

# **Motors basats en tiles**

Professors de VJ

# Motors basats en tiles

- TBE (Tile Based Engines)
- Tile
  - Bloc fonamental d'un tauler.
  - Habitualment quadrat.
- Objectius
  - Estalviar memòria.
  - Reutilitzar gràfics.
  - Generació de contingut dinàmic.

# Objectius

- Reutilitzar gràfics
  - Temps dels artistes és important (i car).
    - Artistes ↔ Conjunt de tiles
    - Dissenyador de nivells ↔ Mapes
- Contingut dinàmic
  - Mapes generats “aleatoriament”.
    - Millor proceduralment.
  - Millora la rejugabilitat.

# Tiles

A nivell formal:

- Imatge
- Amplada i alçada
  - Cal decidir a partir de:
    - Tamany de mapa
    - Restriccions de memòria
  - Habitualment potència de 2: 16, 32, 64, ...
    - Hardware antic: Operacions més simples.
    - Hardware nou: Textures potència de 2.

# Motors basats en tiles

**Conjunt de tiles**



**Mapa**



# Motors basats en tiles

**Conjunt de tiles**



**Mapa**



|   |   |   |
|---|---|---|
| 0 | 2 | 0 |
| 1 | 3 | 4 |
| 1 | 6 | 5 |

# Motors basats en tiles

Els tiles es poden agrupar en una sola textura.

**Textura**



**Conjunt de tiles**

**Mapa**

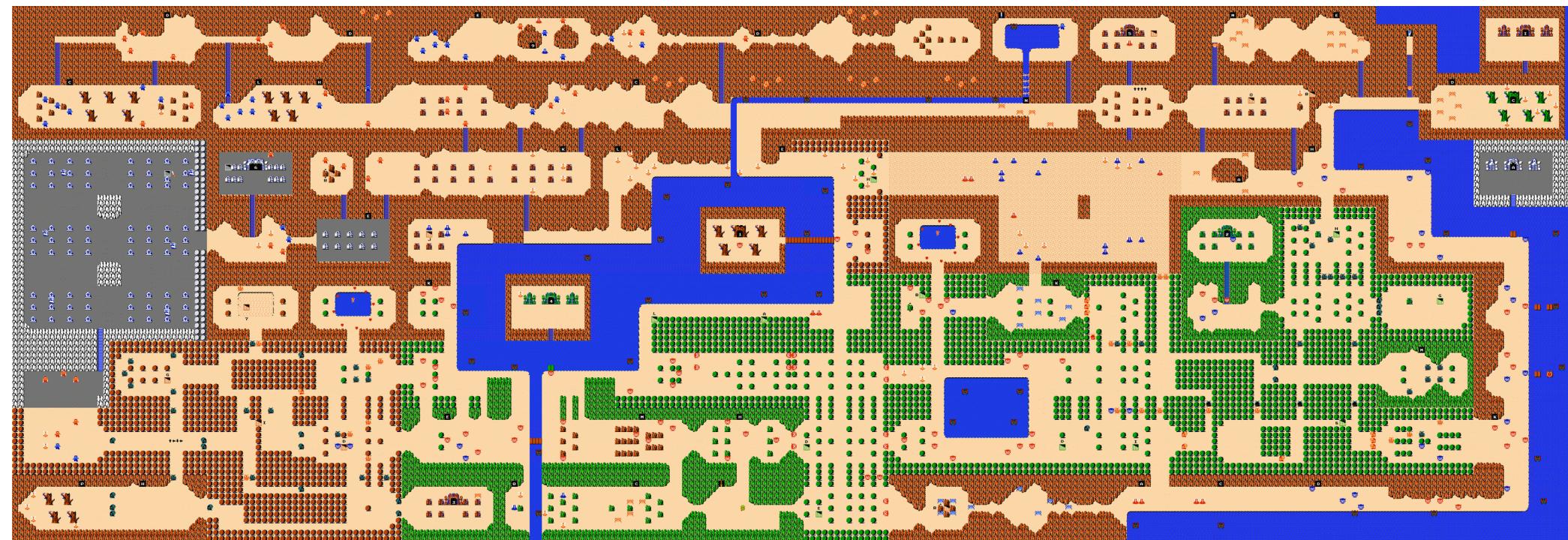


|   |   |   |
|---|---|---|
| 0 | 2 | 0 |
| 1 | 3 | 4 |
| 1 | 6 | 5 |

0      1      2      3      4      5      6

| x | y | x | y | x | y | x | y | x | y | x | y |   |   |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 0 | 2 | 0 | 0 | 1 | 1 | 1 | 2 | 1 | 0 | 2 |

