

## C6 - Projeto

### Jogo Clássico : Tower Defense



Objetivo : Torre não pode ser destruída durante o nível

- Ganhar : Acabar o nível com a saúde da torre  $> 0$
- Perder : A saúde da torre  $\leq 0$ .

Interações :

- Colocar/Remover canhões em slots na grid do mapa
- Dar upgrade em um canhão
- 

Elementos Visuais :

Regras do jogo :

Algoritmos / Simulações Físicas

Quais elementos são aleatórios ? (low priority)

- Mapa ?
- Limite Recursos (canhões ou pontos) ?

## Layout (Exemplo):

Mapa

	0	1	2	3
0	W	W	W	W
1	4	3	2	5
2	W	C	1	0
3	W	W	W	W

C: Canhão

E: Enemy

W: Wall

0: onde os inimigos começam

o m<sup>e</sup> mais alto: onde fica a torre

5: slot disponível para colocar um canhão

EnemyPath:  $[(2,3), (2,2), (1,2), (1,1), (1,0)]$

Enemy:

- Posição,  $[0, 1]$ : Posição unidimensional que mapeia numa posição do enemy path.

Neste exemplo, 0  $\rightarrow$  (2,3)

0,6  $\rightarrow$  (1,1)

1  $\rightarrow$  (1,0)

- Velocidade,  $[0, 1]$

- Posição na grid

- Vida

- Pontos: Pontos ganhos pela morte

Canhão:

- Posição na grid, por exemplo (2,1)

- Range: distância que consegue detetar / acertar num inimigo

- Rate/cooldown: Taxa de disparos / unidade de tempo.  
ou tempo que fica à espera até ao próximo tiro.

Torre :

- Posição : Posição na grid, necessariamente, na última posição do enemyPath.
- Vida

Field :

- Mapa [Rows][Cols]
- enemyPath [ ]
- Pontos : Pontos disponível para gastar em canhões ou upgrades.
- Torre
- enemy start Position : para caso queiramos ter uma transição para a 1ª posição do enemy path.
- lista de inimigos com o tempo com que entram na cena

Quais elementos apresentam aleatoriedade?

- Mapa e Enemy Path, estratégias:

1. Berlin Noise com threshold

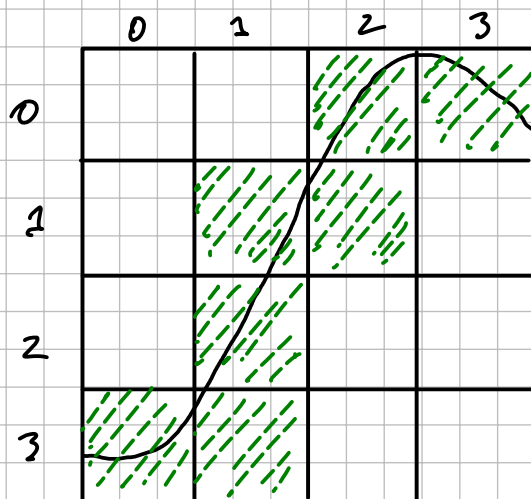
	0	1	2	3
0	.1	.2	.5	.4
1	.3	.5	.7	.7
2	.7	.6	.8	.7
3	.6	.5	.5	.1

$\geq .7$  Por exemplo

2. Equação Matemática alterando os parâmetros

Exemplo:

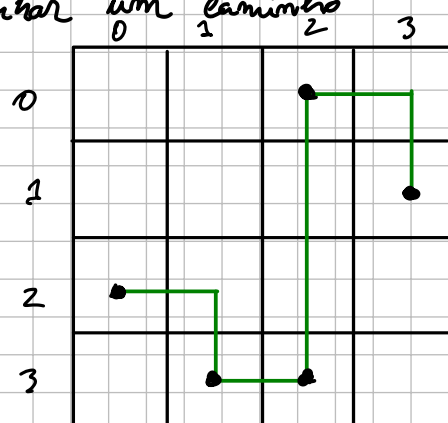
$$f(x) = \sin(ax - b)$$



3. Caminho coluna por coluna:

Iterando coluna a coluna e desenhando um caminho mantendo que ligue esses pontos:

[2, 3, 0, 1]



- Limitar Recursos. Posso limitar a quantidade de canhões
- Gerar inimigos proceduralmente?

