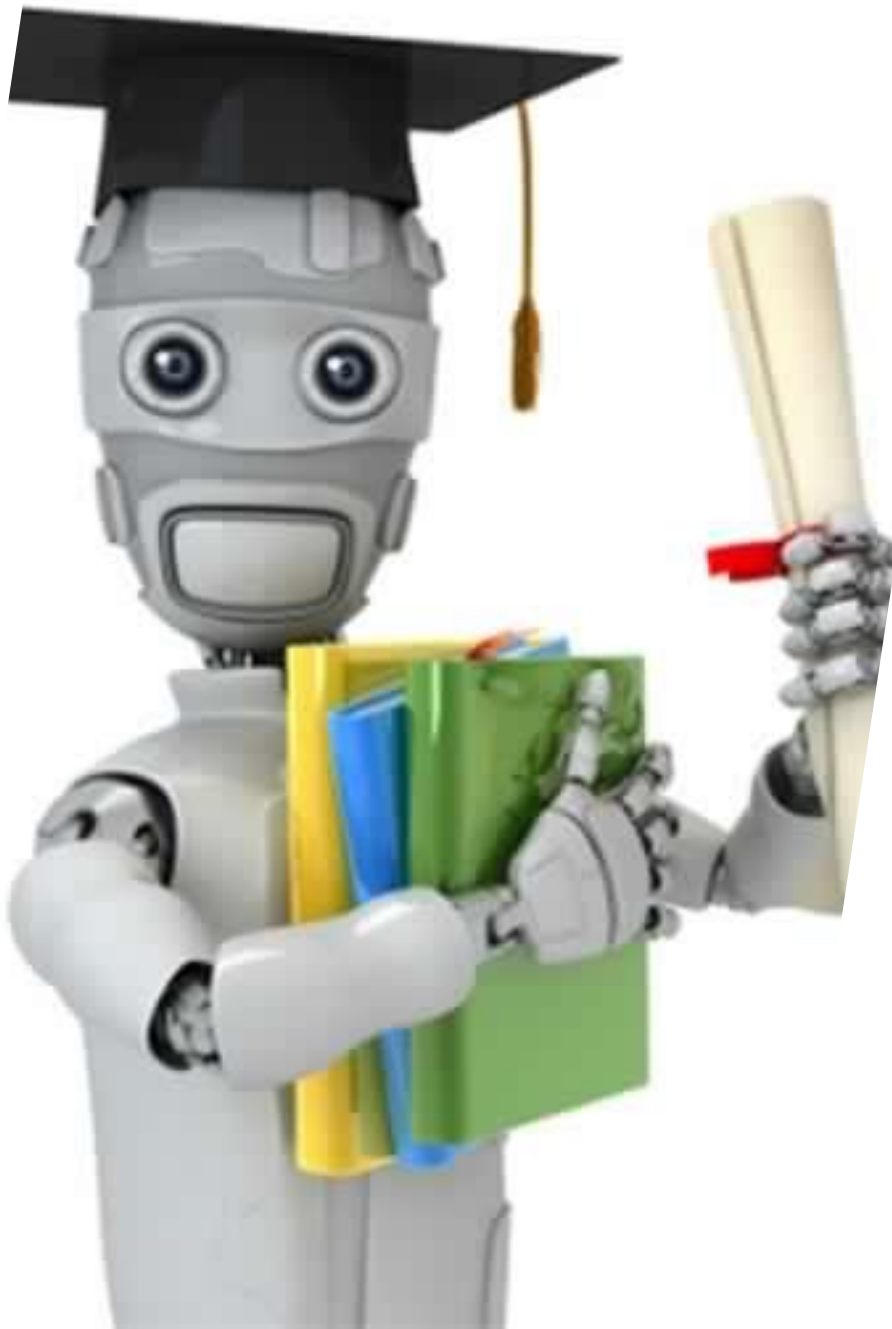
TERRAVISION GAMES



Apress®

¿Qué es la Inteligencia Artificial?





Inteligencia Artificial en la Academia

- ▶ Resultados o simulaciones “perfectas”
- ▶ Ambientes o casos “exagerados”
- ▶ Investigación científica enfocada en expandir las barreras



Inteligencia Artificial en los Videojuegos

- ▶ Eficiencia > precision
- ▶ Ilusión/sensación > realismo
- ▶ HACK THE PLANET!!
(soluciones baratas)

 Will it
BLEND?



¡Sí!

- ▶ Competencias de IA sobre juegos en eventos académicos
- ▶ Eventos online propuestos por personalidades en la academia





Darius Kazemi

@tinysubversions · Feb 27



Replying to @togelius

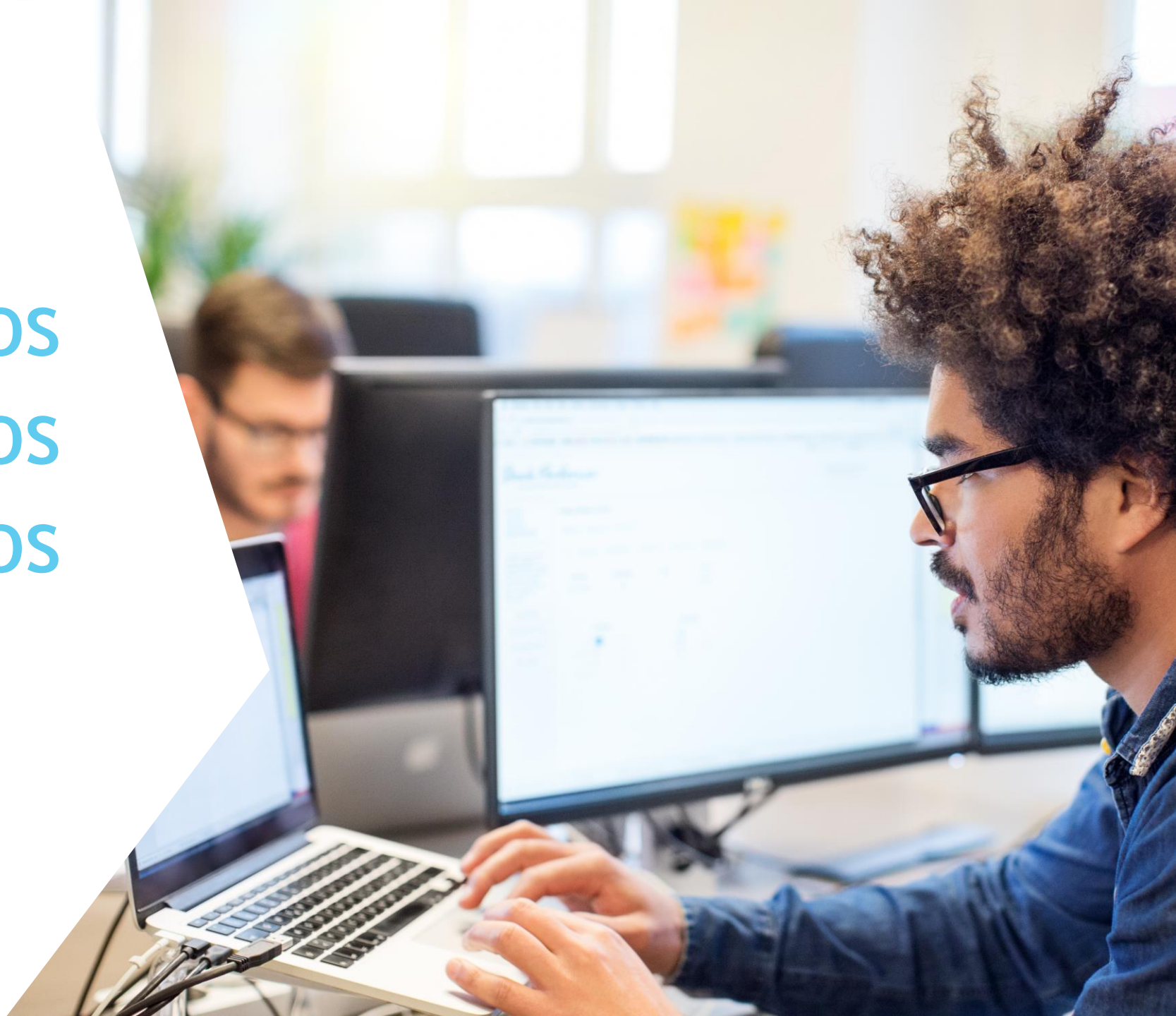
The best advice I ever got for designing artificial intelligence algorithms for games was: if the agent has 3 choices, start with $X = \text{RND}(3)$. It might be good enough AI and you can move on to other things.