# The Legend of Zelda (NES, 1986)

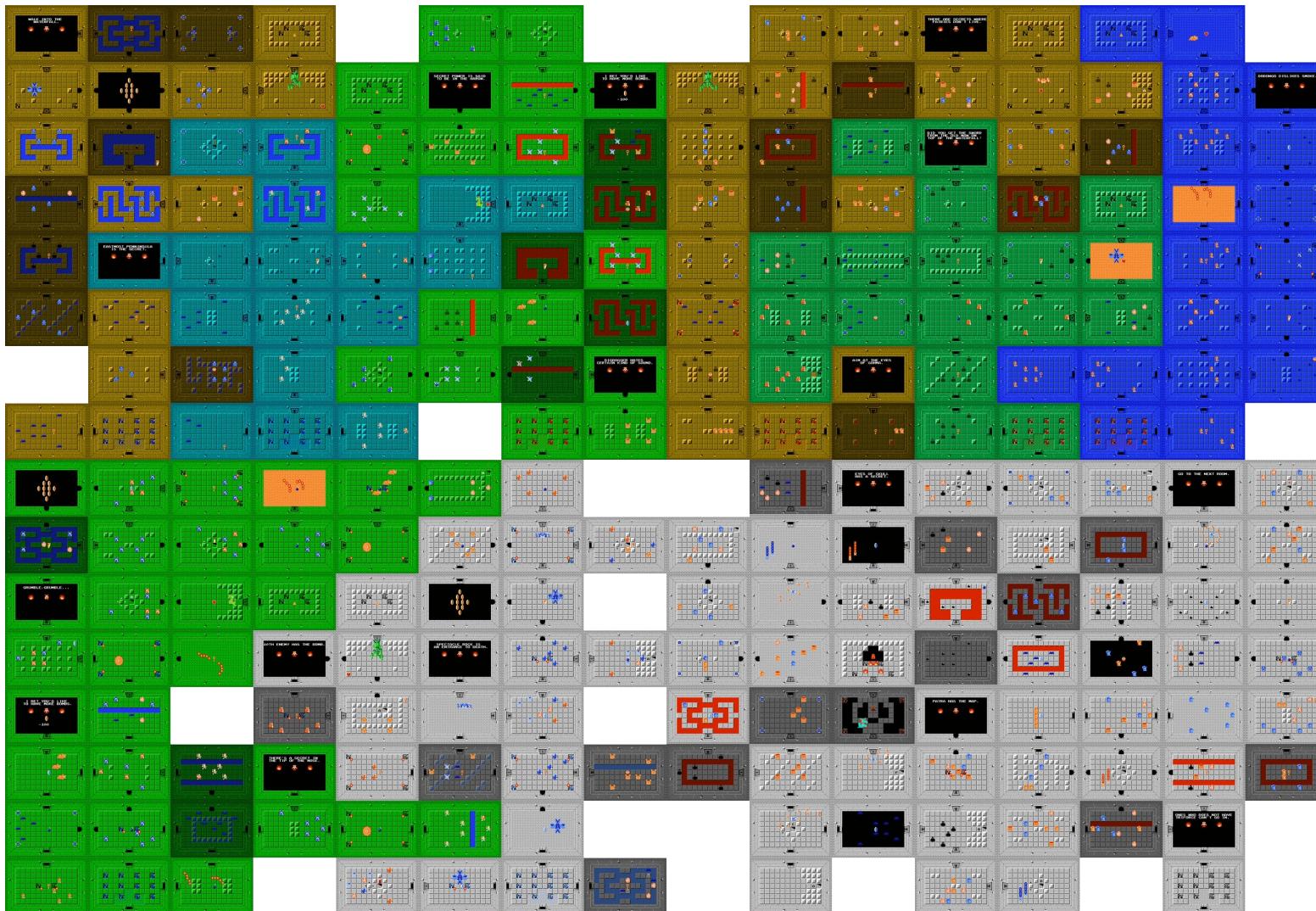


4096 x 1408 pixels

NES → Joc en ROM. Màxim 1 MB.

