USPGameDev

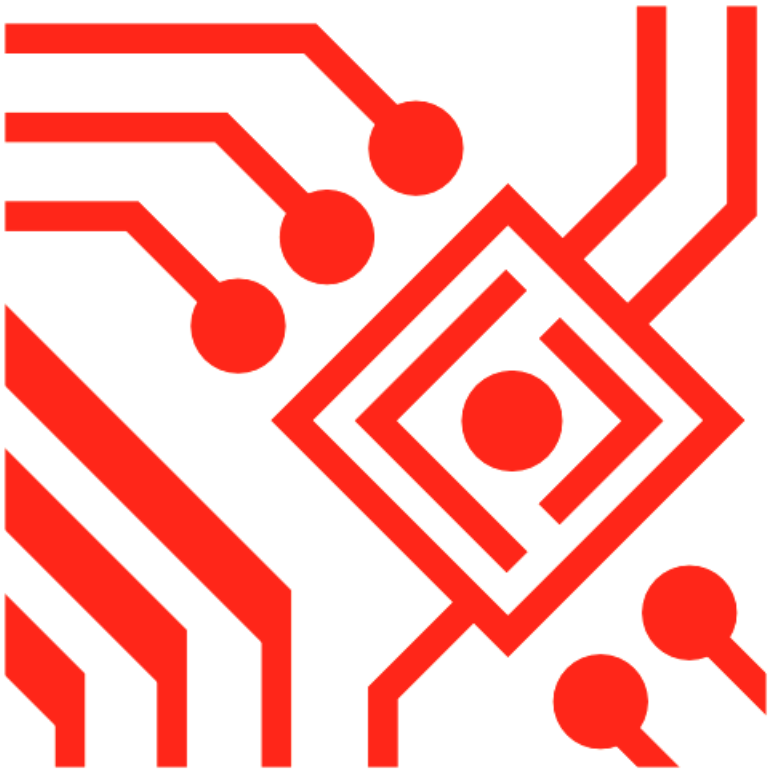
Desafios de Computação no Desenvolvimento de Jogos