La falacia de la complejidad



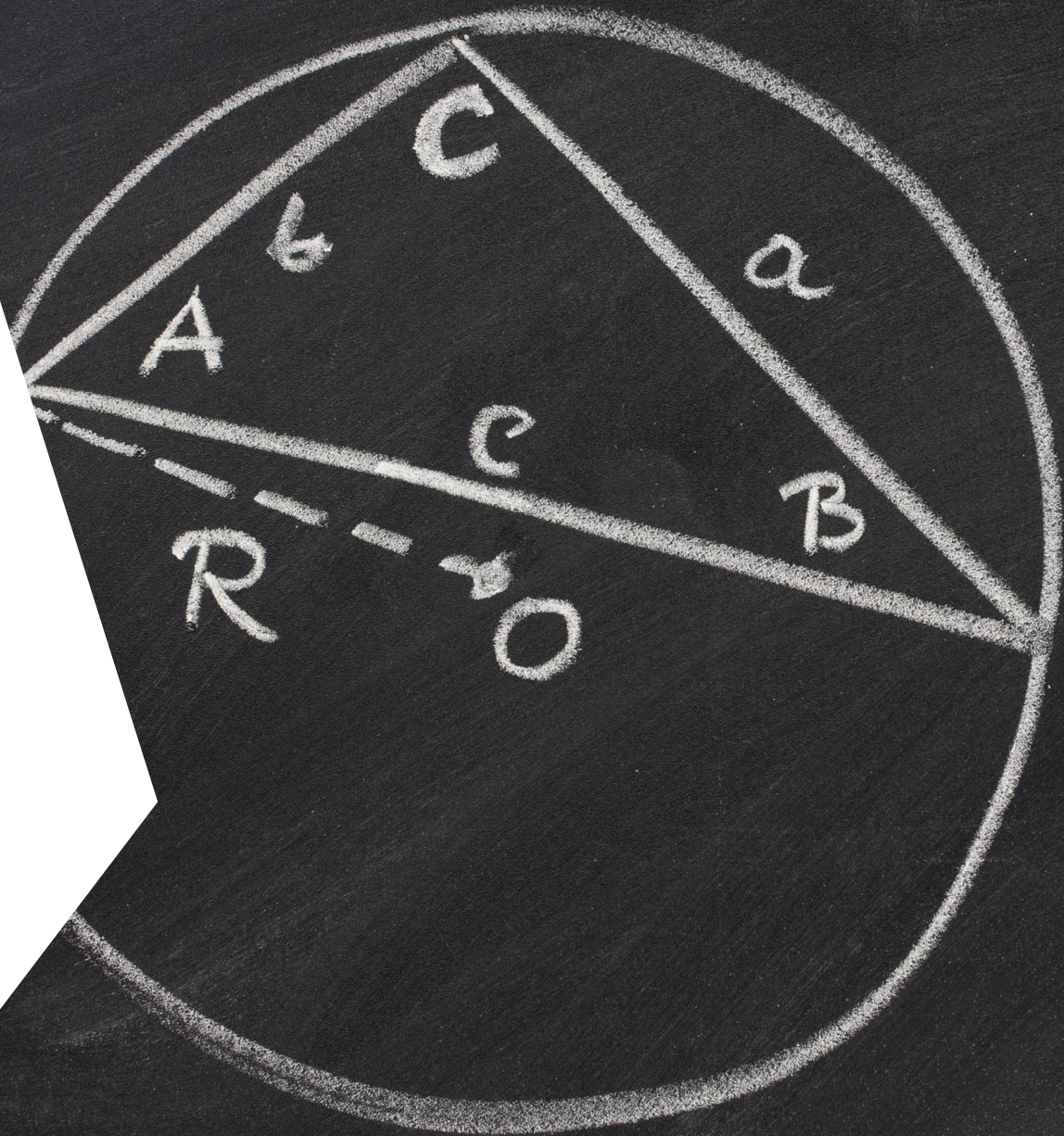
HALO 2

Algunos fundamentos técnicos



Trigonometría

- ▶ Seno
- ▶ Coseno
- ▶ Tangente
- ▶ Cotangente



$$\frac{a}{\sin A}$$
$$\frac{b}{\sin B}$$
$$\frac{c}{\sin C}$$
$$2$$



Hablemos de movimiento

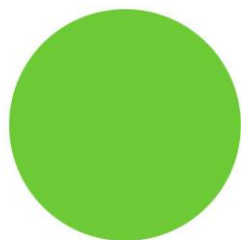
Álgebra lineal

- ▶ Vectores
- ▶ Suma y resta de Vectores
- ▶ Vector unitario y multiplicación por escalar
- ▶ Producto punto
- ▶ etc