# The Legend of Zelda (NES, 1986)

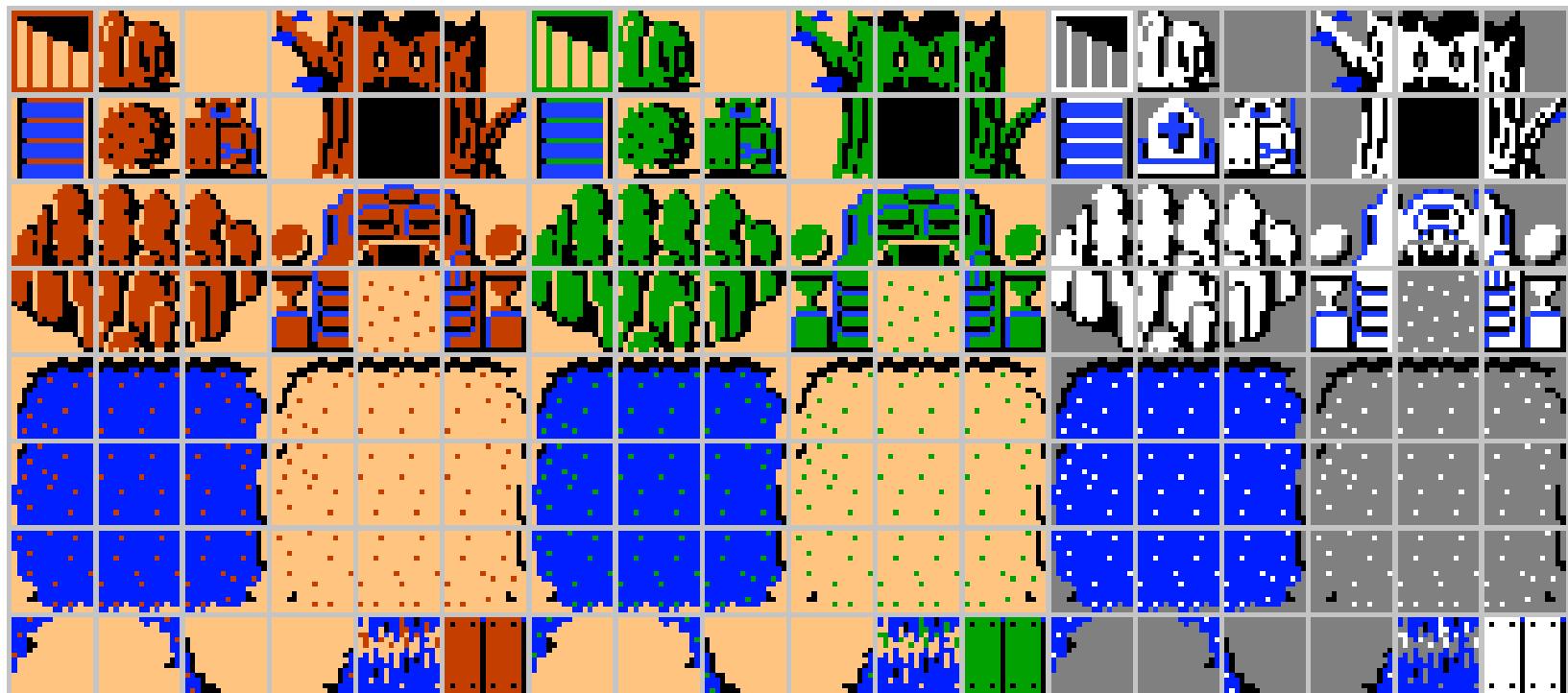
Encara hi havia més:



4096 x 2816

# The Legend of Zelda (NES, 1986)

## Tiles (Zelda Overworld)



Complete overworld tileset from the original Legend of Zelda.  
Ripped by Zephie187, please give credit!

$$18 \times 8 = 144 \text{ Tiles de } 16 \times 16$$

# The Legend of Zelda (NES, 1986)

Consum de memòria (suposant 4 bits/pixel)

- Com a imatge:
  - $4096 \times 1408 = 2.75$  MB
- Com a mapa de tiles:
  - Joc de tiles:  $18 \times 8$  tiles de  $16 \times 16 = 18$  KB
  - Mapa:  $256 \times 88 = 22$  KB
  - Total: 40 KB
- 100 vegades menys.

# Layers

La majoria de jocs necessiten objectes estàtics al mapa:

- Vegetació, Objectes, ...

No volem tenir un tile per a cada combinació de terra i objecte.