Vinícius K Daros + Wilson Kazuo Mizutani
www.uspgamedev.org

Encontro BCC
12/09/2014

Quem somos

Jogos digitais



Estrutura básica

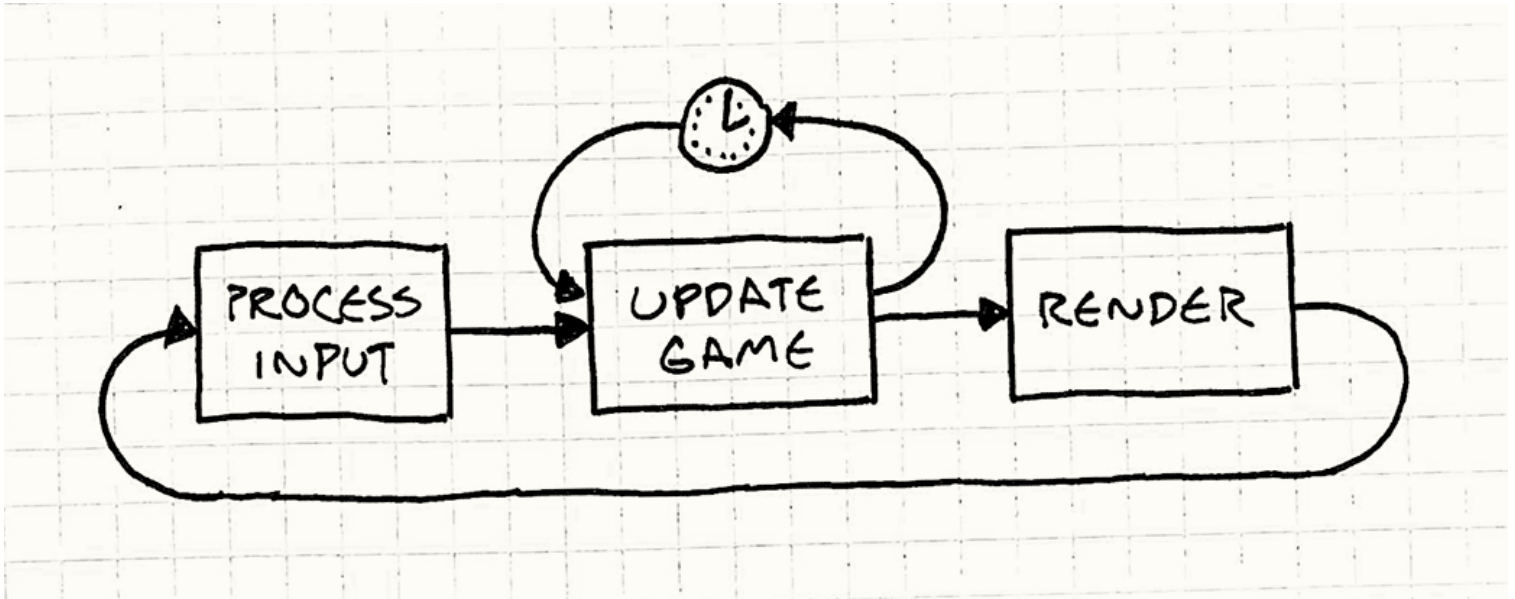
Programas interativos

Execução sem fim

Detecção de eventos

Controle de tempo

Game Loop



Fonte: http://gameprogrammingpatterns.com/game-loop.html

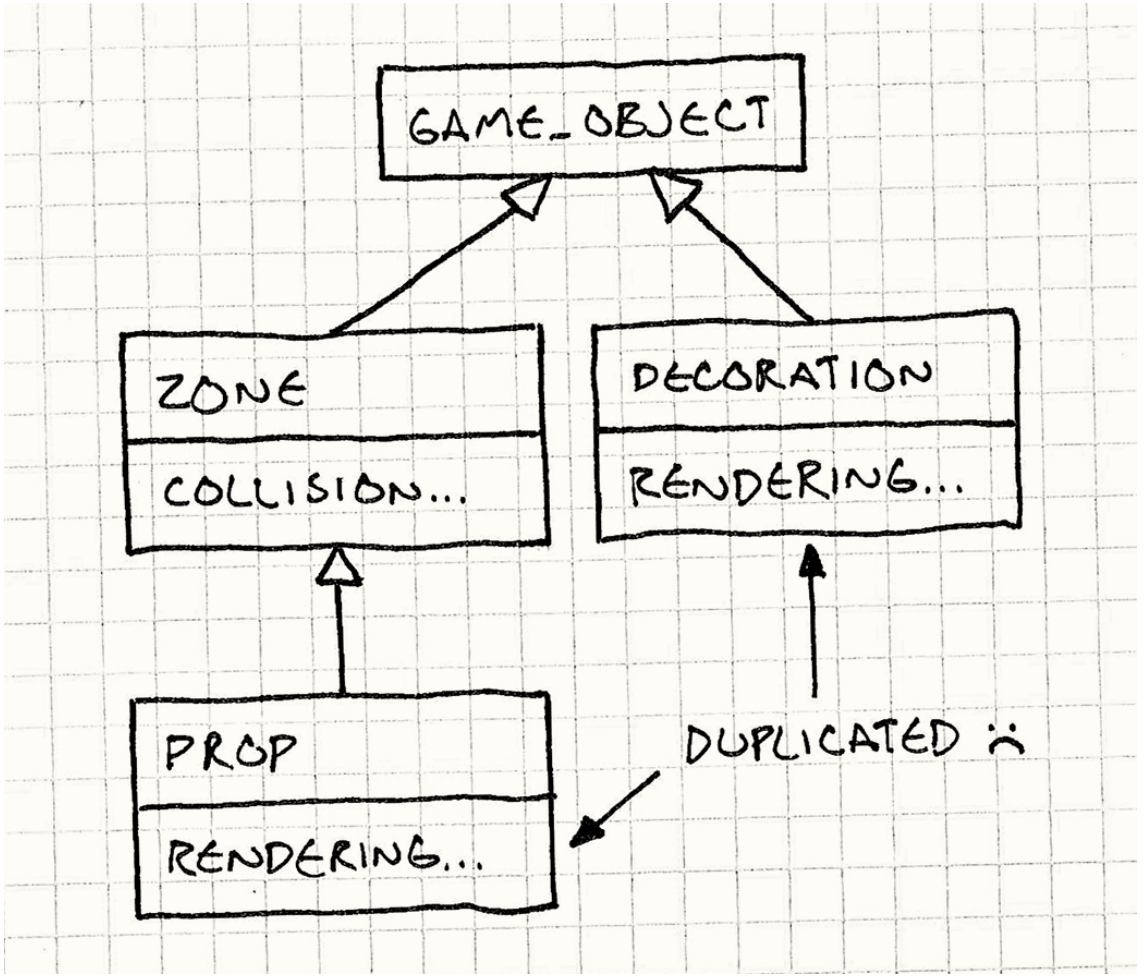
Jogos digitais

Padrões de projeto

Padrões de projeto

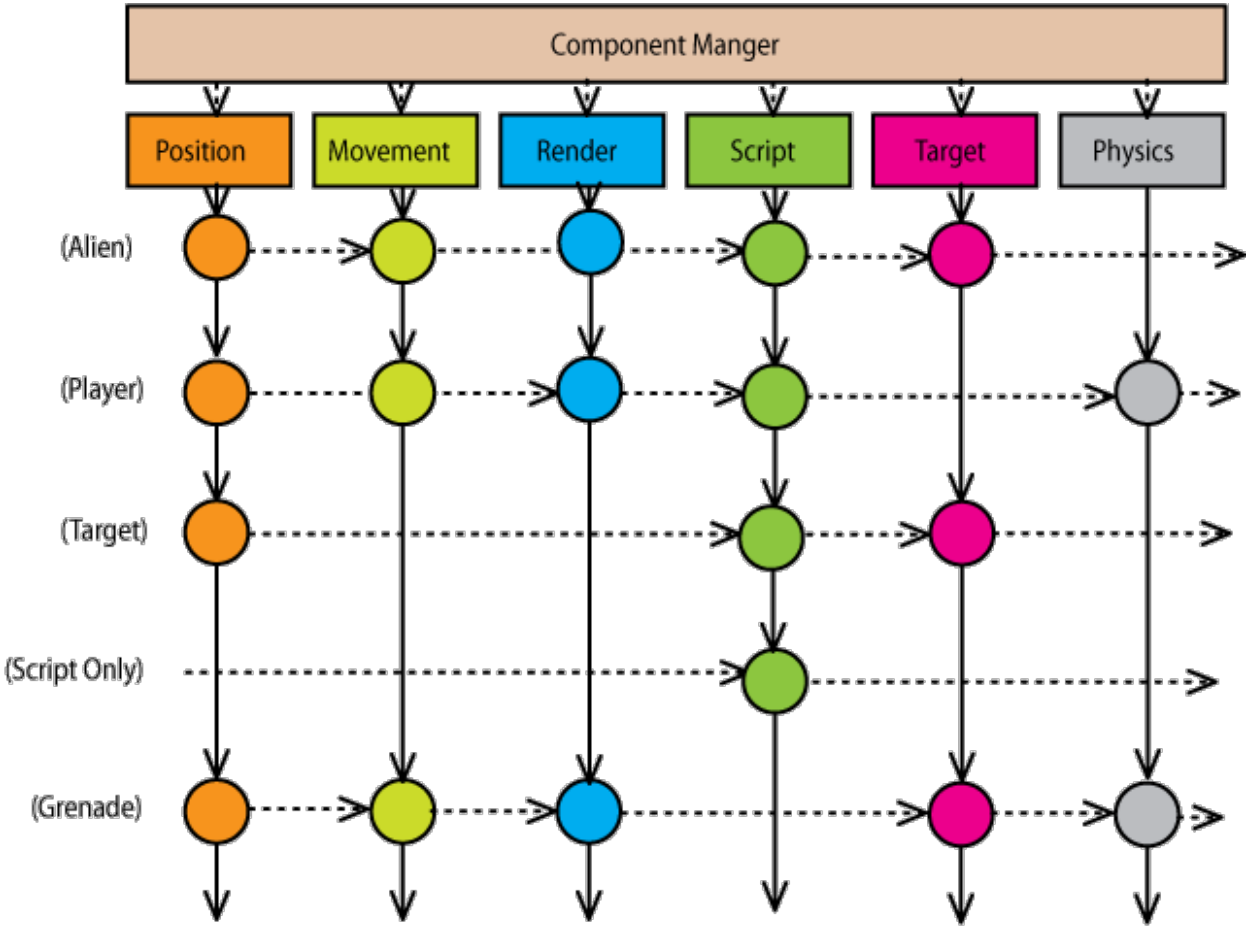
Como organizar os tipos de elementos do jogo?

