

Trabajo de fin de grado

Videojuego basado en la generación procedimental de mundos o niveles



Grado en Ingeniería Multimedia

Rubén Martínez Vilar

Tutores:

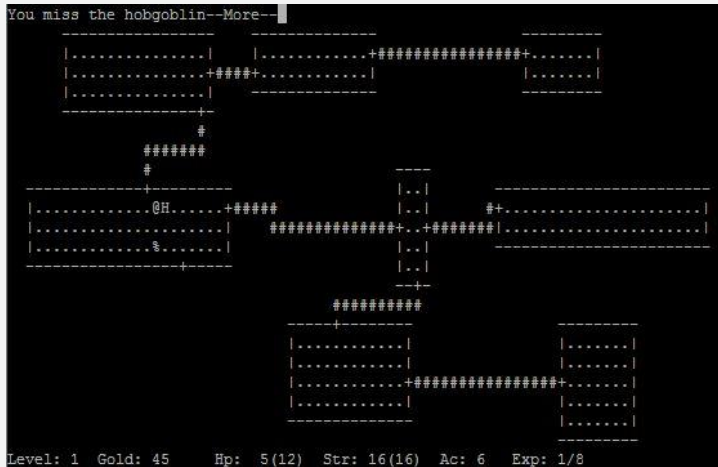
Faraón Llorens Largo

Francisco José Gallego Durán



Universitat d'Alacant
Universidad de Alicante

Introducción



Rogue (1980)



Spelunky (2008)



Minecraft (2008)



No Man's Sky (2015?)

Objetivos

- **Estudio de la definición de generación procedimental de contenido** dentro del sector de los videojuegos.
- **Estudio de distintos videojuegos** que utilizan técnicas de generación procedimental de contenido.
- **Estudio de distintos algoritmos de generación de mazmorras.** Implementación visual.
- **Diseño y desarrollo de un videojuego** sencillo de estilo rogue-like, haciendo uso justificado de alguno de los algoritmos estudiados para la generación del contenido.

El videojuego

Características de un rogue-like:

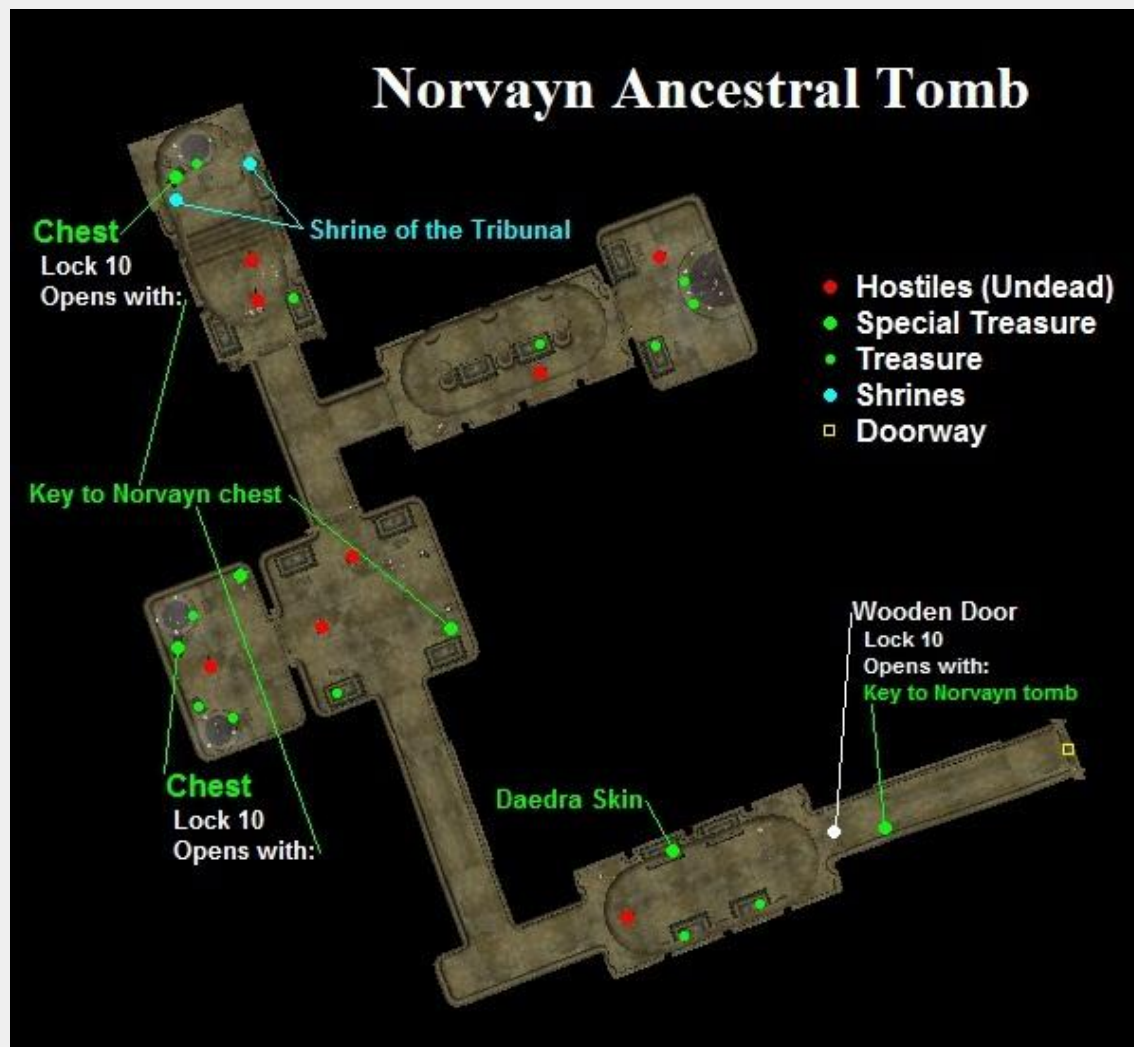
- Contenido generado mediante algoritmos.
- Permadeath.

Características del videojuego:

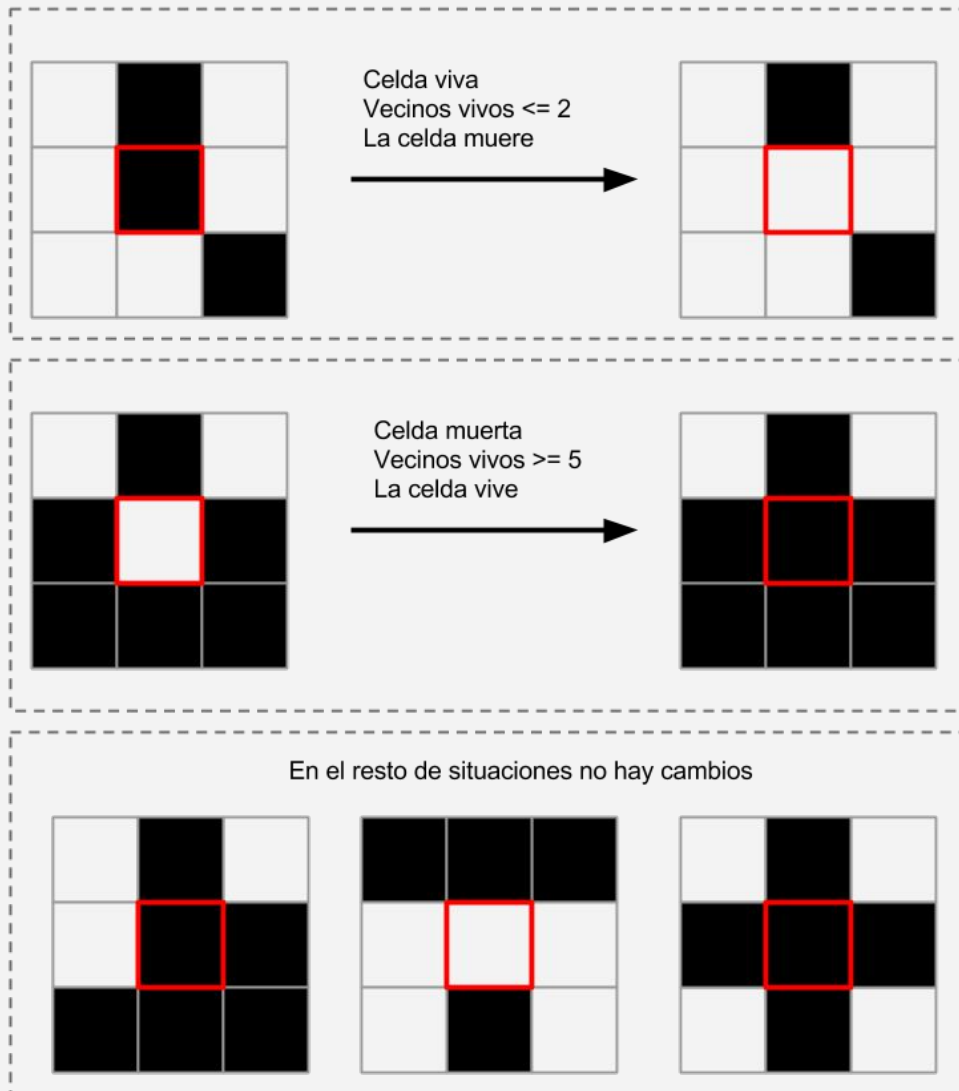
- Rogue + Spelunky en un entorno 3D.
- Objetivo: conseguir tesoros.
- Trampas.
- Enemigos.
- Pociones.