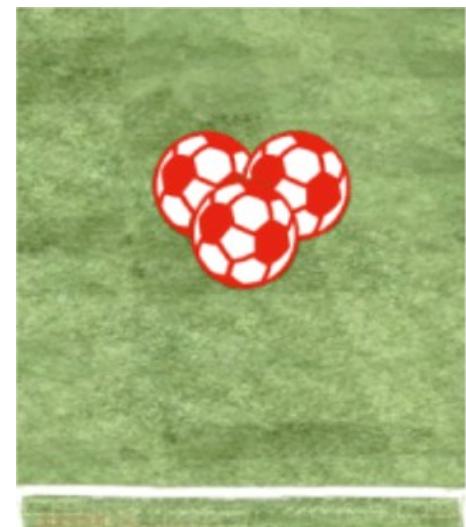
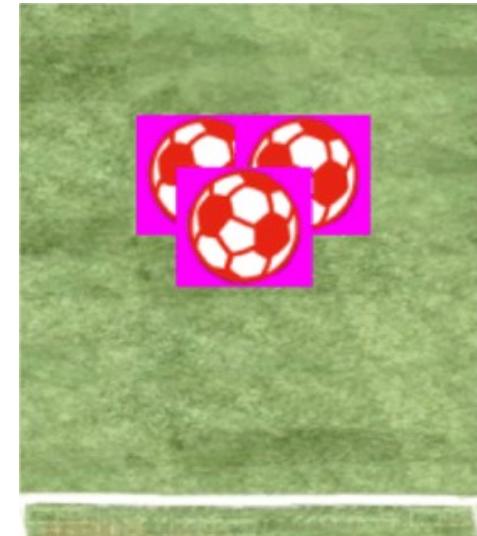
Solució:

- Múltiples capes (layers).

# Transparència

Múltiples capes → Transparència

- Via color clau (color key)
  - Mètode antic.
  - Un color s'ignora quan es pinta.
- Via canal alpha (RGBA)
  - OpenGL i DirectX ho suporten directament.
  - Permet transparència variable ( $A \neq 0$  i  $A \neq 1$ ).
    - Elimina el aliasing de la transparència.