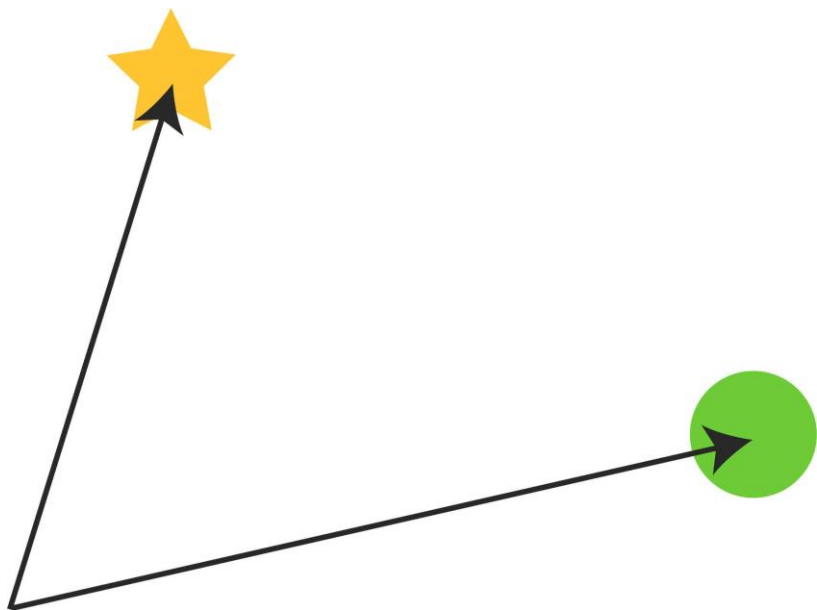
Saquemos
cuentas



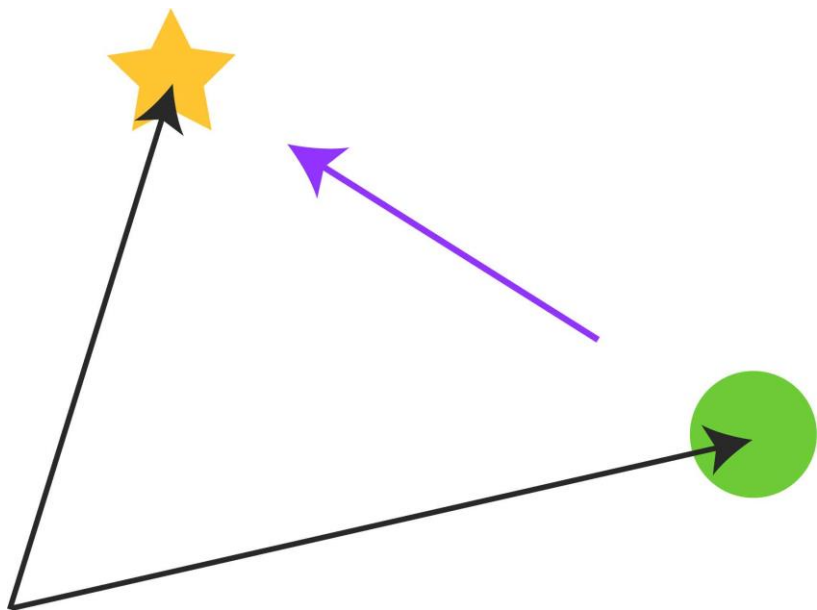
Saquemos
cuentas



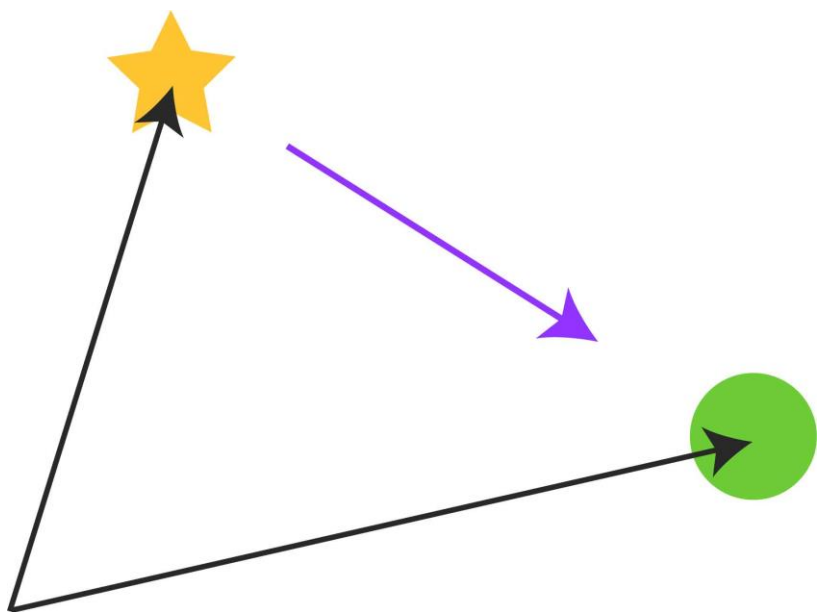
Saquemos
cuentas



Saquemos
cuentas



Saquemos
cuentas

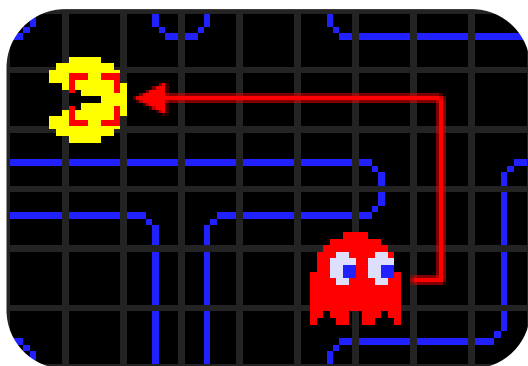




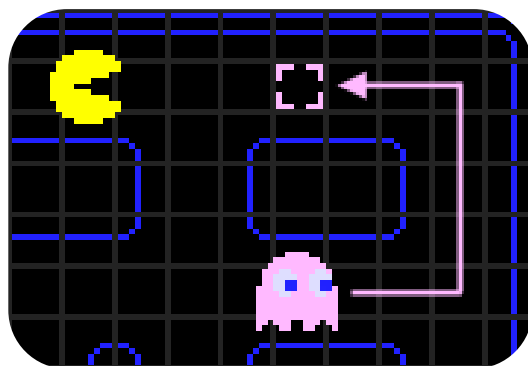


Hablemos de Pac-man