Unity

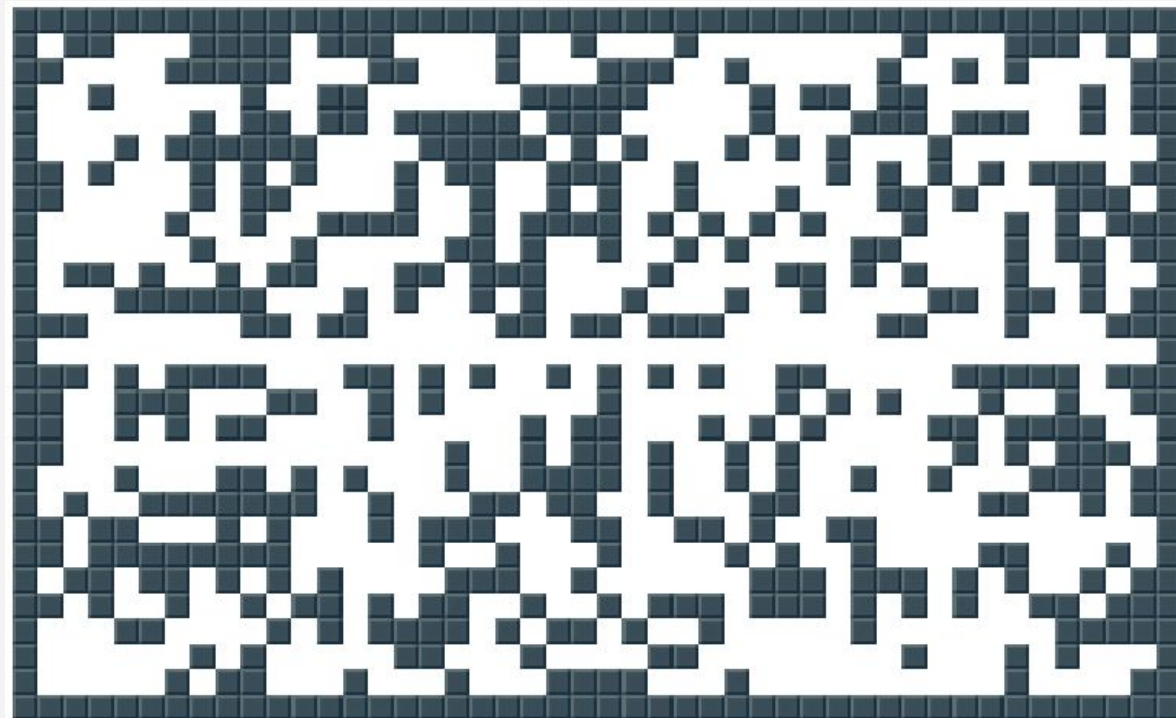
El videojuego



Autómata Celular



Autómata Celular

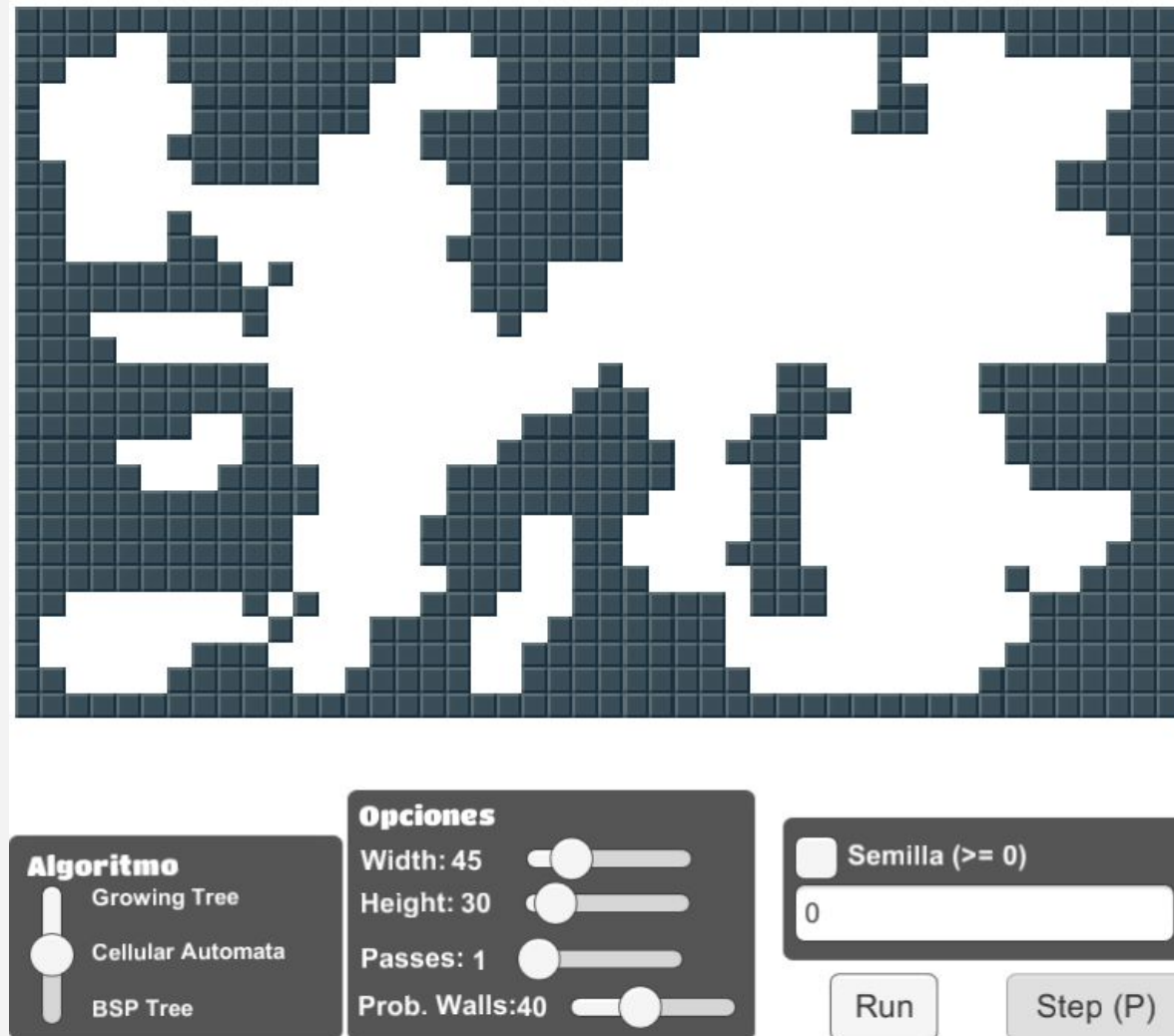


Algoritmo
☐ Growing Tree
☒ Cellular Automata
☐ BSP Tree

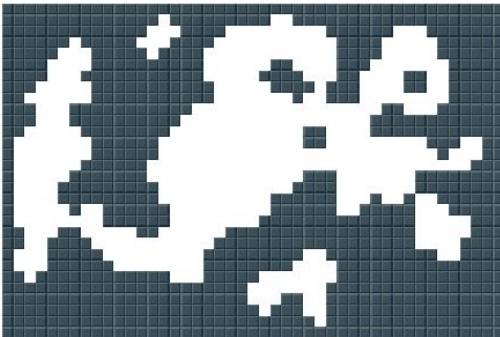
Opciones
Width: 45
Height: 30
Passes: 1
Prob. Walls: 40

☐ Semilla (≥ 0)