Herança vs. Composição



Fonte: http://gameprogrammingpatterns.com/component.html

Herança vs. Composição



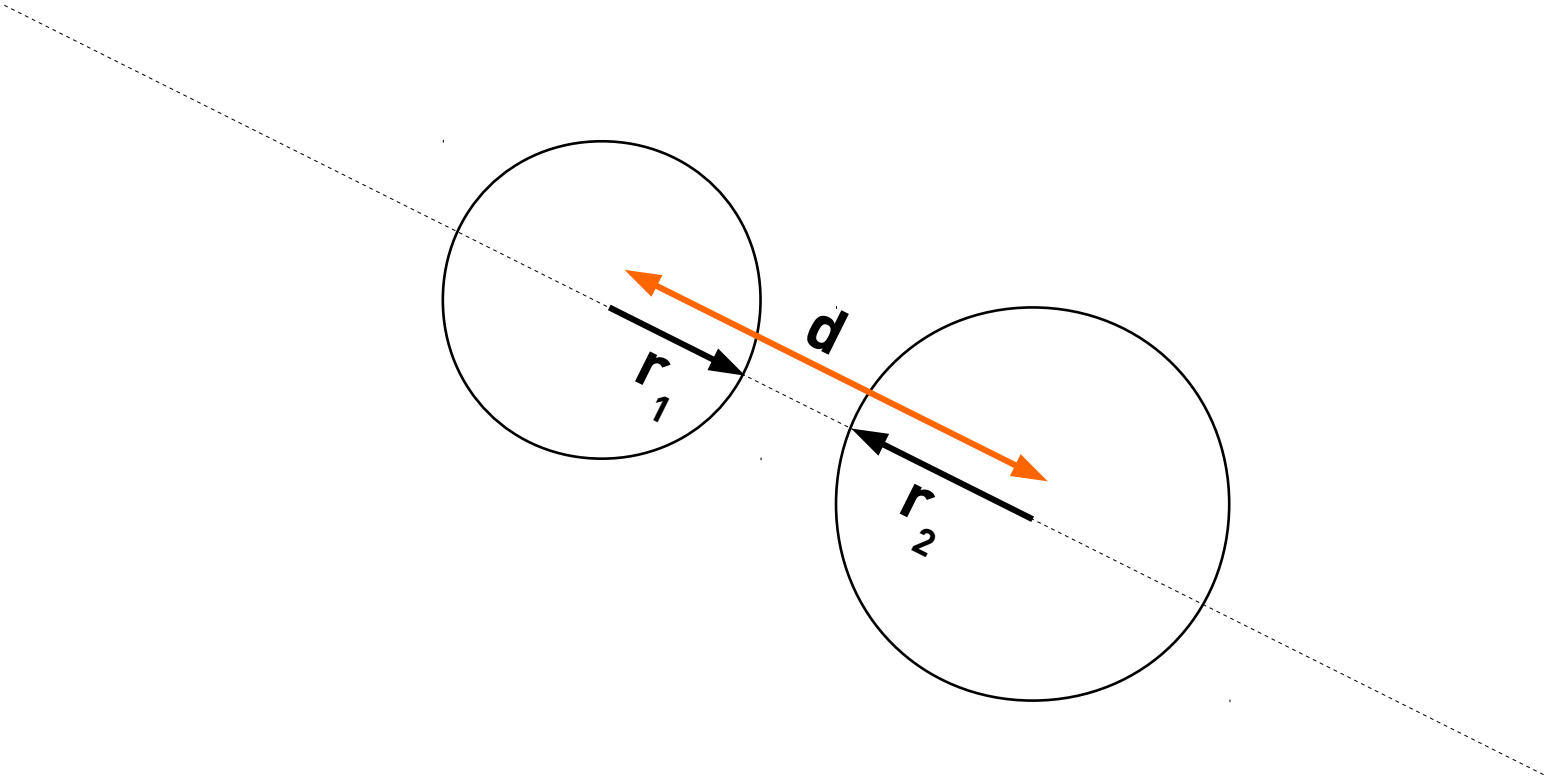
Fonte: http://cowboyprogramming.com/2007/01/05/evolve-your-heirachy/