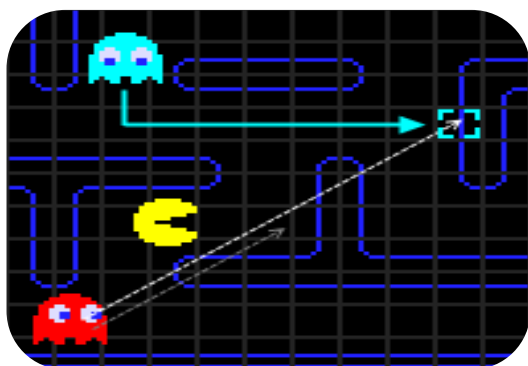




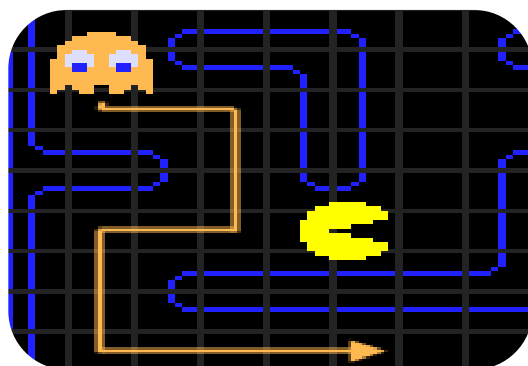
PERSEGUIR



PREDECIR



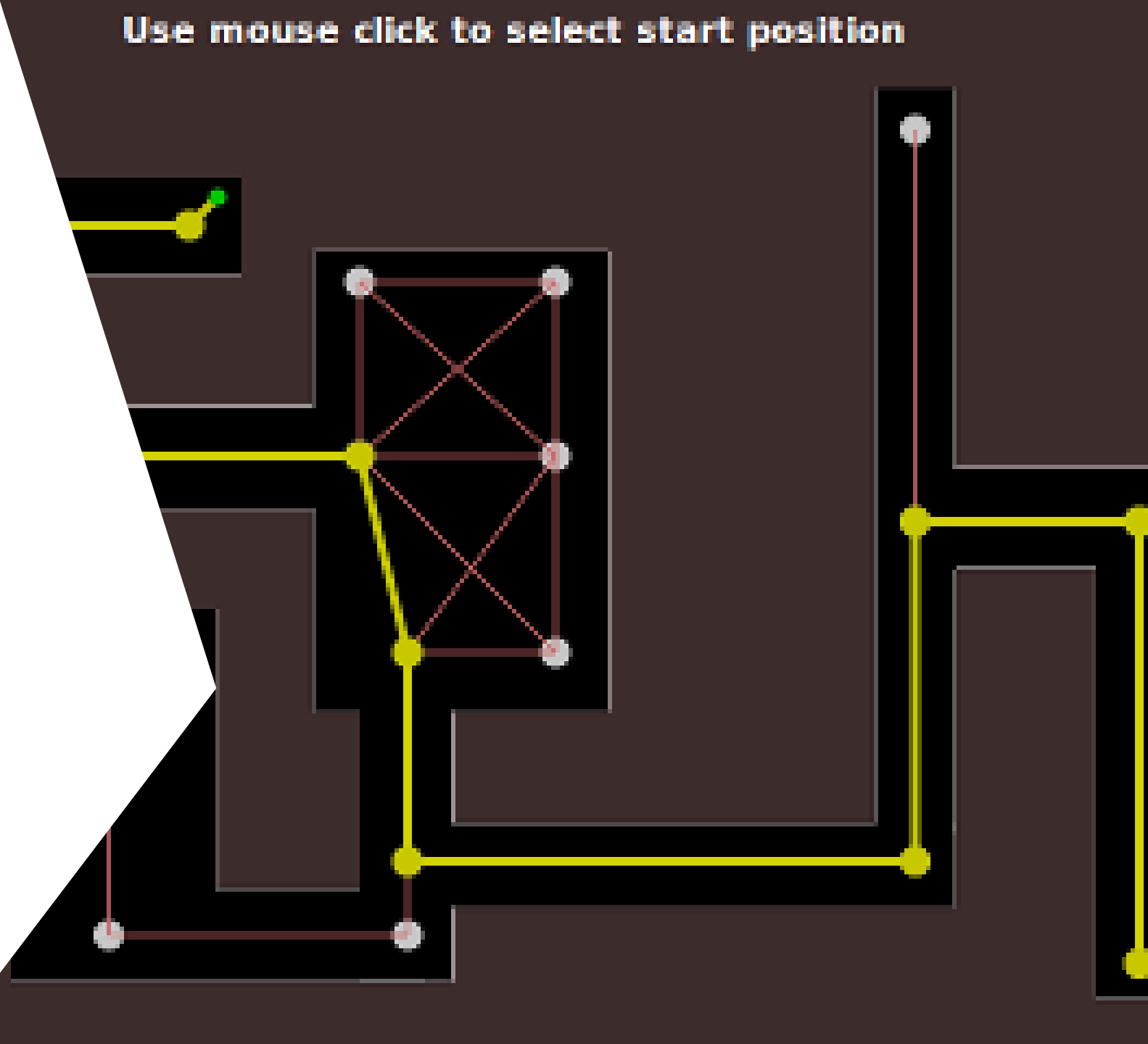
EMBOSCAR

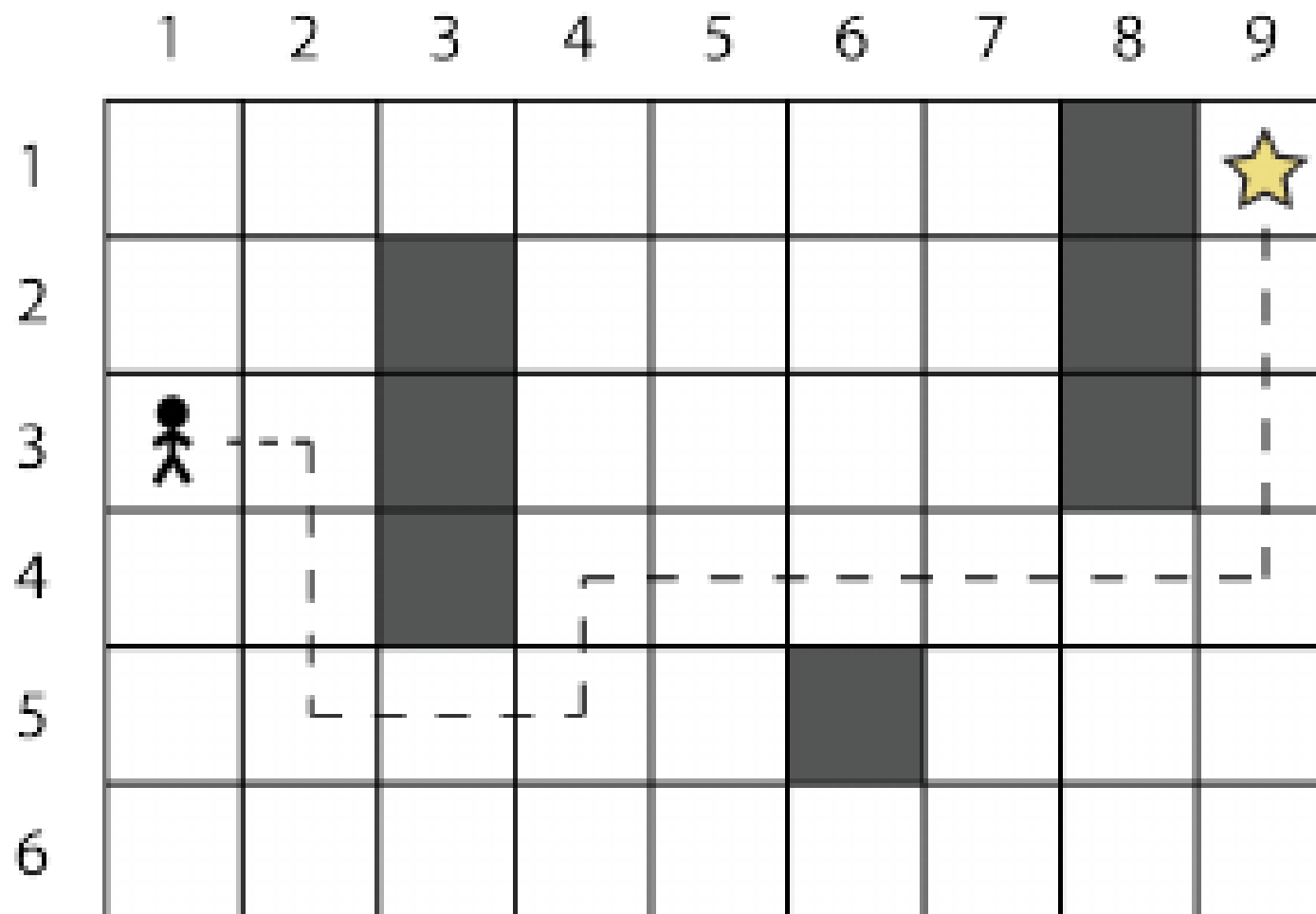


MERODEAR

Grafos

- ▶ Representación en memoria
- ▶ Algoritmos de búsqueda
- ▶ BFS
- ▶ Dijkstra
- ▶ A*





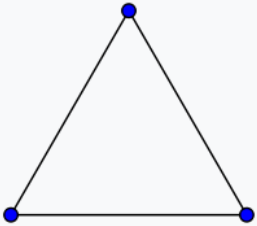