# Layers

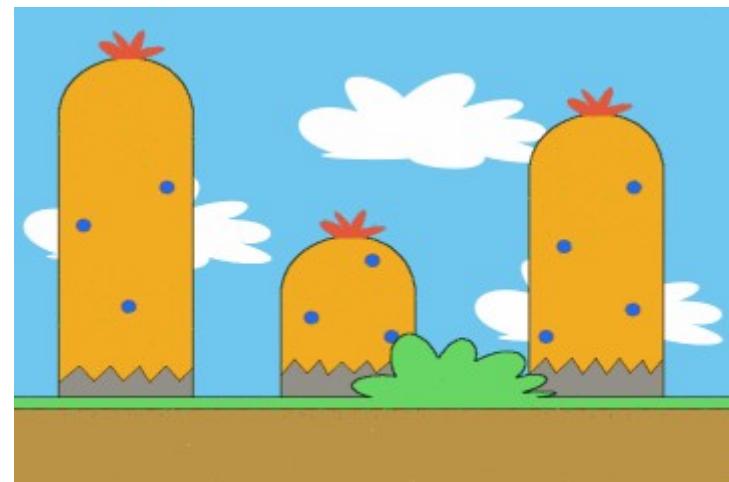
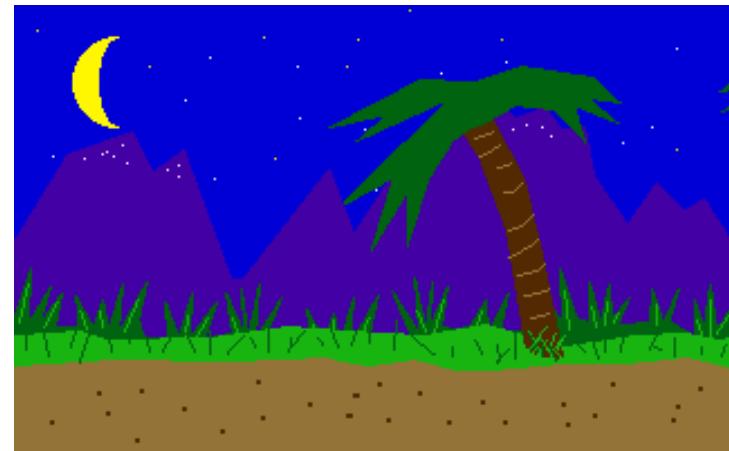
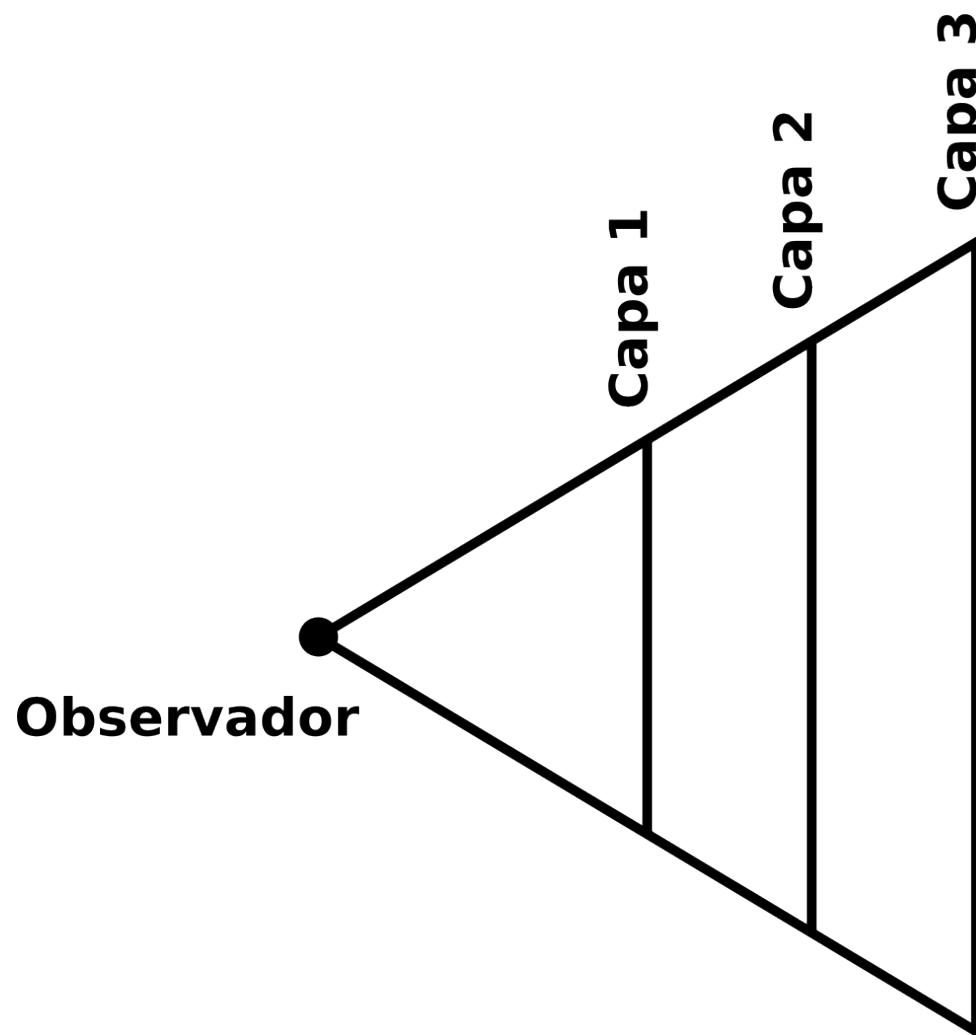
Hi ha dos tipus de capes:

- Capes denses
  - Es fan servir com a fons.
  - S'implementen com a mapes de tiles.
- Capes disperses
  - Es fan servir per mostrar objectes.
    - Normalment estàtics.
  - S'implementen com un vector.