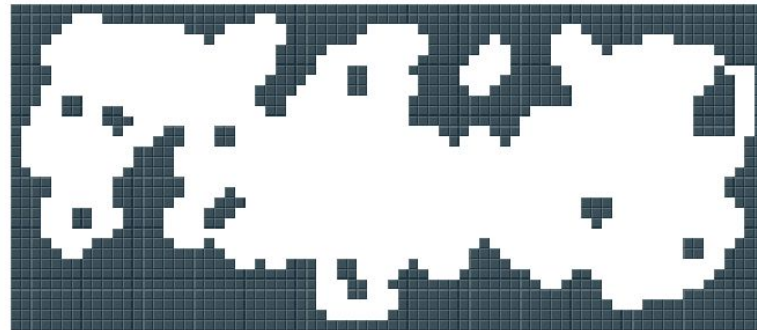
Autómata Celular



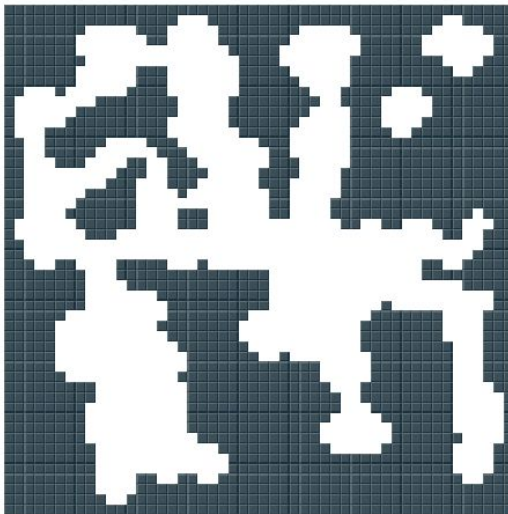
Autómata Celular



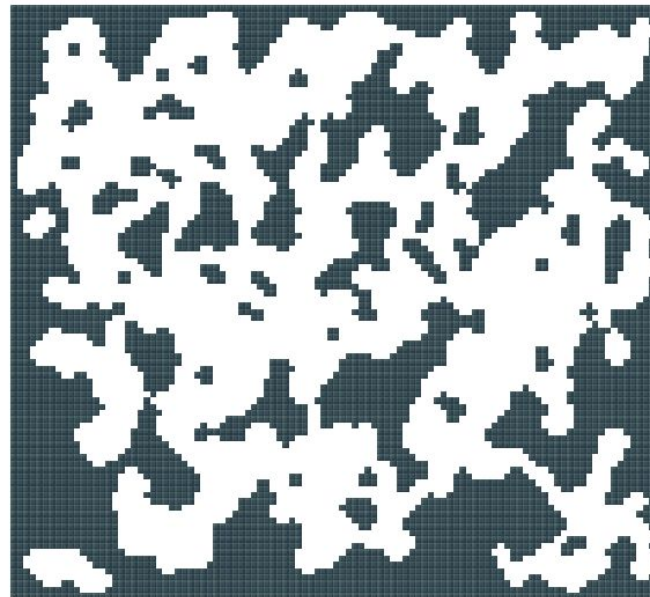
45x30, 2 pases, 40% probabilidad de pared



75x30, 5 pases, 35% de probabilidad de pared



50x50, 5 pases, 40% probabilidad de pared



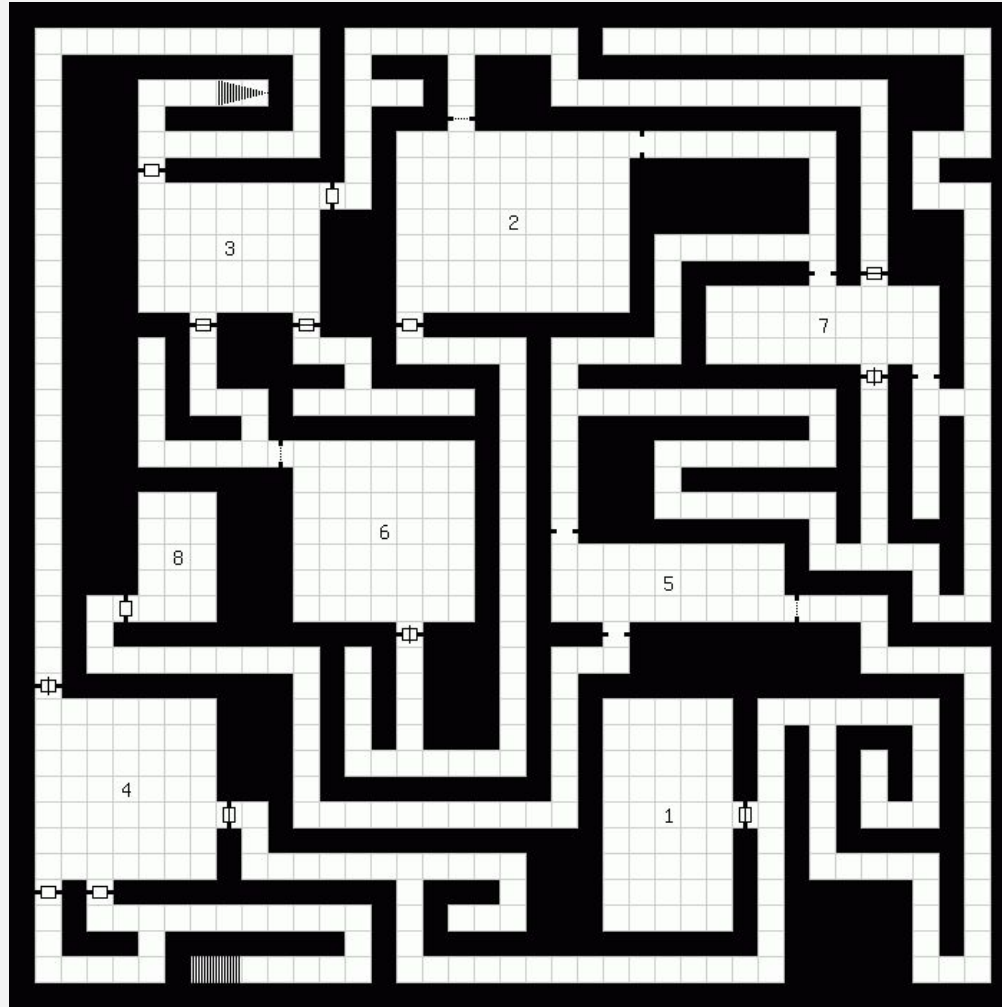
100x90, 8 pases, 40% probabilidad de pared

Autómata Celular. Conclusiones.

Hay ciertos problemas que podemos encontrar con este método:

- Conectividad no asegurada.
- La distinción entre habitaciones y pasillos no está clara, a veces obtenemos espacios demasiado abiertos.
- Los resultados son demasiado imprevisibles. Queremos tener más control sobre los resultados de la generación.
- Se hace demasiado difícil determinar dónde y cómo se colocan los objetos y trampas.

Método aleatorio



<https://donjon.bin.sh/adnd/dungeon/>

Laberinto. Growing Tree

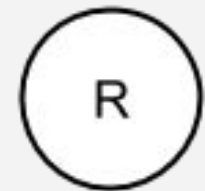
Drunken Walk



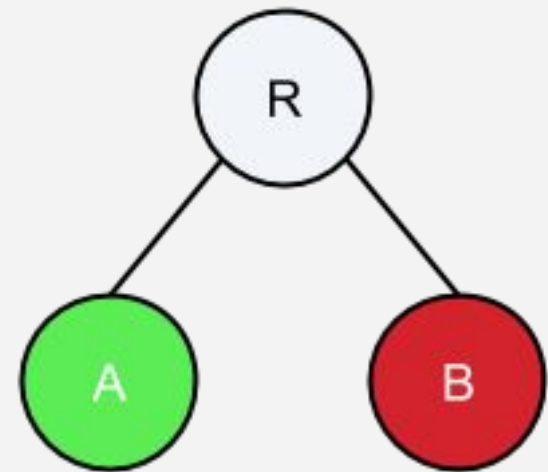
Método aleatorio. Conclusiones

- Se basa en técnicas sencillas.
- Problemas de solapamiento de habitaciones.
- Pasillos demasiado irregulares.
- Requiere demasiada intervención extra.

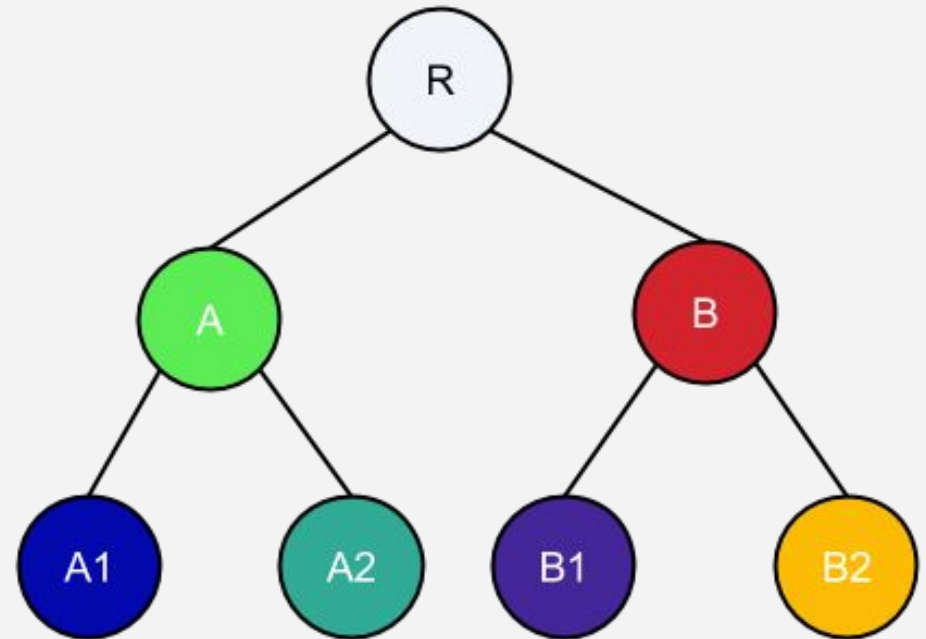
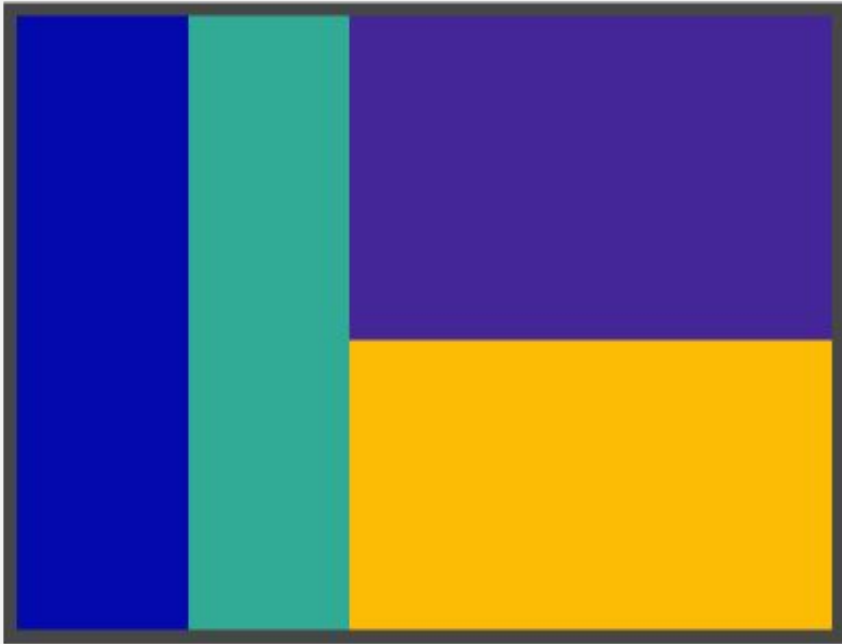
BSP