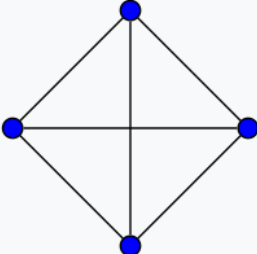
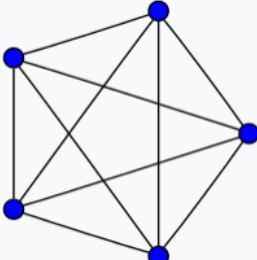
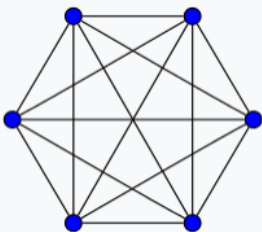
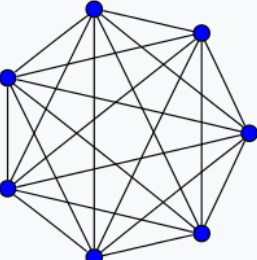
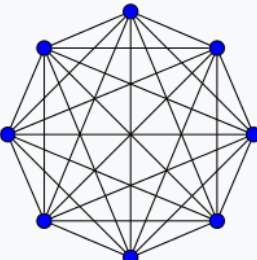
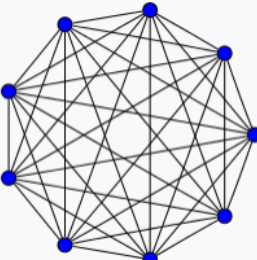
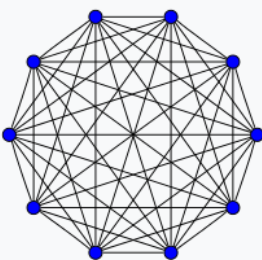
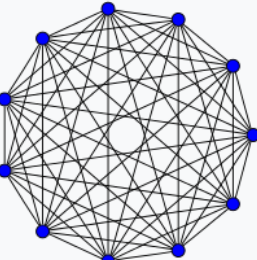
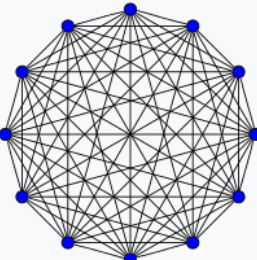