# Parallax

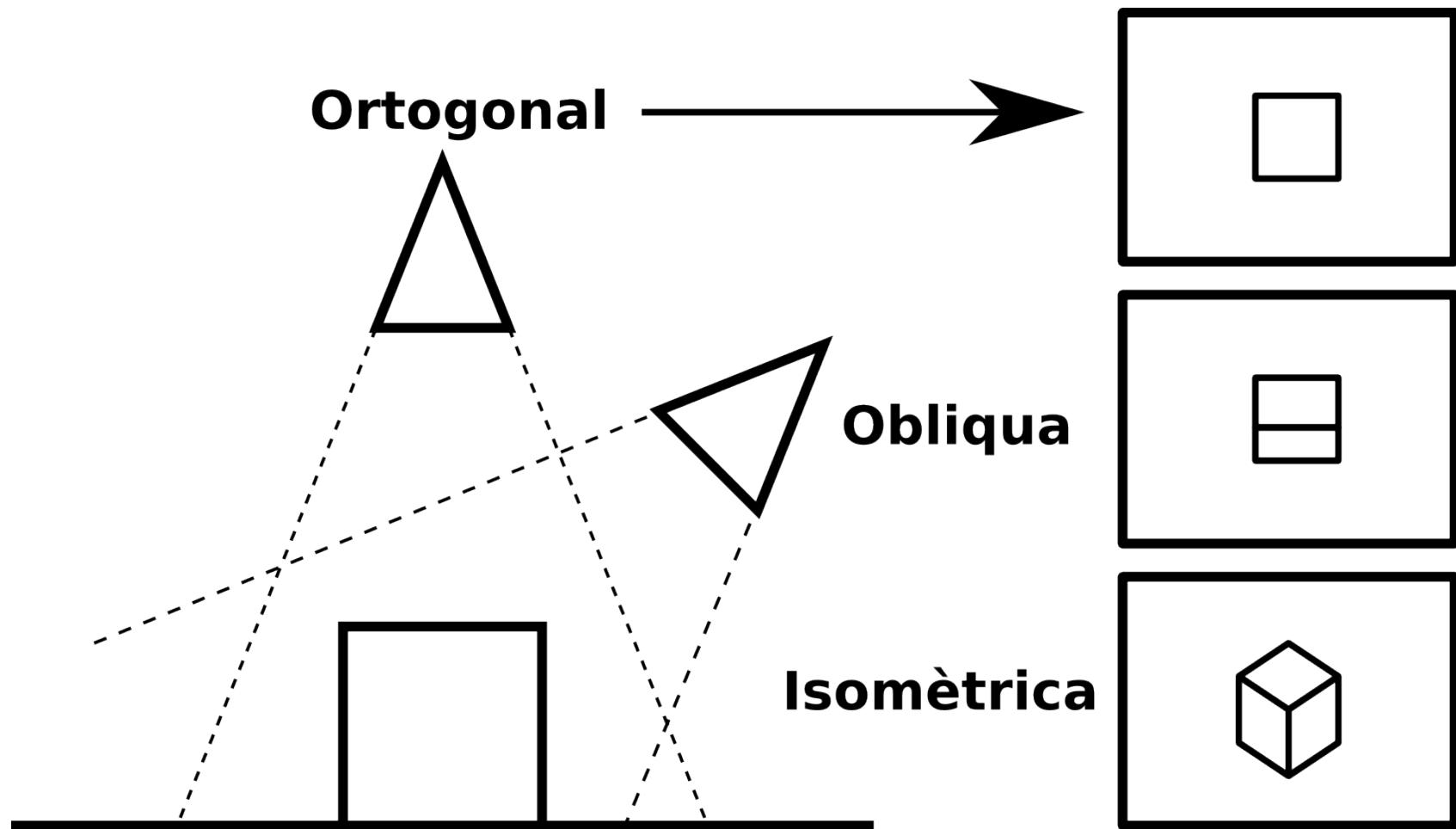
- Parallax scrolling
  - Tenir diverses capes.
  - Es mouen a diferents velocitats.
- Resultat
  - Efecte de profunditat.
- Implementació
  - Adaptar tamany i velocitat a la distància.

# Parallax

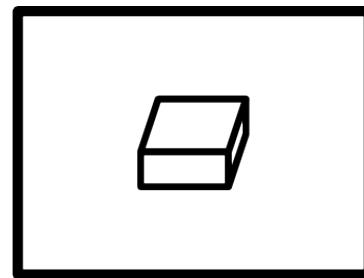
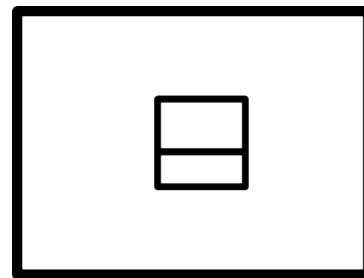


# Projeccions

Es poden fer servir diferents projeccions:



# Obliqua



# Isomètrica

Elevació de 30° i rotació de 45°.