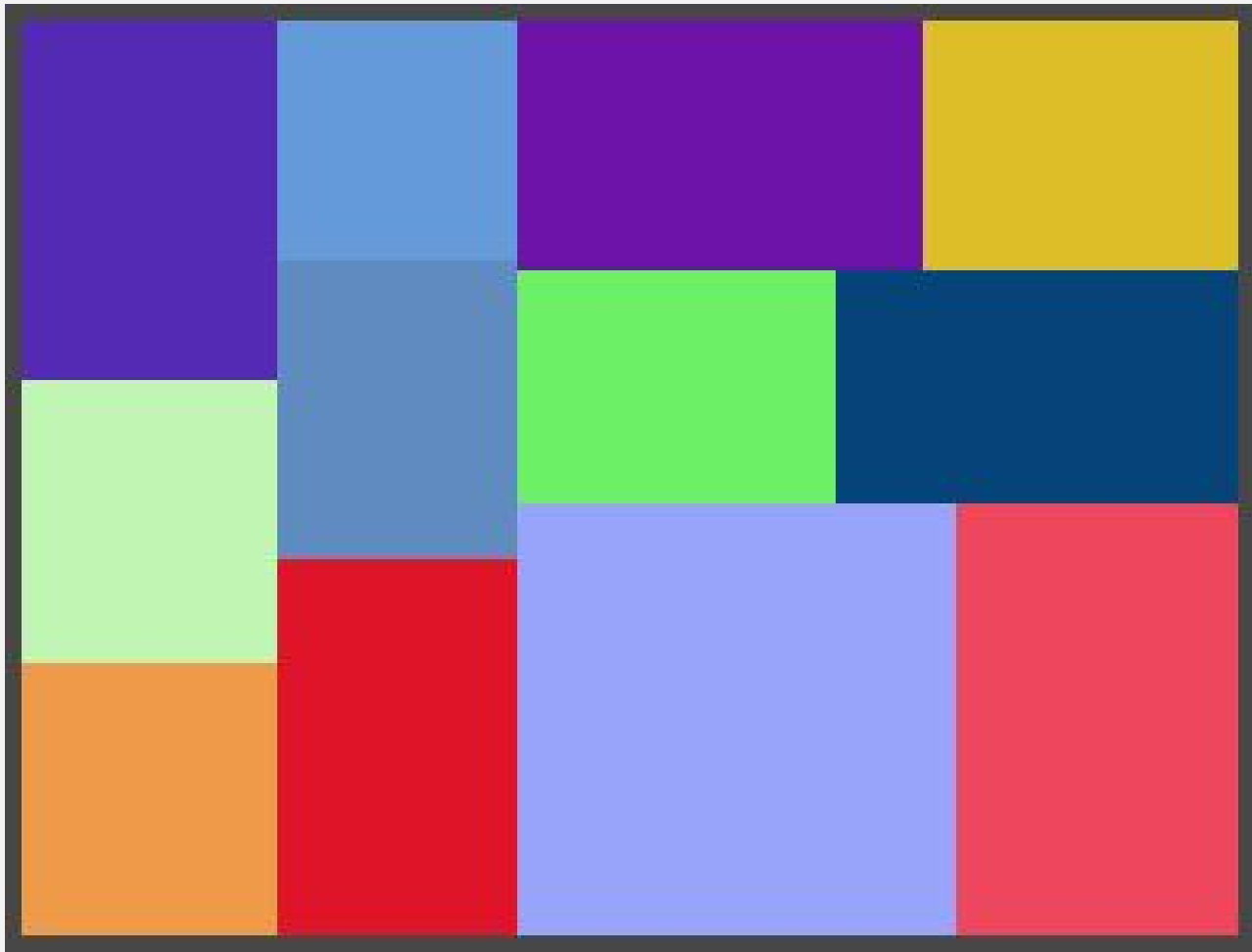
BSP



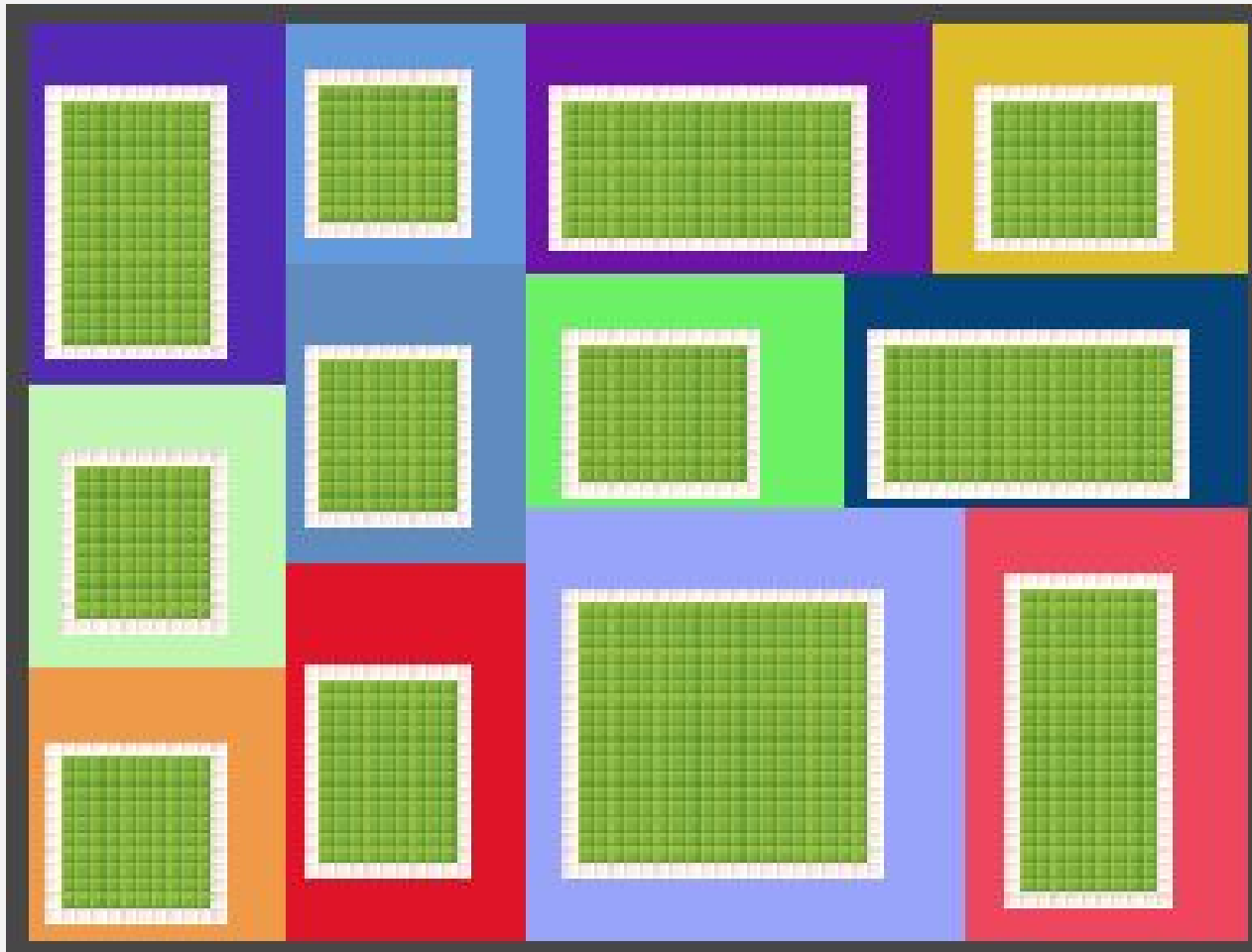
BSP



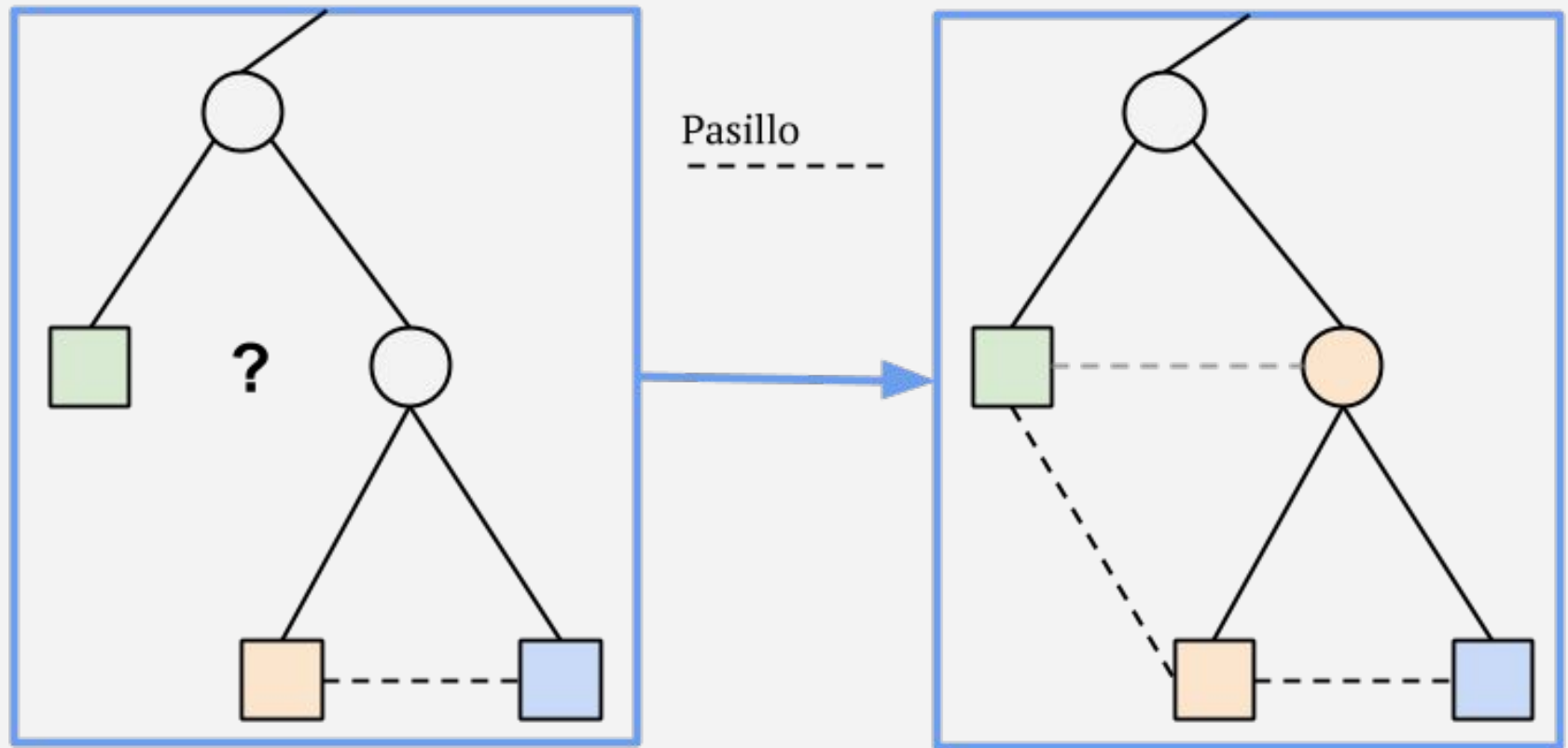
BSP



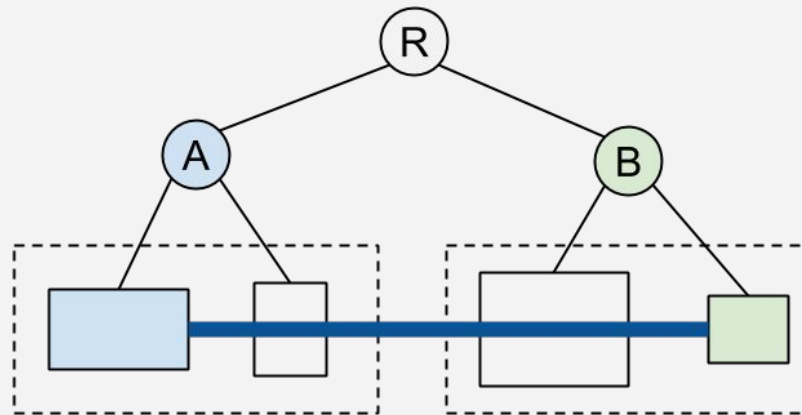