Simulando Física

Colisões

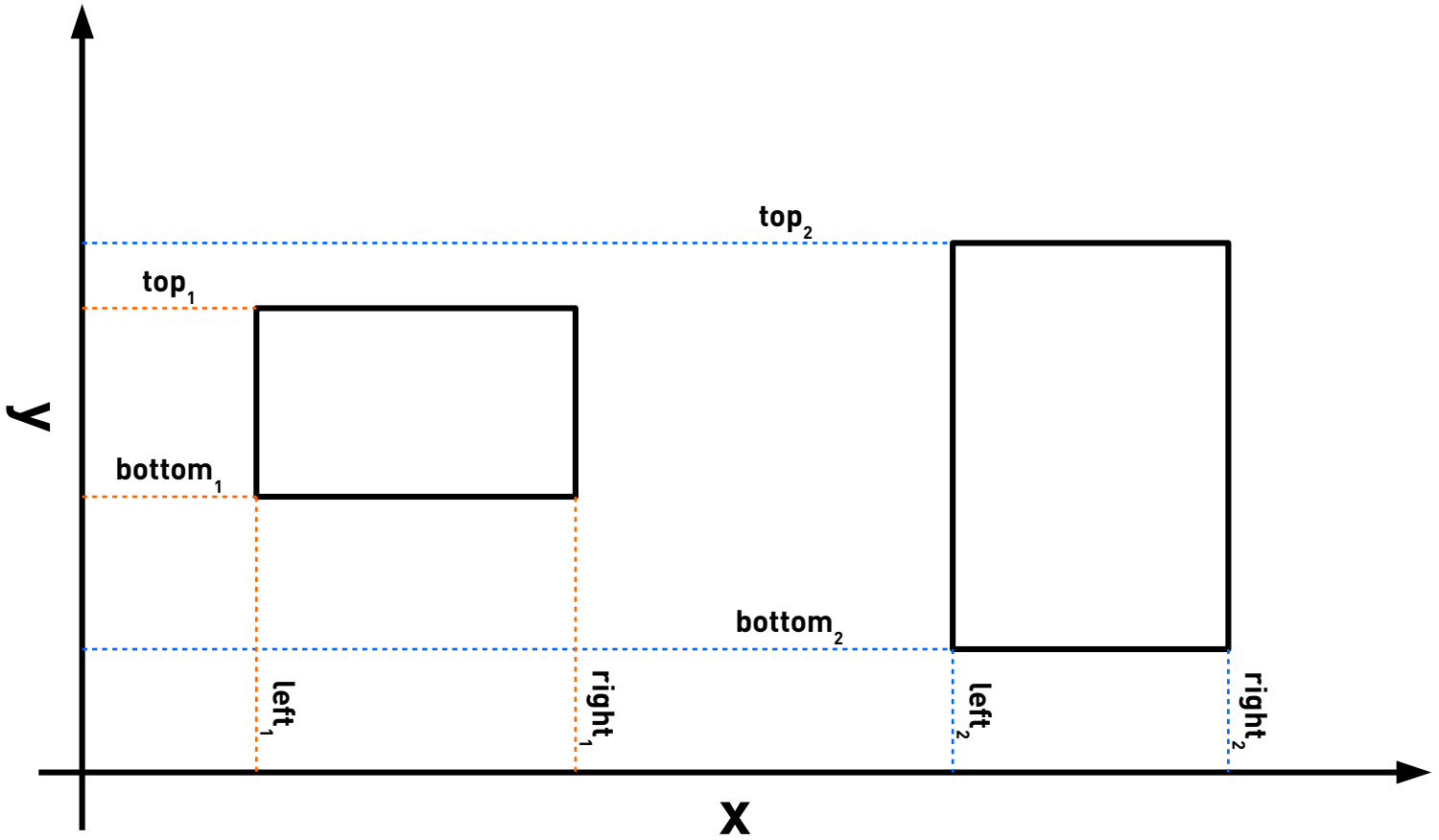
Casos fáceis

Círculos



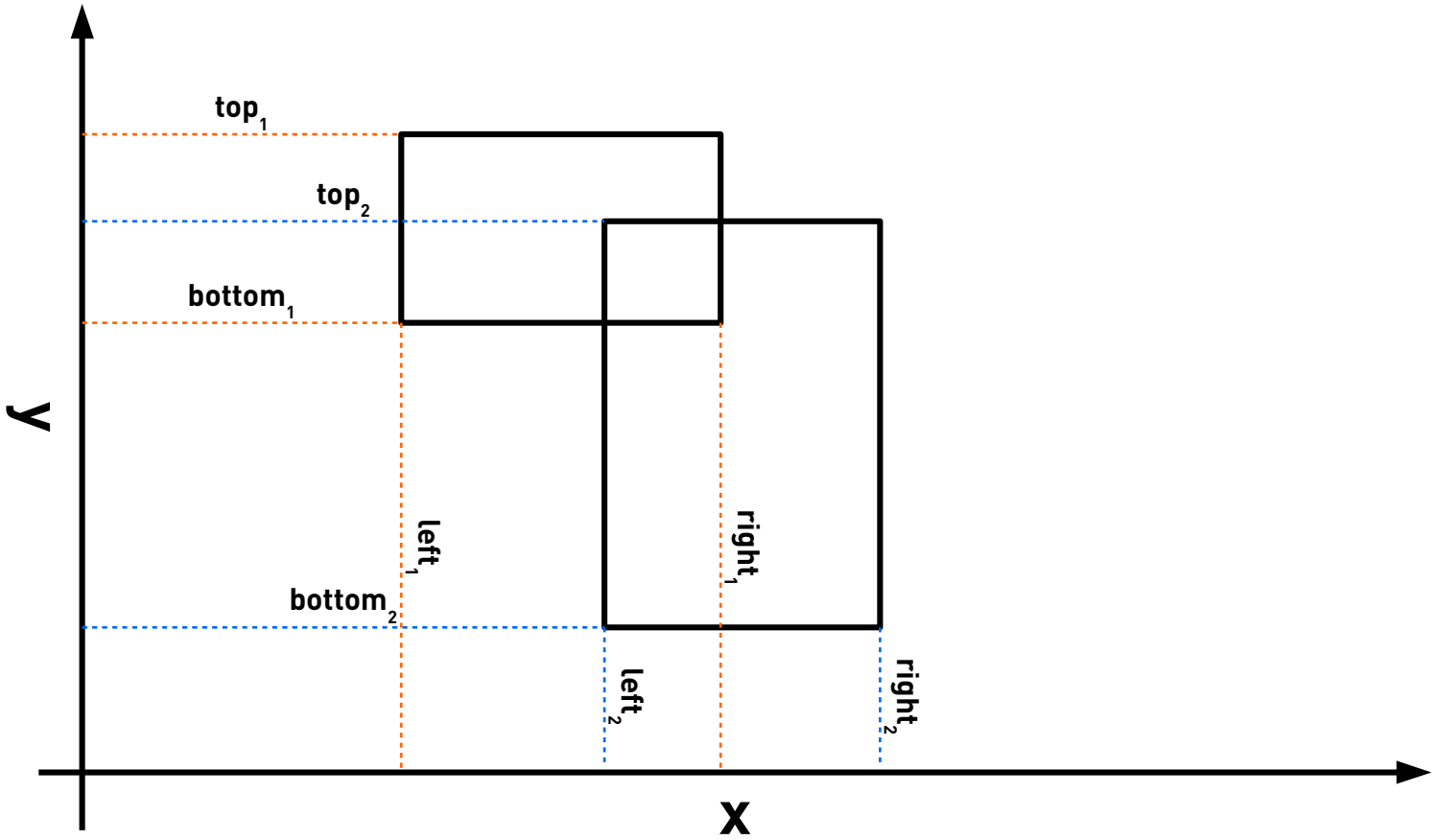
Casos fáceis

Axis-aligned bounding boxes



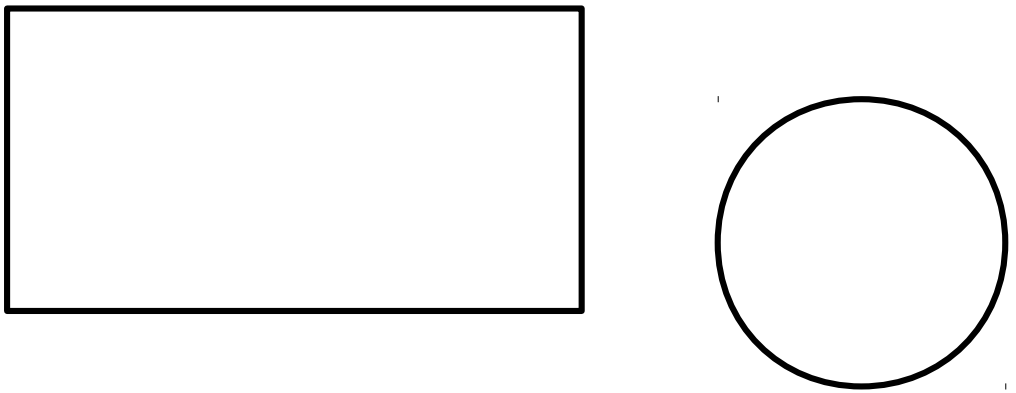
Casos fáceis

Axis-aligned bounding boxes



Círculo vs. retângulo

Vamos assumir uma AABB

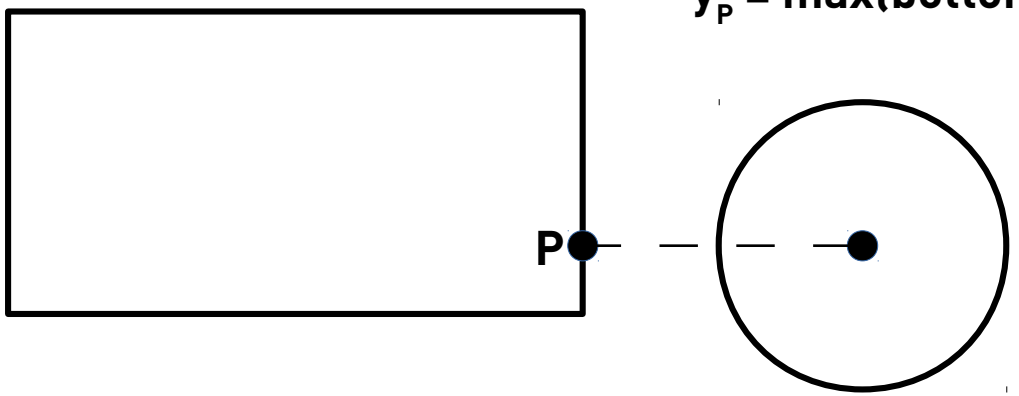


Círculo vs. retângulo

Ponto do retângulo mais próximo da circunferência

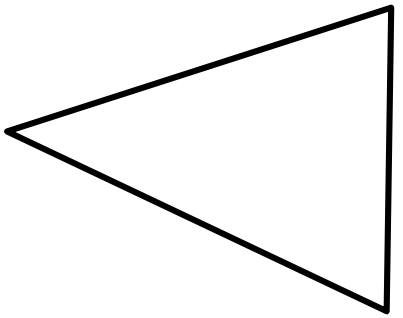
$x_P = \max(\text{left}, \min(\text{right}, x_C))$

$y_P = \max(\text{bottom}, \min(\text{top}, y_C))$



Ponto vs. Polígono convexo

Como saber se o ponto está dentro?



Ponto vs. Polígono convexo

Produto vetorial

$$\mathbf{v} \times \mathbf{w} = \begin{vmatrix} i & j & k \\ v_x & v_y & 0 \\ w_x & w_y & 0 \end{vmatrix} = \begin{vmatrix} v_x & v_y \\ w_x & w_y \end{vmatrix} k$$

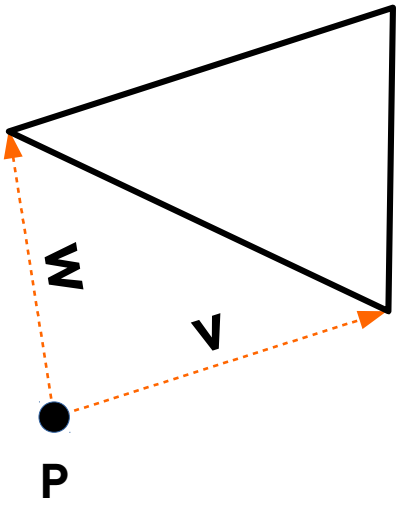
Ponto vs. Polígono convexo

Usamos o sinal do determinante!

$$\mathbf{v} \times \mathbf{w} = \begin{vmatrix} i & j & k \\ v_x & v_y & 0 \\ w_x & w_y & 0 \end{vmatrix} = \begin{vmatrix} v_x & v_y \\ w_x & w_y \end{vmatrix} k$$

Está fora se $\begin{vmatrix} v_x & v_y \\ w_x & w_y \end{vmatrix} > 0 \iff v_x w_y - v_y w_x > 0$

$\iff v_x w_y > v_y w_x$ (!!!)