↳ 1 approach seria gerar uma quantidade de inimigos de acordo com o total de tiros/s. Ou seja, ter uma relação entre:

Field	Enemy
$\frac{\text{Total dano}}{\text{unidade tempo}}$	$\frac{\text{total de vida}}{\text{Ang (velocidade)}}$

Objetivo: Torre não pode ser destruída durante o nível

- Ganhar: Acabar o nível com a saúde da torre  $> 0$

- Perder: A saúde da torre  $\leq 0$ .

Regras do jogo:

- Proibido posicionar canhões em slots não-livres

- Em slots não-livres, é possível libertar disparando nele (árvore, pedra,...)  
(low priority)

- O canhão dispara no inimigo mais próximo da chegada dentro do seu range

uma das duas [ - Se o inimigo chegar na torre  $\Rightarrow Vida_{torre} -= Vida_{inimigo}$  ;  
- Se o inimigo chegar na torre  $\Rightarrow Vida_{torre} --$  ;

- Se  $Vida_{torre} \leq 0 \Rightarrow$  Game Over e Perder

- Se  $Vida_{torre} > 0 \wedge$  nível acabou  $\Rightarrow$  Ganhou o nível

- Possível dar upgrade em um canhão (low priority)  
 $\hookrightarrow$  Enemy (Range, cooldown)  $\uparrow$

- Se  $Vida_{inimigo} \leq 0 \Rightarrow Pontos_{Field} += Pontos_{inimigo}$

## Interação Visuais

	0	1	2	3
0				
1				
2				
3				

Cursor : posição na janela  
↳ mapear na grid

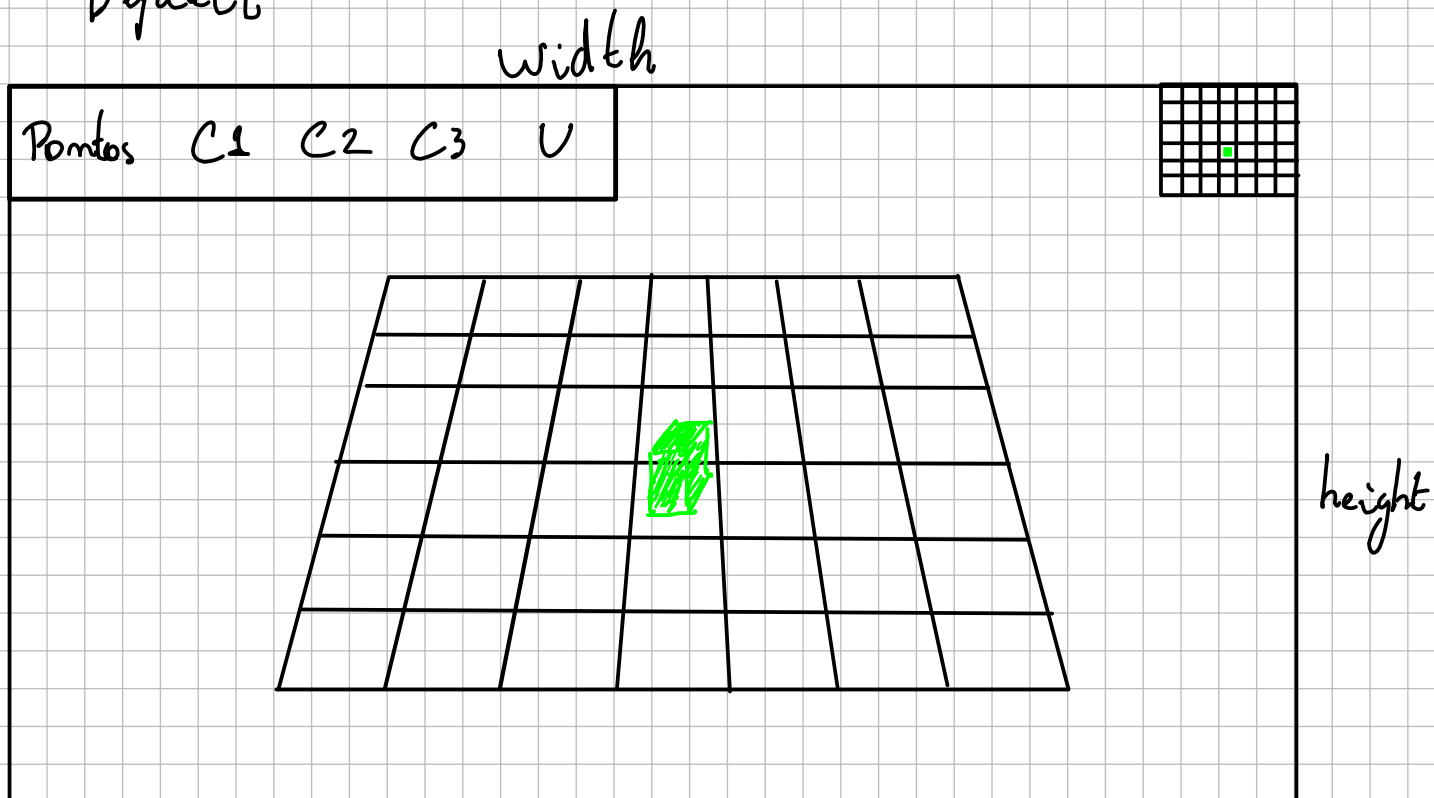
Problema : O mapa não vai, necessariamente, ser projetado ortogonalmente no ecrã. Então pode ser difícil mapear posição janela → posição na grid