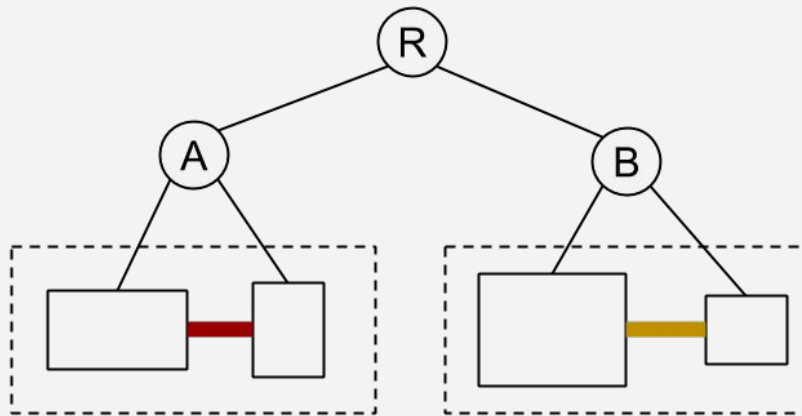
BSP - Habitaciones



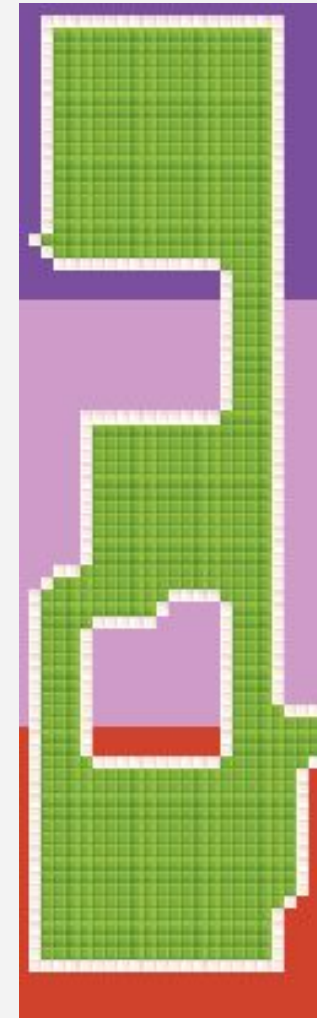
BSP - Pasillos



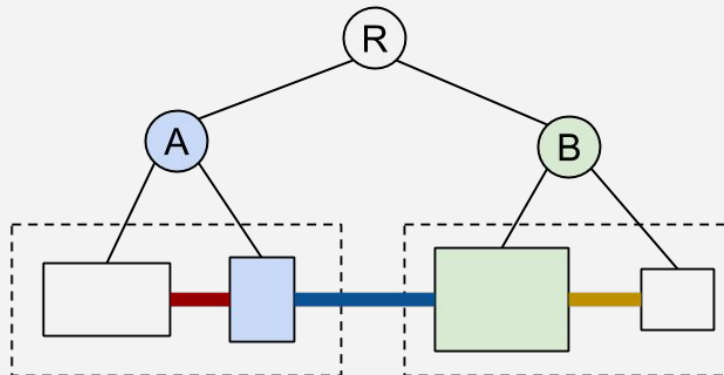
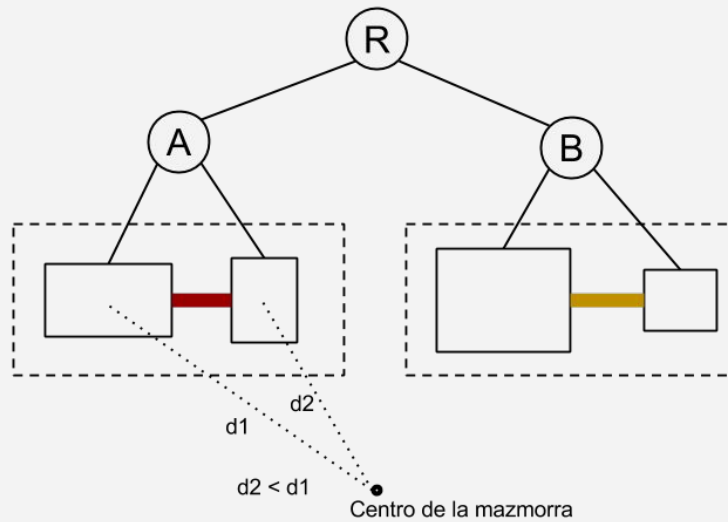
BSP - Pasillos