Simulando Física

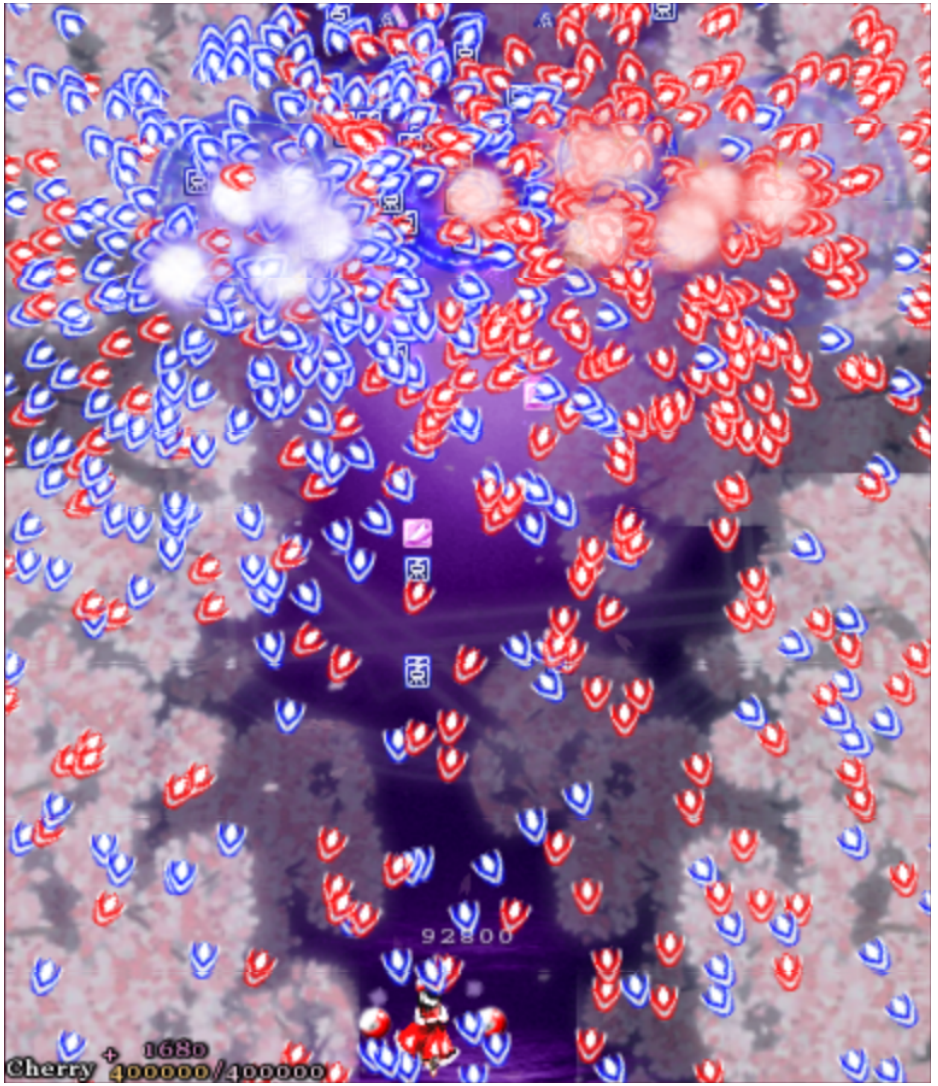
Detectar objetos próximos

Implementação ingênua

Todos contra todos

Colide?	obj2	obj3	obj4	...
obj1	?	?	?	?
obj2		?	?	?
obj3			?	?
...				?