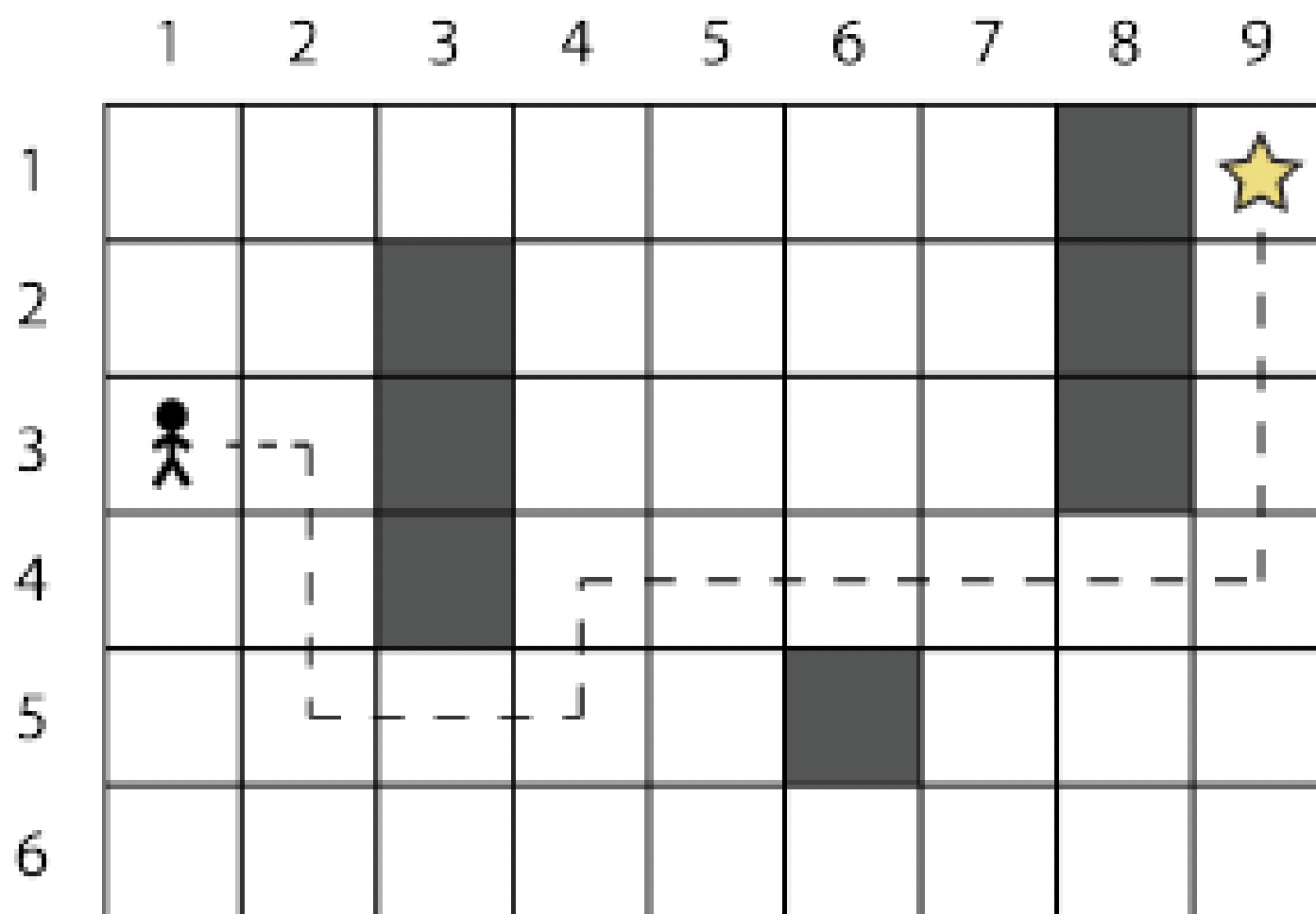
Arreglo 2D

	A	B	C	D
A	1	3	1	Nil
B	2	1	Nil	Nil
C	7	3	1	2
D	3	Nil	2	2

Grafo completo

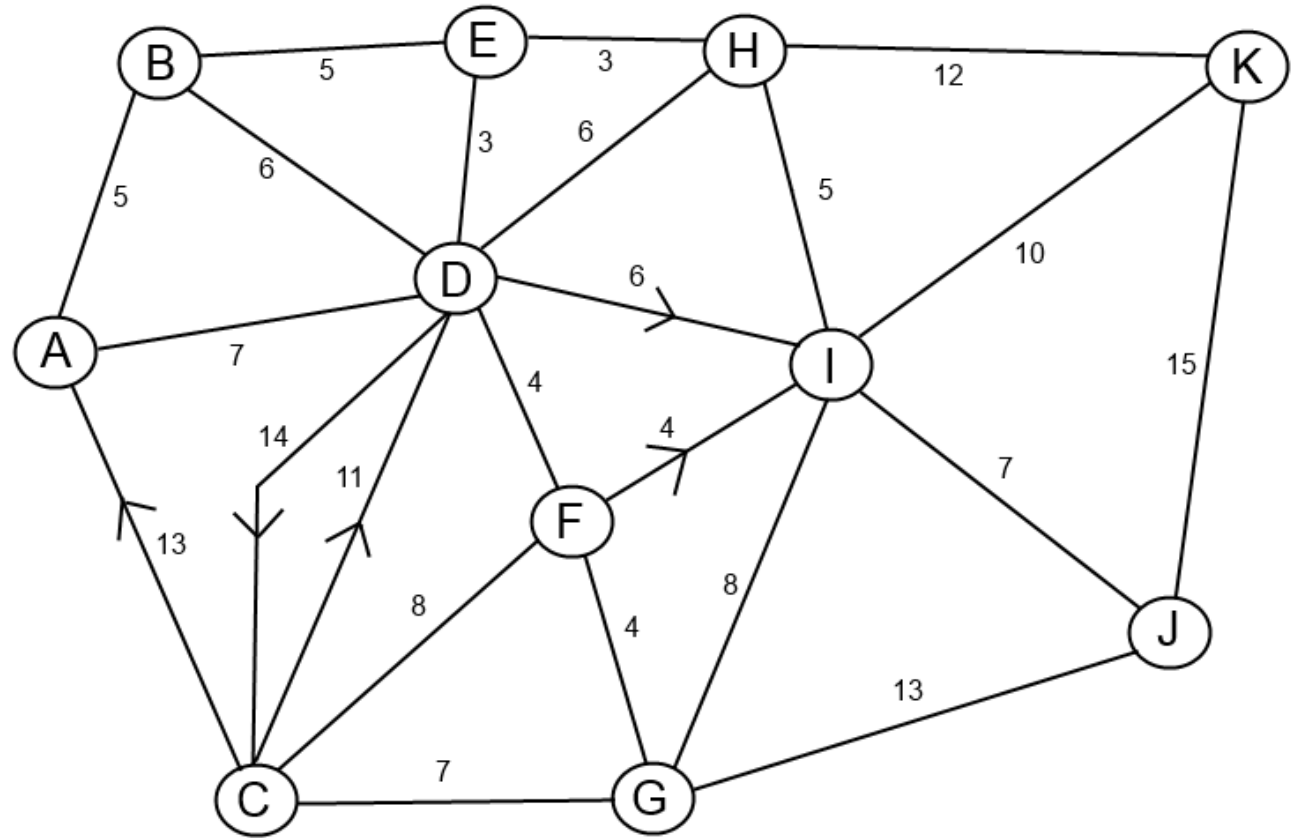
- ▶ El arreglo 2D tiene sentido
- ▶ Todos los nodos conectan entre sí

$K_1: 0$	$K_2: 1$	$K_3: 3$	$K_4: 6$
			
$K_5: 10$	$K_6: 15$	$K_7: 21$	$K_8: 28$
			
$K_9: 36$	$K_{10}: 45$	$K_{11}: 55$	$K_{12}: 66$
			

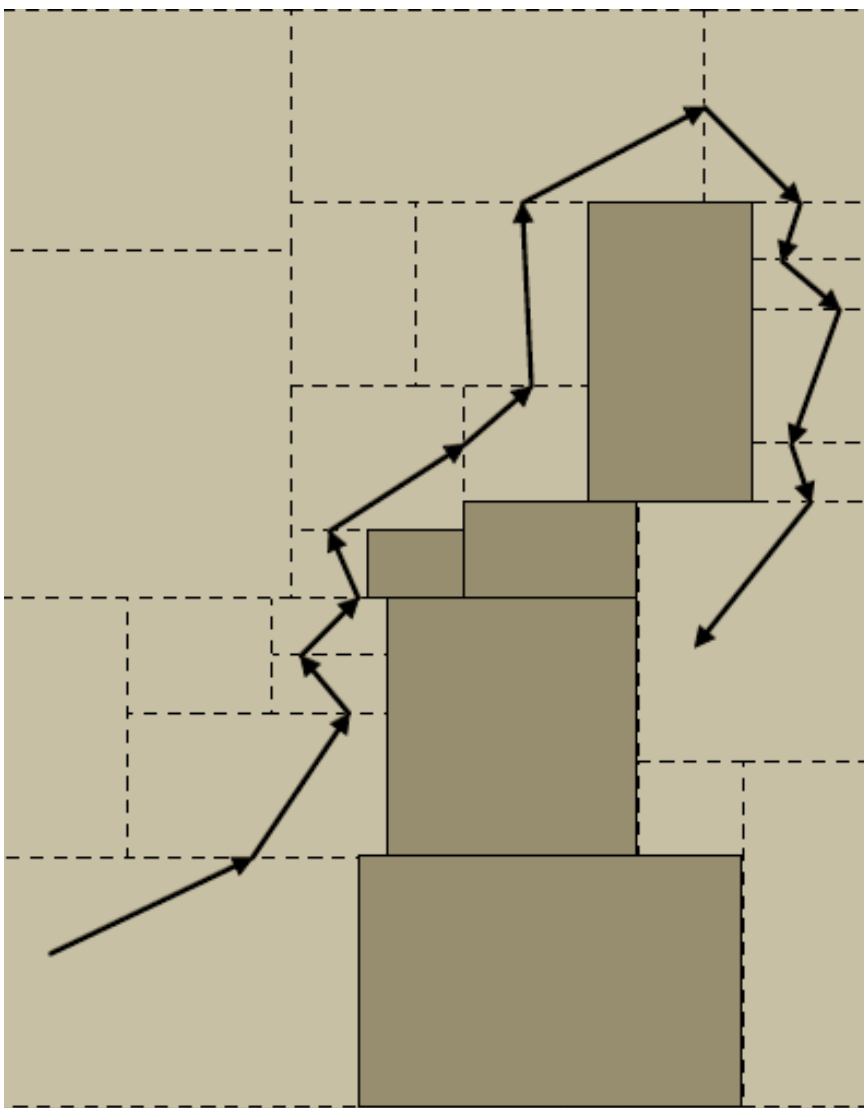


¿Y en este caso?