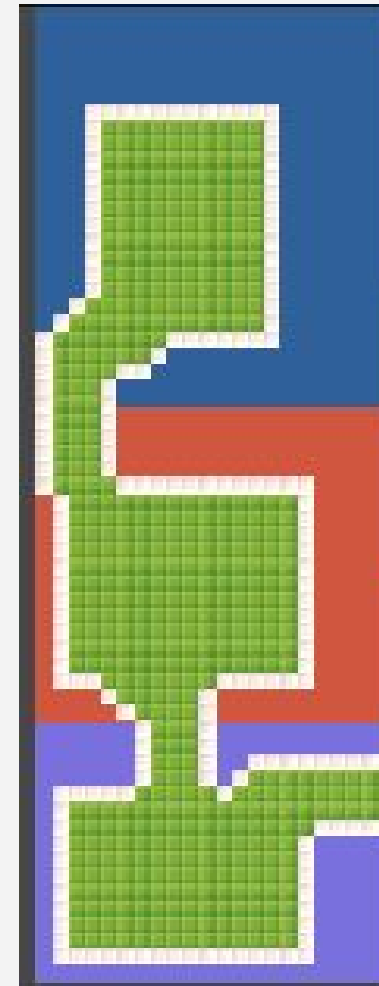
Método A. Aleatorio 50%.



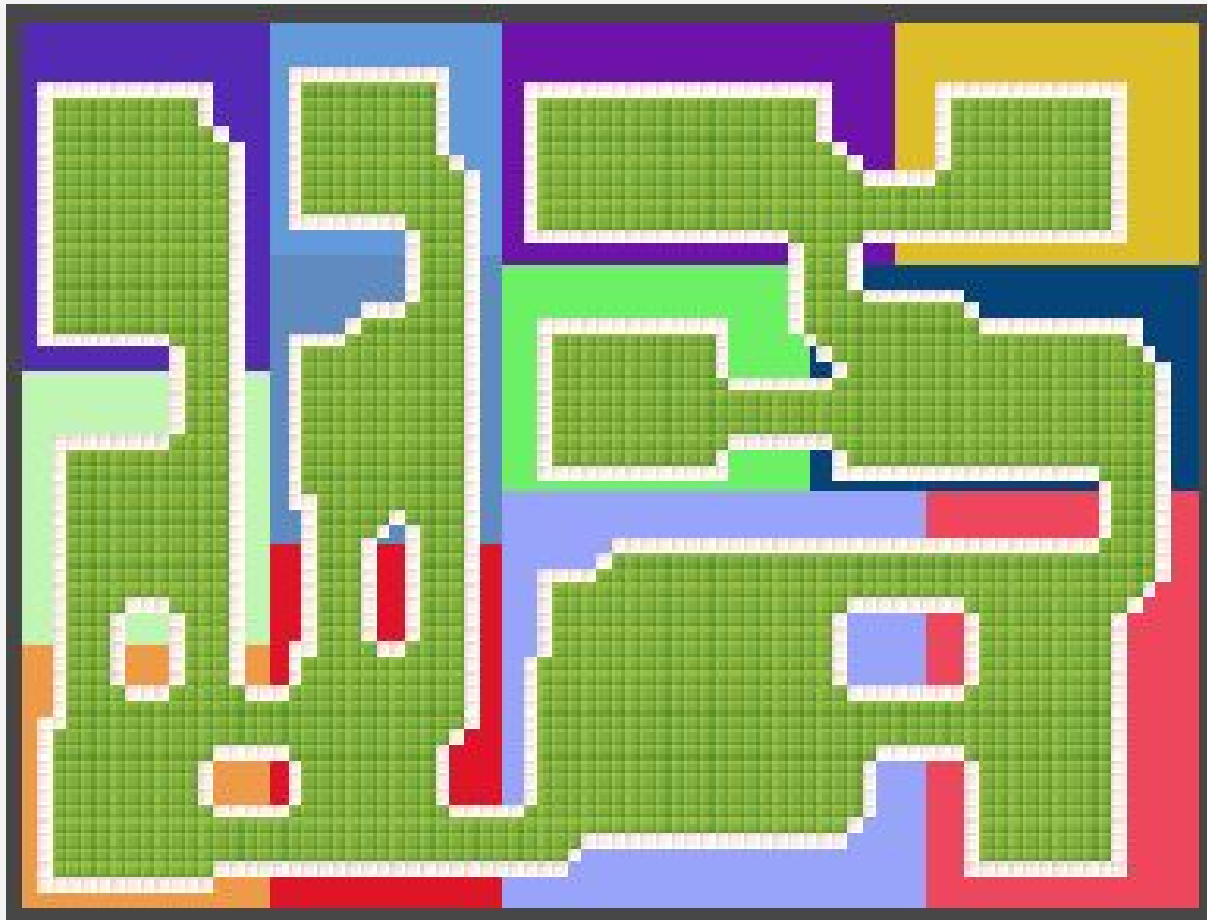
BSP - Pasillos



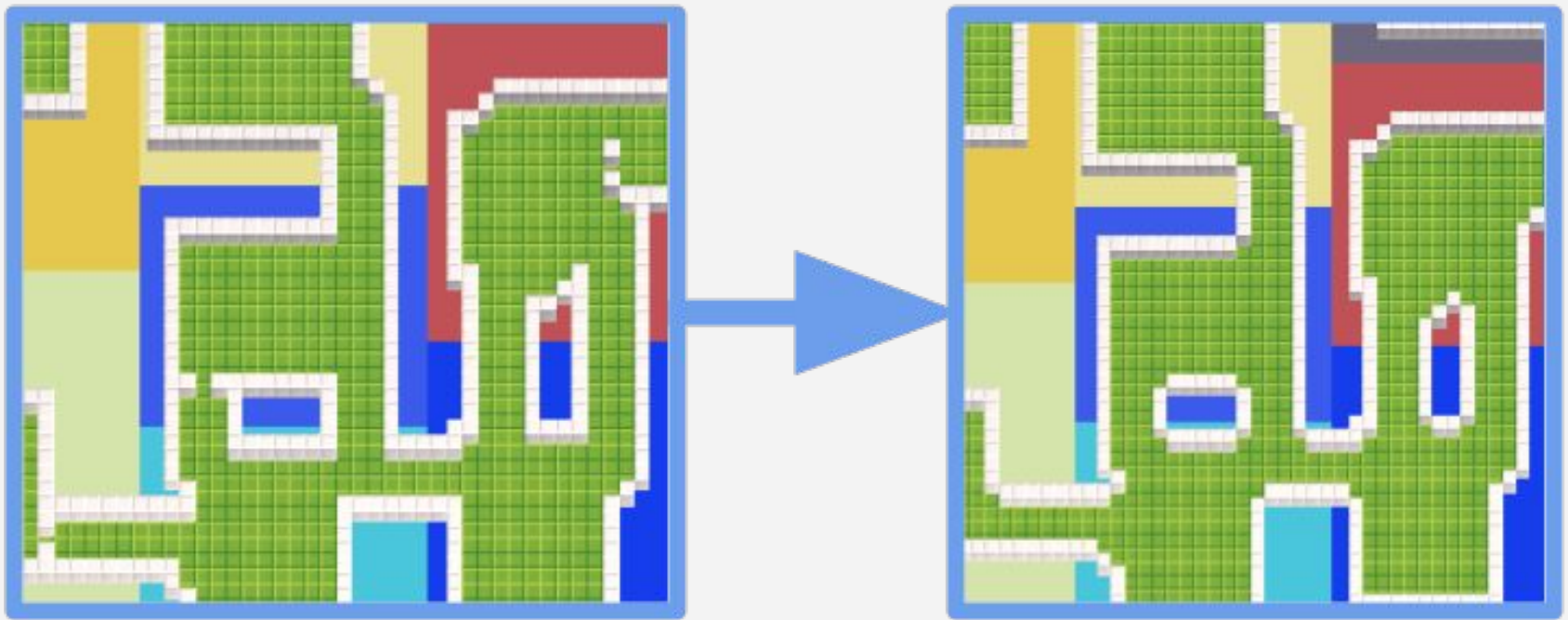
Método B. Distancia al centro.