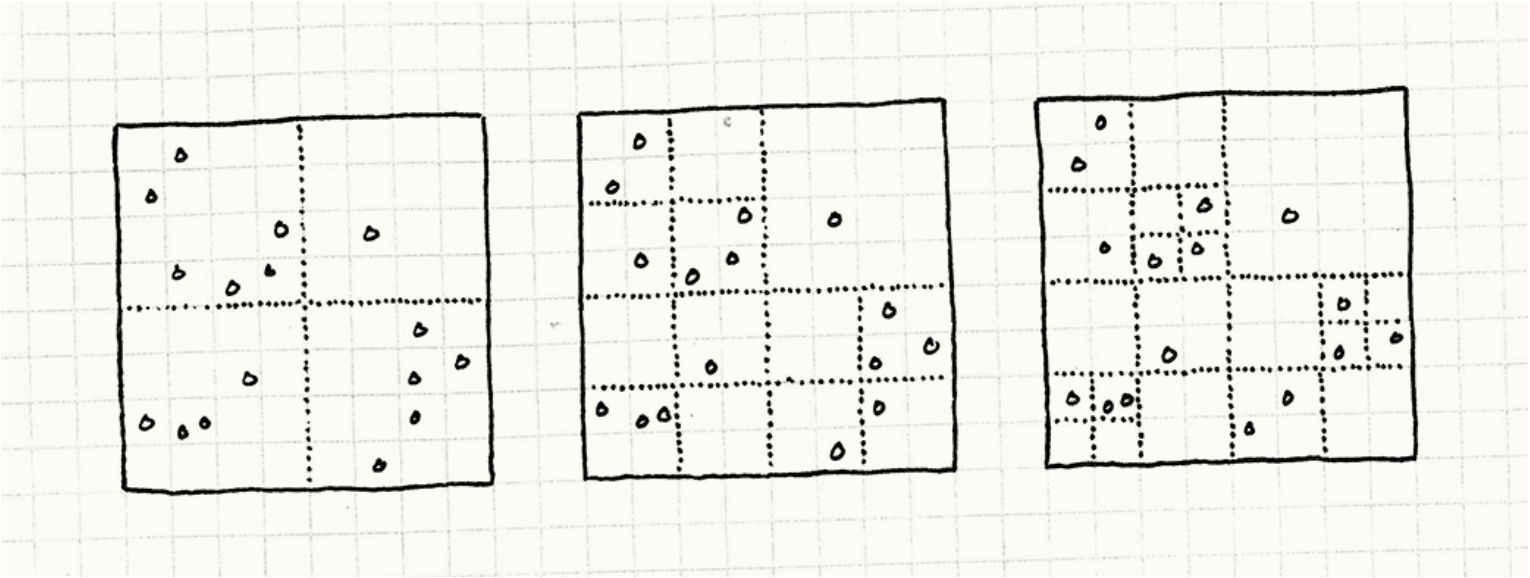
Implementação ingênua



Partição espacial

A ideia é agrupar objetos próximos

Exemplo: Quad-Tree



Fonte: http://gameprogrammingpatterns.com/spatial-partition.html

Partição espacial

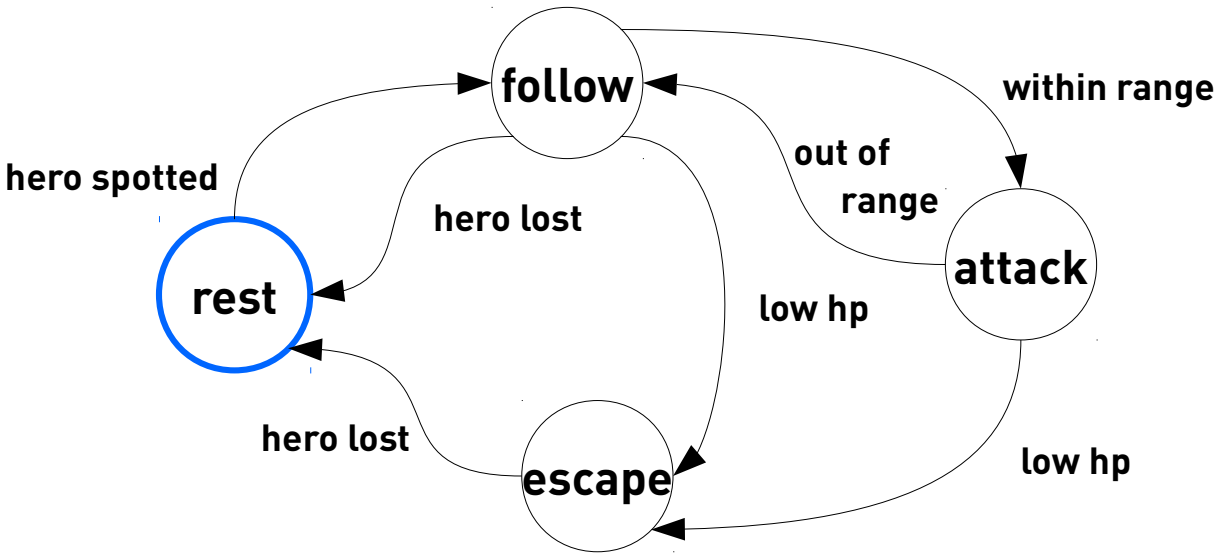
Outras estruturas conhecidas:

- **Grade fixa**
- **K-Dimensional Tree**
- **Interval K-Dimensional Tree**
- **Binary Space Partitioning Tree**
- **R-Tree**