Q\*Bert



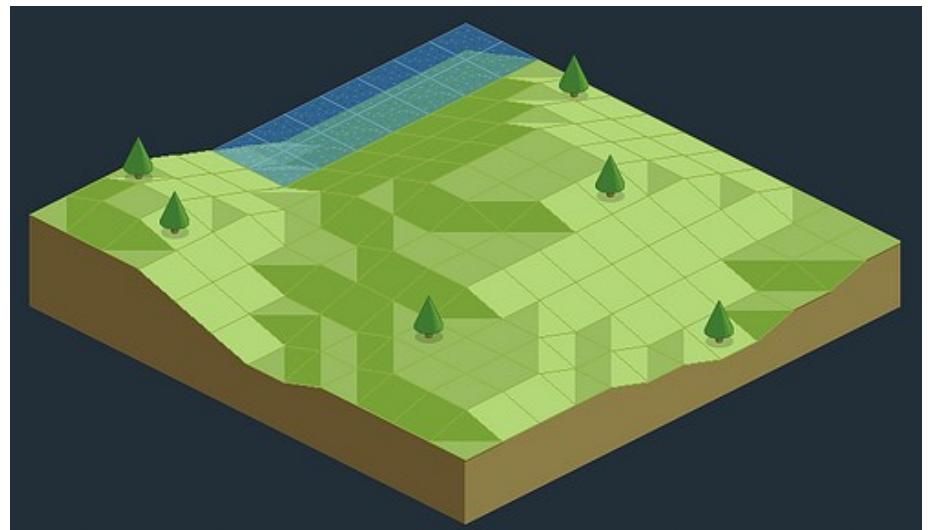
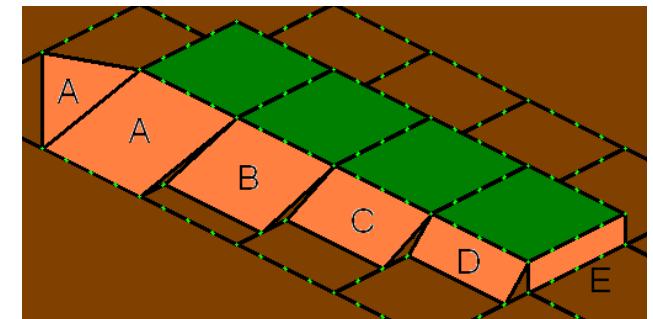
Diablo

# Isomètrica

## Tècniques avançades

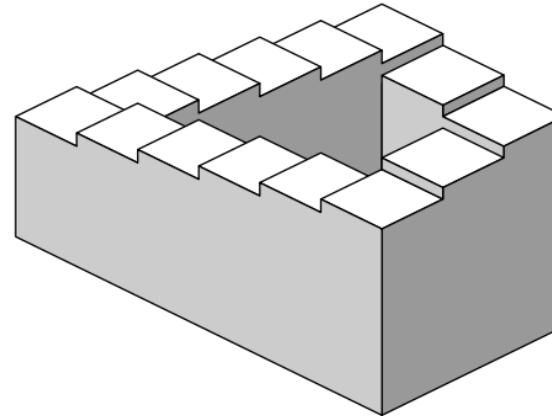
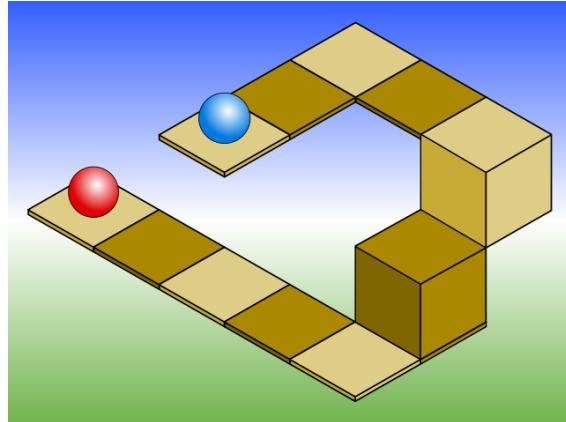
- Elevation layers

- Les capes addicionals es poden posar de forma que encaixin en altura.
- Cal desplaçar el mapejat dels tiles a pantalla.
- És interessant tenir tiles de connexió.



# Inconvenients

- Problemes en la percepció de l'altura.
- Possibilitat d'escenes impossibles.



## Solució

- Fer servir projecció perspectiva.
  - Gràfics 3D.
  - Encara que lògica 2D.