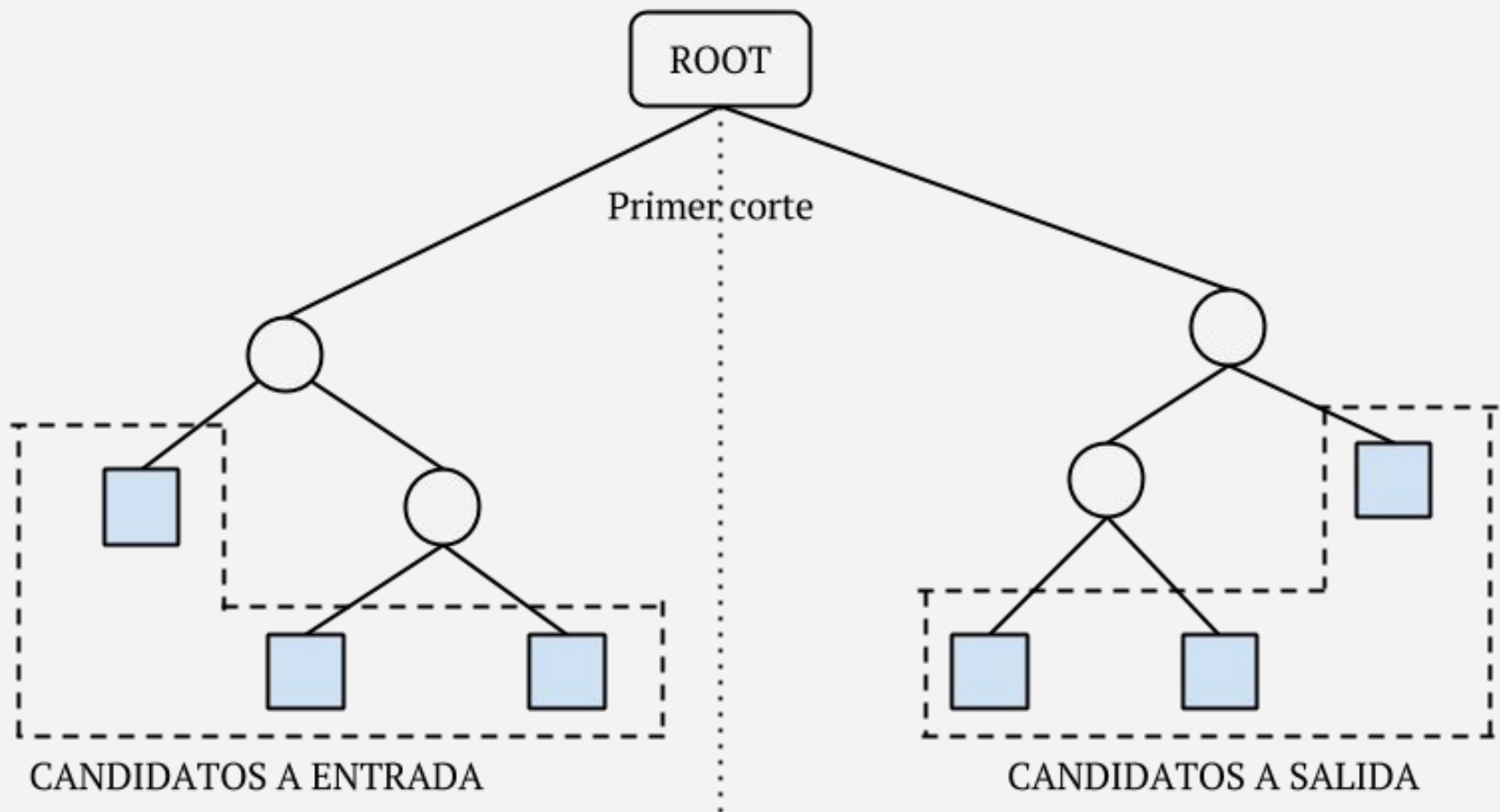
BSP



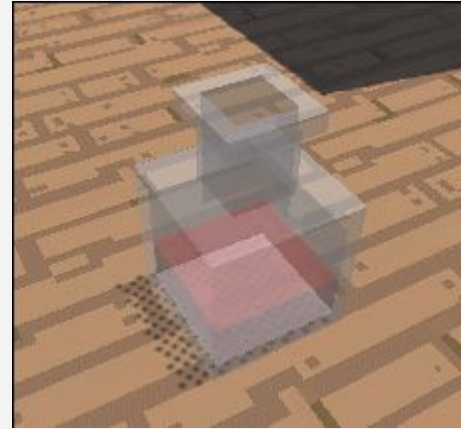
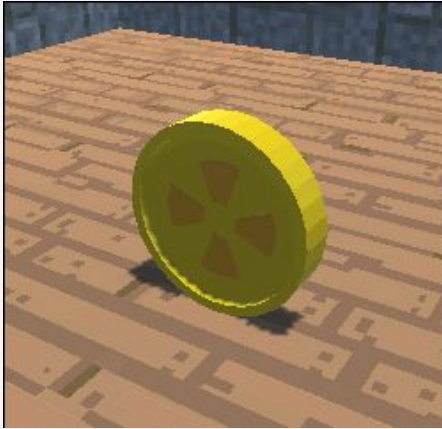
BSP - Autómata celular



BSP - Entrada y salida



Contenido - Objetos



Factor de reducción (R) = 10%, Probabilidad inicial (Pi) = 20%

| NÚMERO DE POCIONES (N) | PROBABILIDAD ($P_i - R * N$) |
|------------------------|--------------------------------|
| 0 | 20% |
| 1 | 10% |
| 2 | 0% |

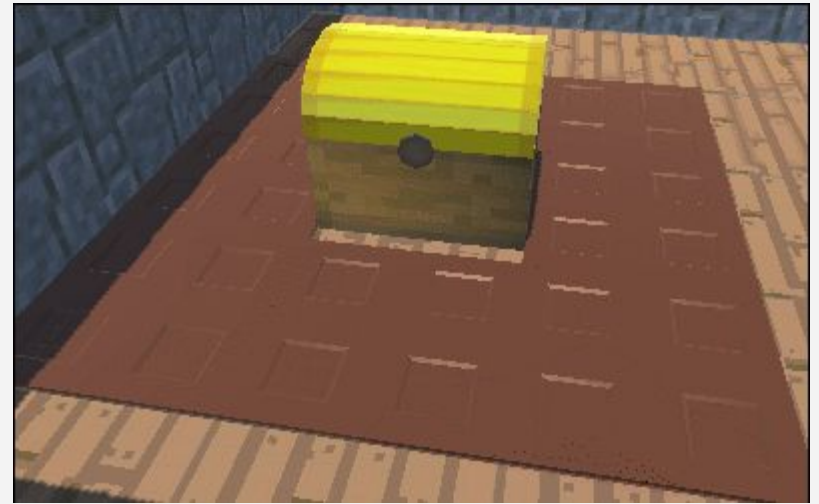
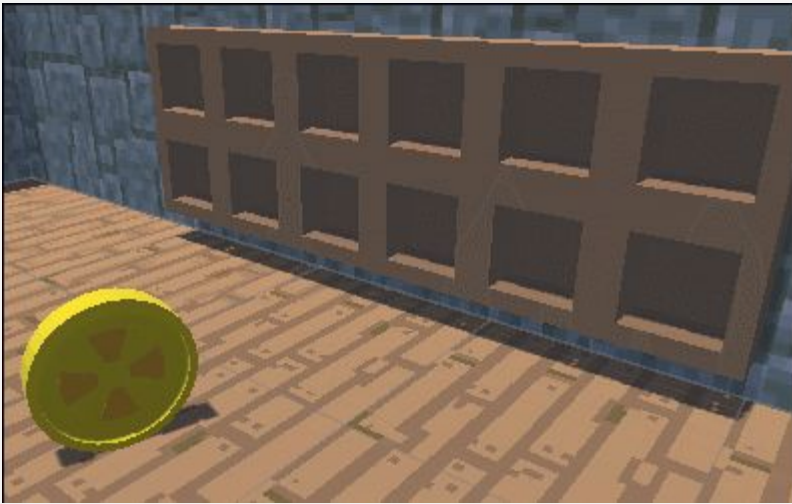
Contenido - Objetos

Factor de reducción (R) = 5%, Probabilidad inicial (Pi) = 20%

| NÚMERO DE COFRES (N) | PROBABILIDAD ($P_i - R * N$) |
|----------------------|--------------------------------|
| 0 | 20% |
| 1 | 15% |
| 2 | 10% |
| 3 | 5% |
| 4 | 0% |