Inteligência Artificial

Definindo comportamentos

Máquina de Estados Finita

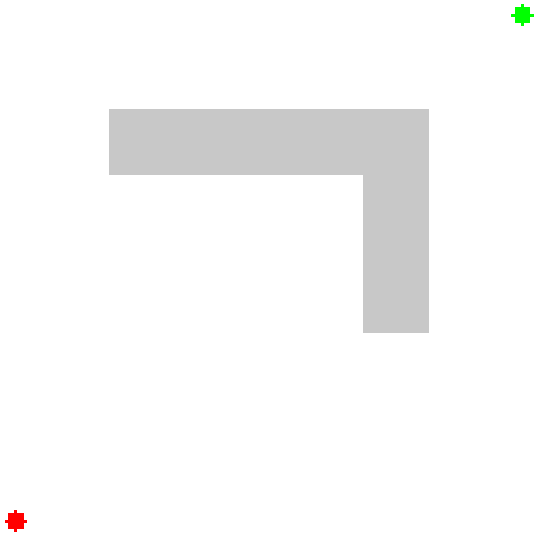


Inteligência Artificial

Path-finding

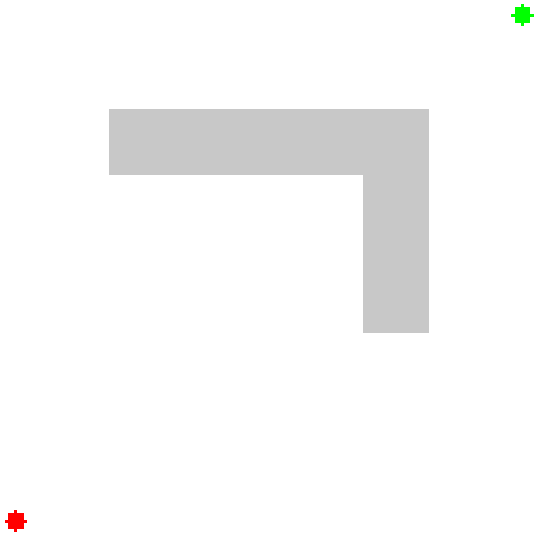
Algoritmo A*

Variação do algoritmo de Dijkstra



Algoritmo A* relaxado

Heurística com peso

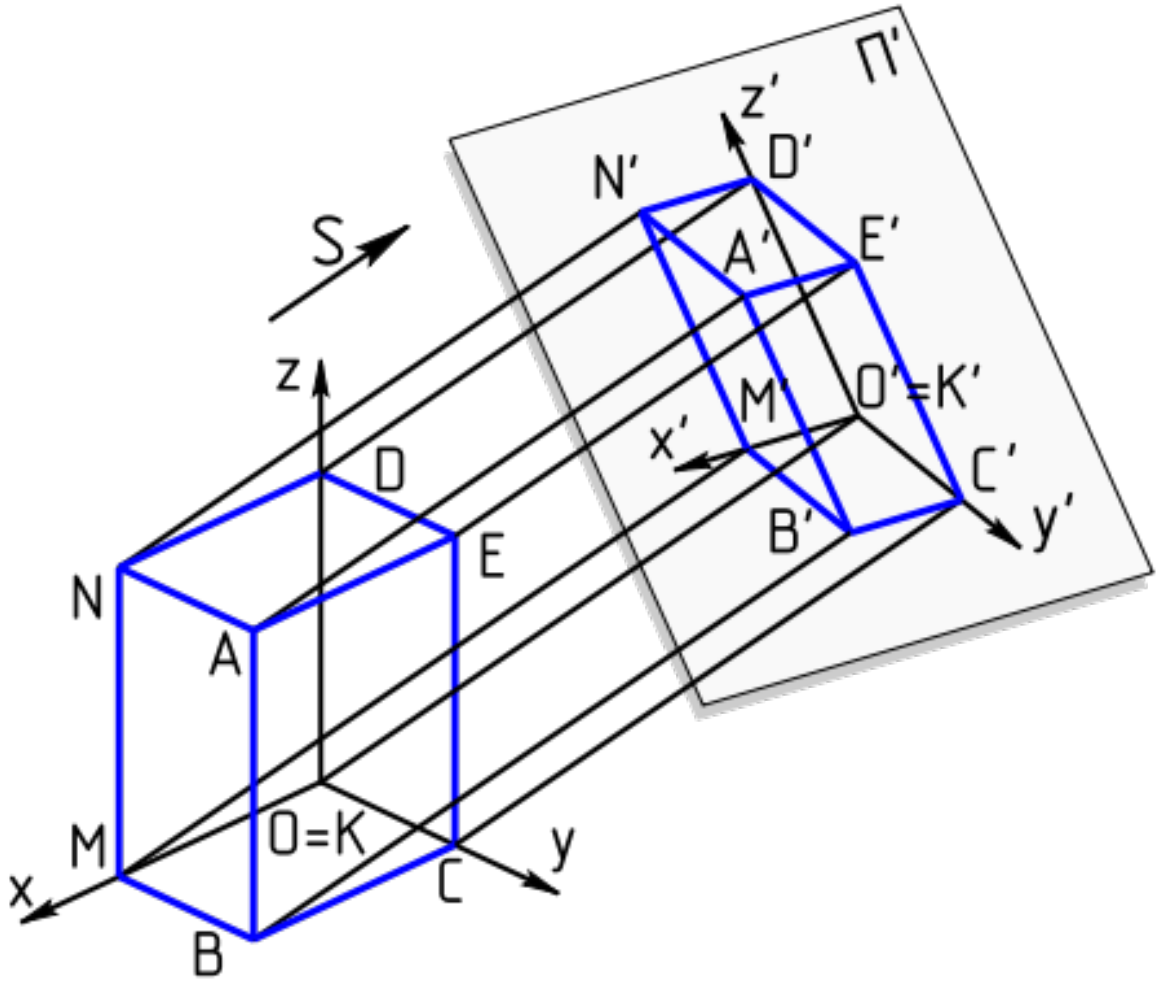


Renderização

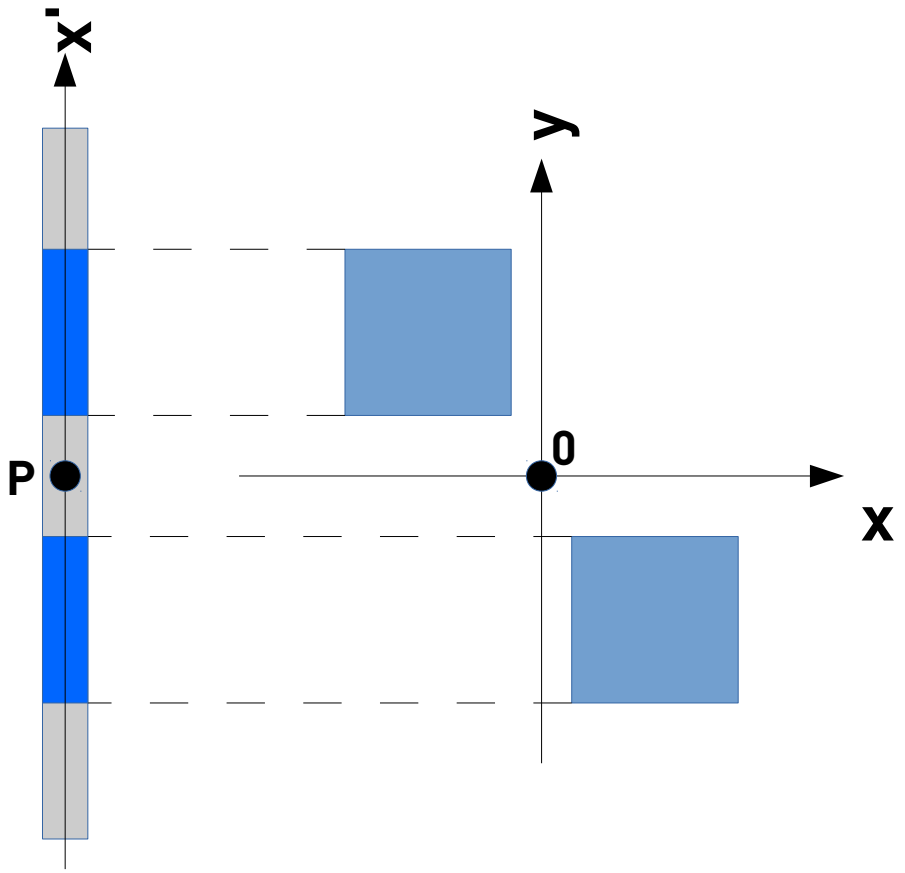
Apresentando o jogo

Spritesheets

Projeções tridimensionais



Projeções tridimensionais



Transformação Linear:

$T: R^2 \rightarrow R^2$

$T(v+u) = T(v) + T(u)$

$T(k.v) = k.T(v)$

$T(0) = 0$

Transformação Afim:

$P: R^2 \rightarrow R^2$

$P(v) = T(v) + p$

$P(0) = p$

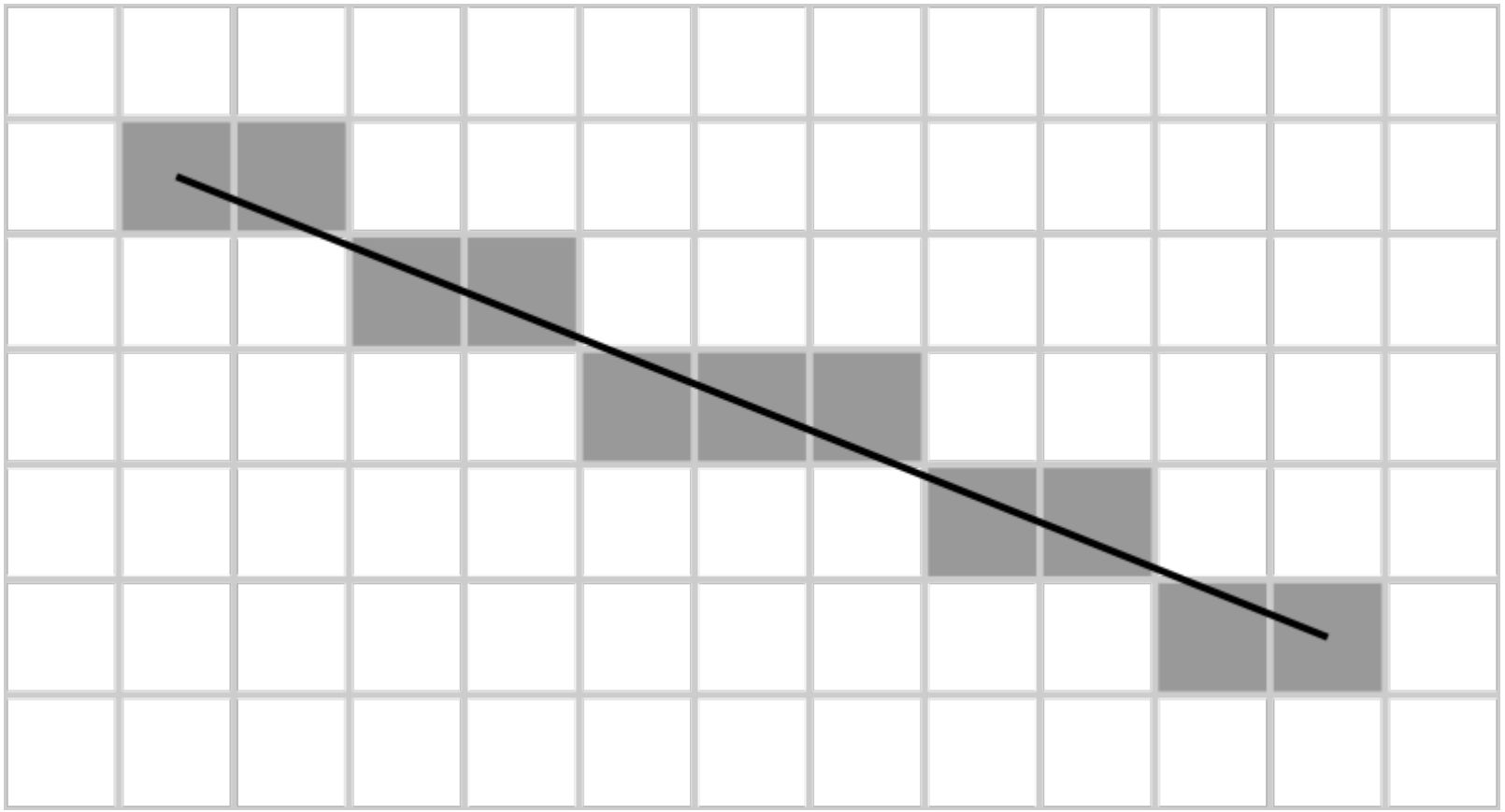
Projeções tridimensionais

Coordenadas homogêneas

$$\begin{bmatrix} a & b & u \\ c & d & v \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} ax + by + u \\ cx + dy + v \\ 1 \end{bmatrix} = \begin{bmatrix} a & b & 0 \\ c & d & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} + \begin{bmatrix} u \\ v \\ 0 \end{bmatrix}$$