- ▶ Arreglo de “listas”
- ▶ Más eficiente en memoria



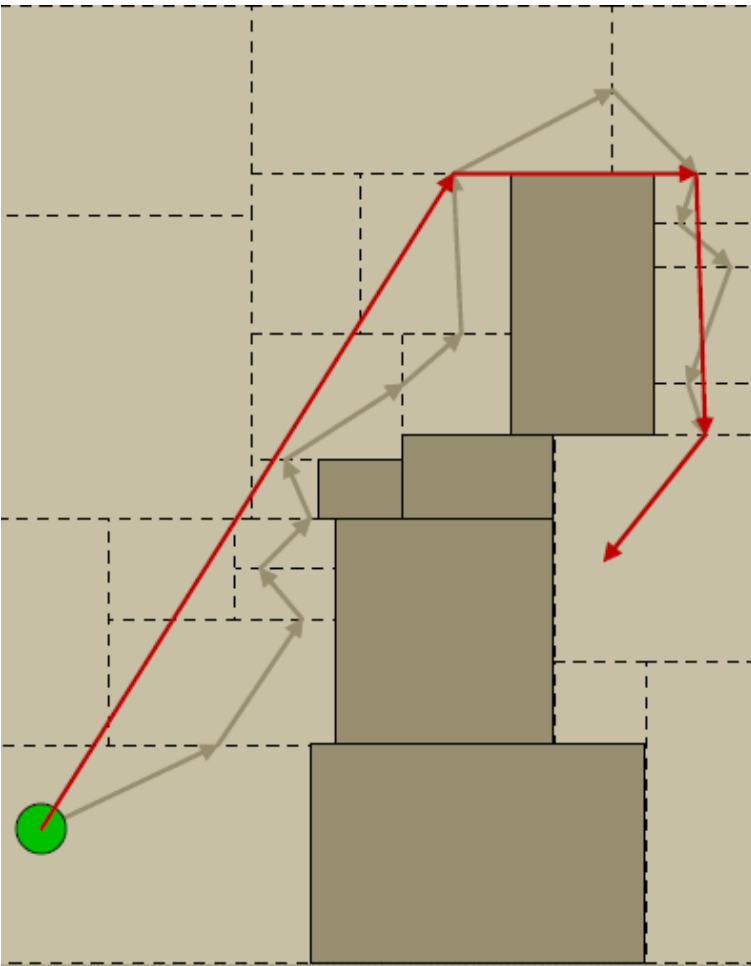




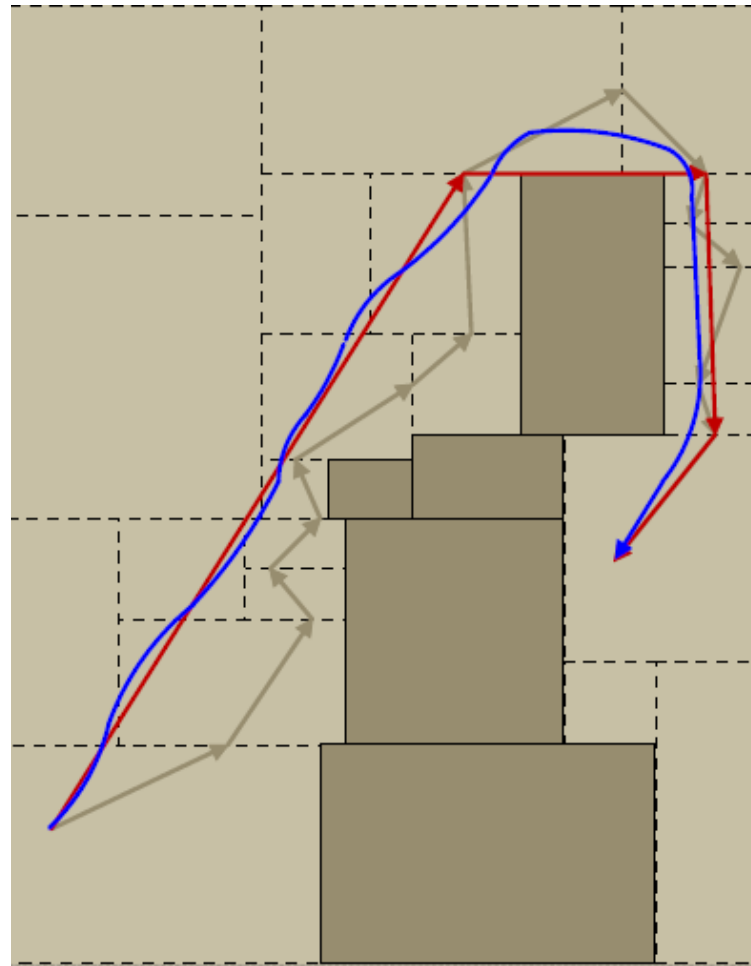
Búsqueda de
caminos en L4D

Búsqueda de caminos en L4D

Optimización



Suavizado



B

S

P



B

Binary

S

Space

P

Partitioning

BSP en pocas palabras



Tomar el
área inicial y
picarla en 2

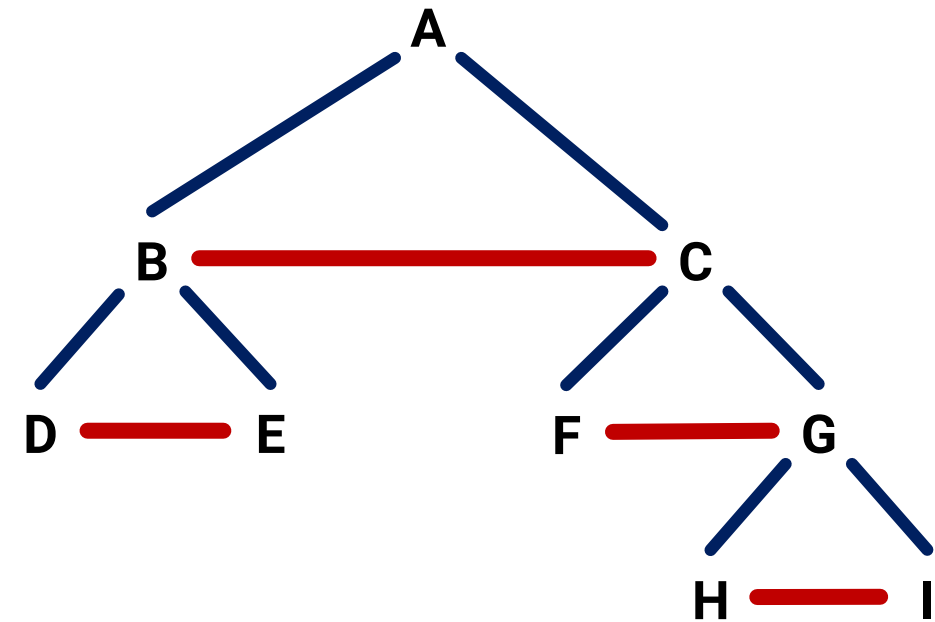
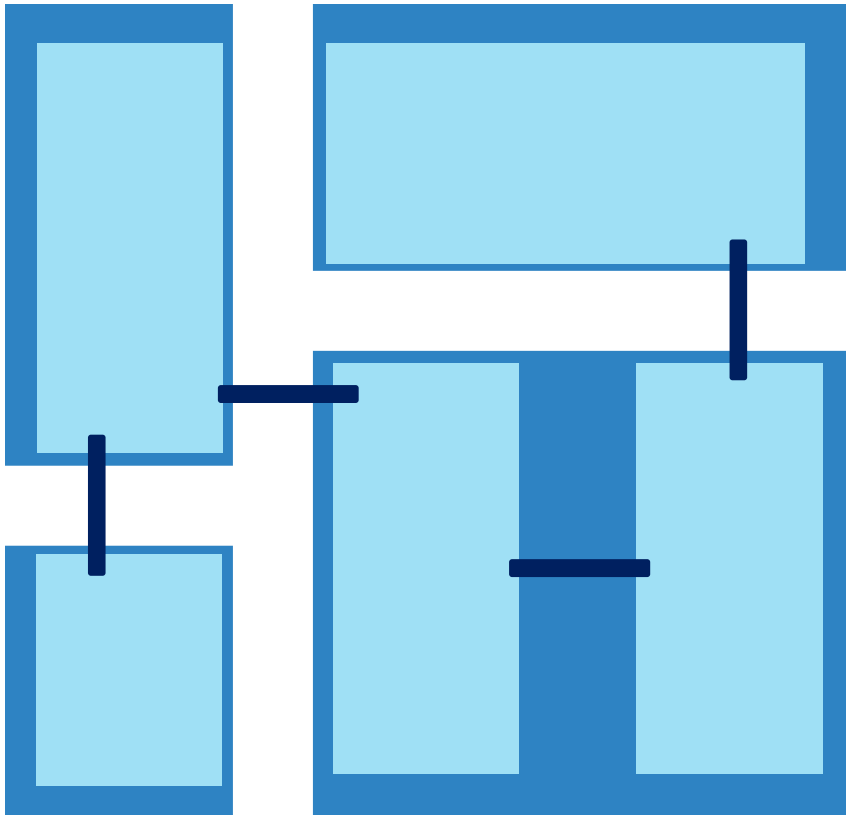


Seguir
picando
(si es