Contenido - Trampas



Contenido - Trampas



Contenido - Trampas



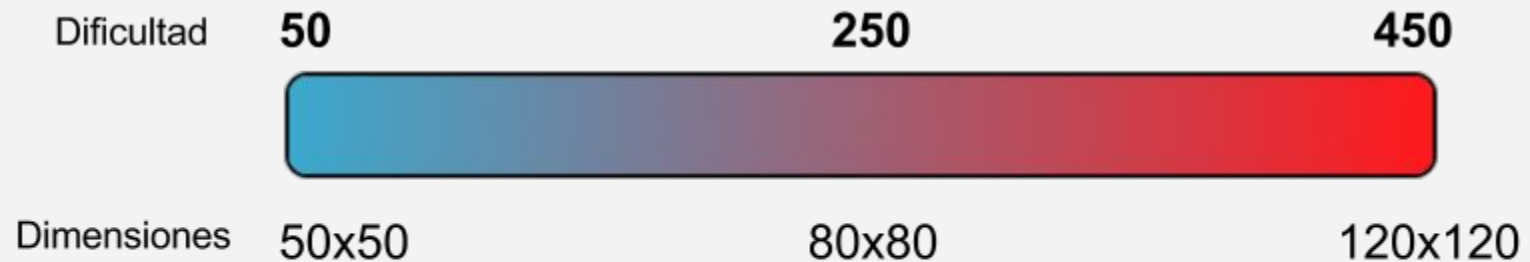
Contenido - Enemigos



Dificultad

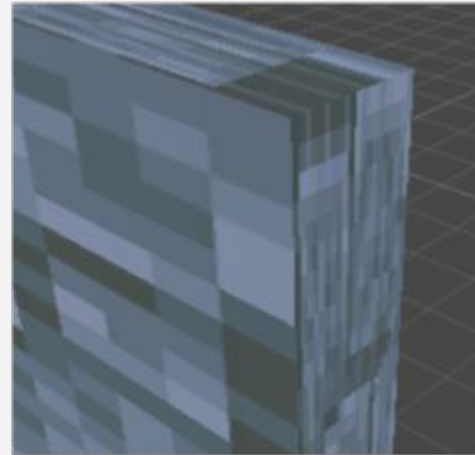
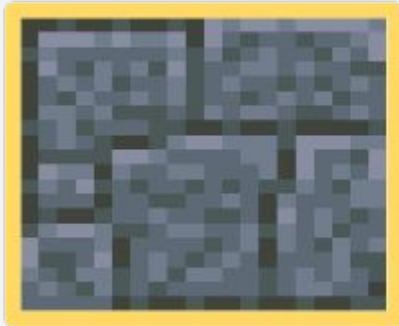
| | |
|----------|----|
| Cangrejo | 5 |
| Goblin | 10 |
| Pociones | 5 |

$\text{Dificultad} = (\text{núm. cangrejos} \times 5 + \text{núm. goblins} \times 10) - (\text{núm. pociones} \times 5)$

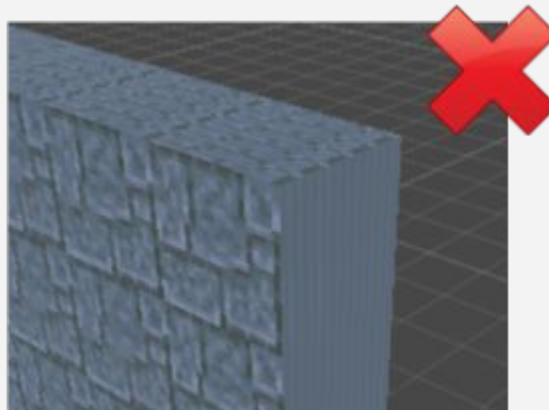


Más enemigos
Más rocas en pasillos
Más habitaciones
Mismo número de pociones

Tiling de texturas.



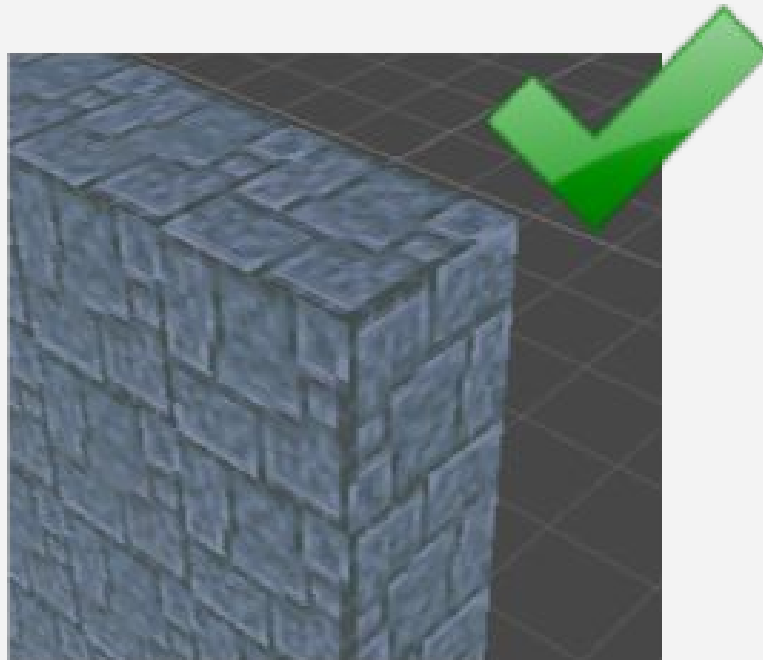
```
material.textureScale = Vector2(localScale.x, localScale.y);
```



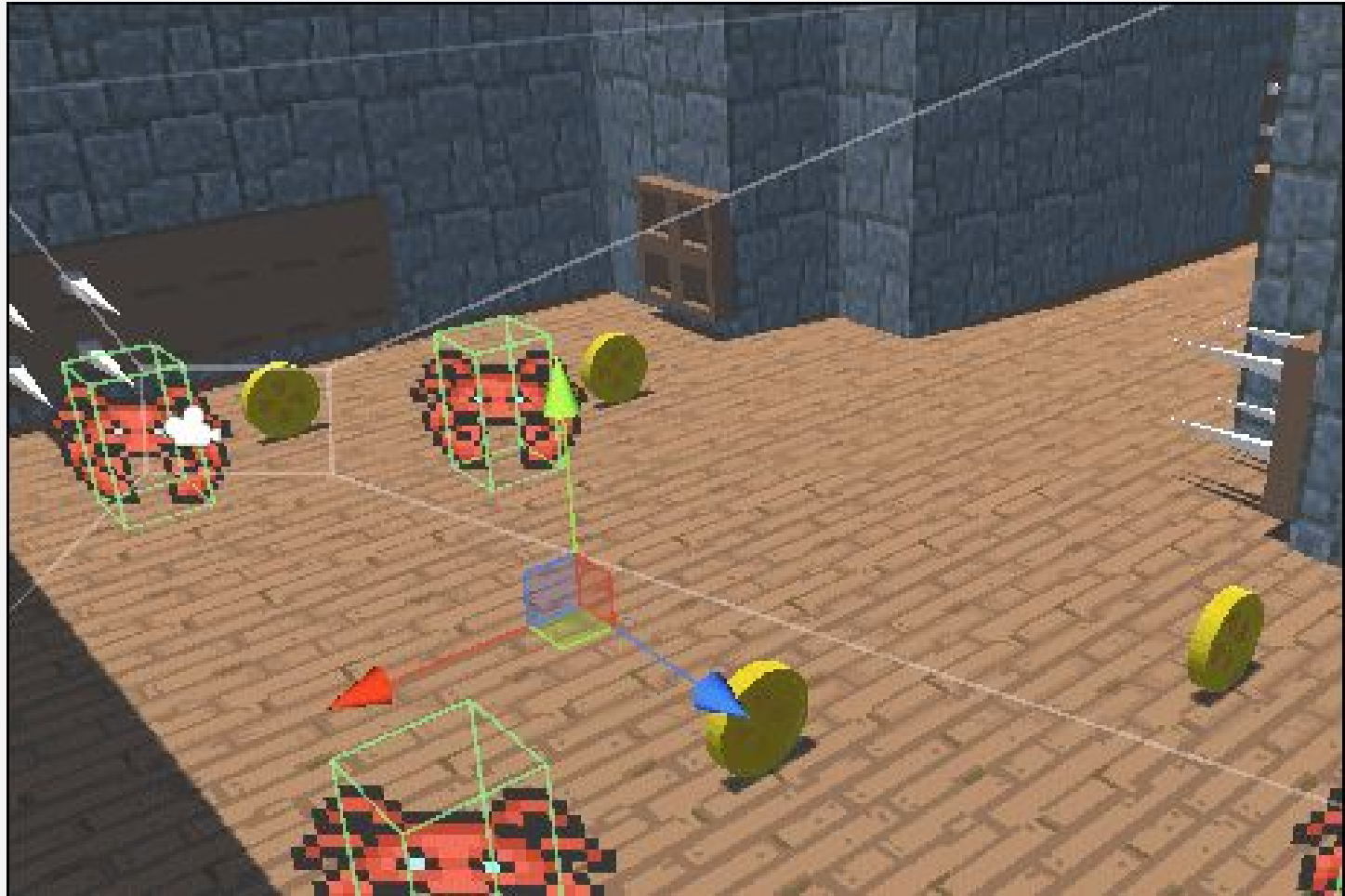
Tiling de texturas.

```
float length = currentScale.x;  
float width = currentScale.z;  
float height = currentScale.y;
```

```
//Top  
mesh_UVs[4] = Vector2( 0, width );  
mesh_UVs[5] = Vector2( length, width );  
mesh_UVs[8] = Vector2( 0, 0 );  
mesh_UVs[9] = Vector2( length, 0 );
```



Billboards



Mejoras.

- Variedad en los sets de escenario.
- Diferentes alturas en la mazmorra.
- Varios pisos en mazmorras.
- Más enemigos y con mejor IA.
- Jefes de mazmorra.
- Items para el jugador. Armas, pociones, etc.
- Llaves y puertas.
- Sombras en los billboards.
- BSP. Oclusión de habitaciones y colisiones.

Gracias por la atención

Rubén Martínez Vilar

Tutores:

Faraón Llorens Largo

Francisco José Gallego Durán



Universitat d'Alacant
Universidad de Alicante