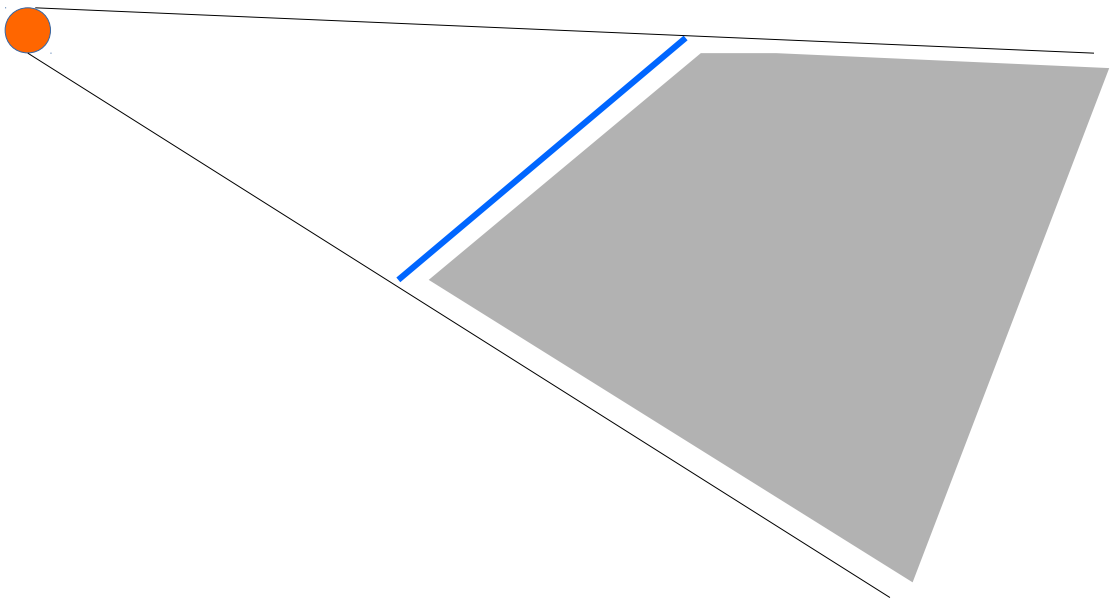
$$\begin{bmatrix} a & b & u \\ c & d & v \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 0 \end{bmatrix} = \begin{bmatrix} ax + by \\ cx + dy \\ 0 \end{bmatrix}$$

Rasterização



Sombras

Shadow-casting simples



Conteúdo procedimental

Autômatos celulares

Gramáticas

Regras

A → ABA

B → CC

Expansão

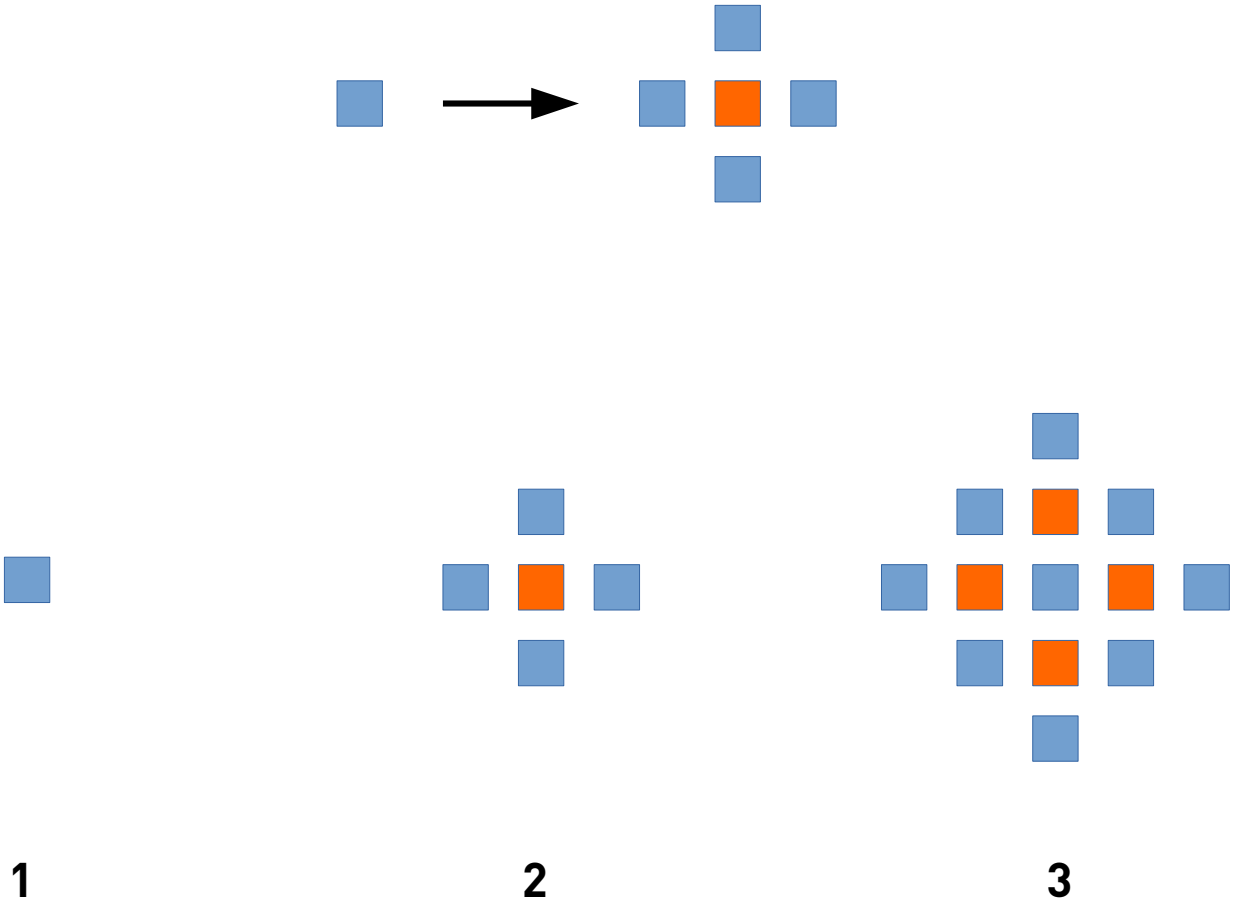
1) A

2) ABA

3) ABACCABA

4) ABACCABACCABACCABA

Autômato celular



Conteúdo procedimental

Outros algoritmos e técnicas

- **Noises**
- **SATs**

Unlimited Slide Works

Obrigado!

www.uspgamedev.org

www.fb.com/uspgamedev

contato@uspgamedev.org

Unlimited Slide Works

- Networking
  - Peer-2-peer (Age of Empires)
  - Client/Server (QuakeWorld)
- Shaders
- Multithreading