## Alternativa :

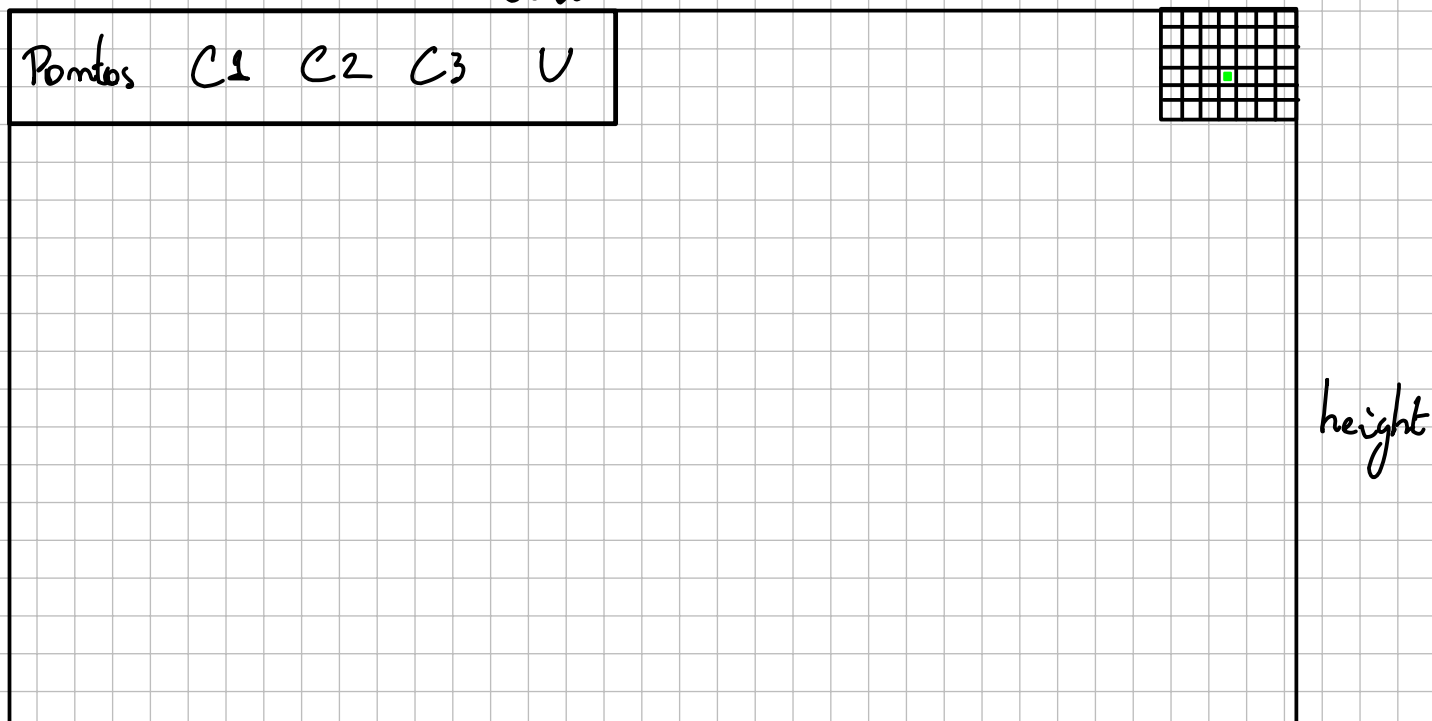
Ter uma posição que representa a slot na grid e utilizo as setas para navegar