necesario)

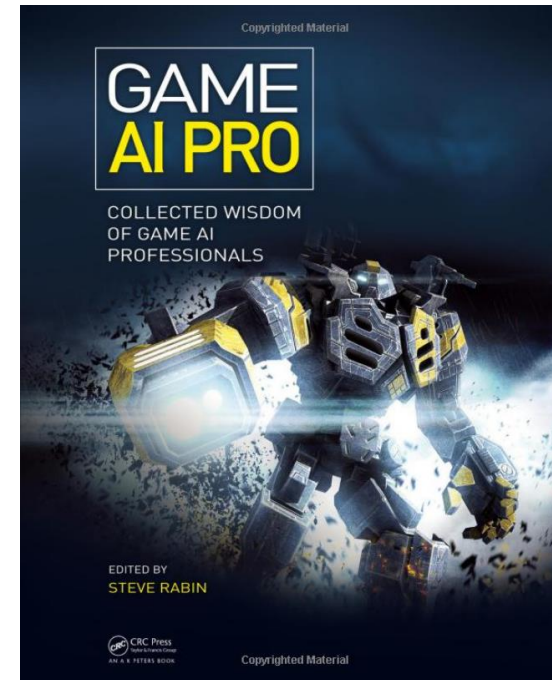
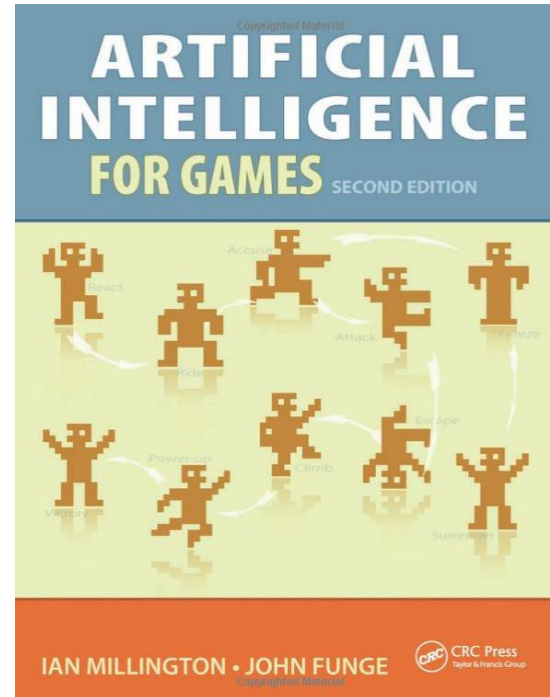
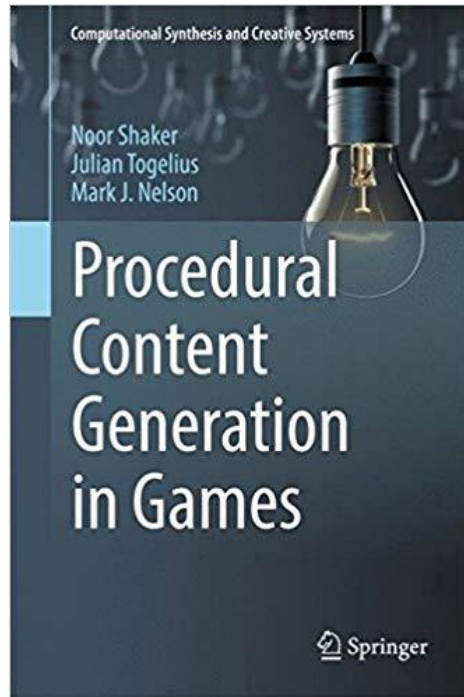


Conectar
las celdas
usando
pasillos

Binary Space Partitioning







¿Dónde comenzar?



Material en la web

- ▶ AI and Games (YouTube)
- ▶ aigamedev.com
- ▶ pcgbook.com



MUCHAS GRACIAS

- ▶ @pctroll (Twitter)
- ▶ jorge.palacios.rocks
- ▶ Código disponible en:
<https://github.com/pctroll/jsconfco2018>