- 
- Premir 'P' → Pausar jogo
  - Premir '+' / '-' → Aumentar / Diminuir velocidade do jogo
  - Premir '1', '2', '3' → Colocar um dos cantões predefinidos na posição selecionada se tiver pontos disponíveis
  - Premir 'W', 'A', 'S', 'D' / '↑', '←', '↓', '→' → Navegar o slot selecionado
  - Premir 'u' → Realizar upgrade no cantão do slot selecionado se houver pontos disponíveis
  - Premir 'v' → Alternar view entre default e visão do inimigo mais próximo da chegada

# Elementos Visuais Default

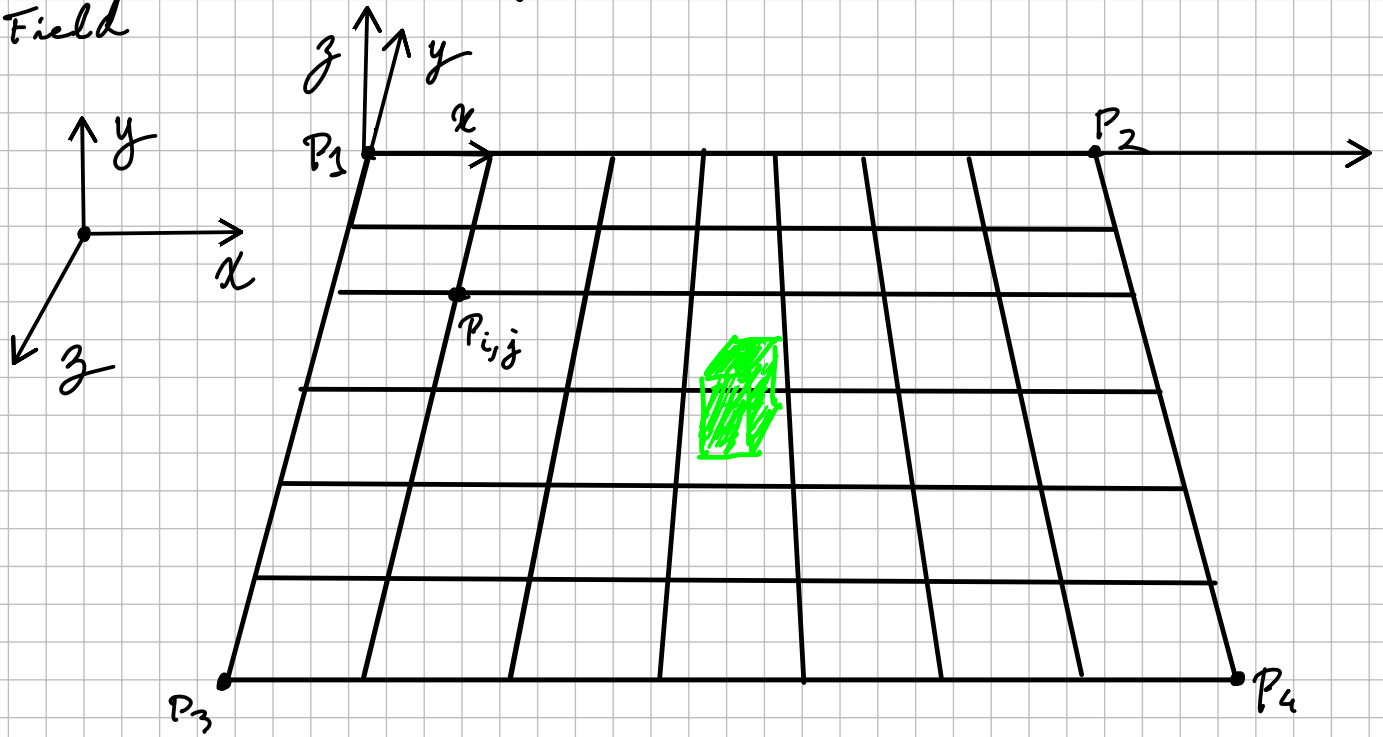


## Perspetiva do mínimo width





Espacio Real dos Objetos  
Field



$i$ : row,  $j$ : 'column',  $dw = \frac{1}{cols}$ ,  $dh = \frac{1}{rows}$

$$P_1 = (-0,5; 0,5; 0)$$

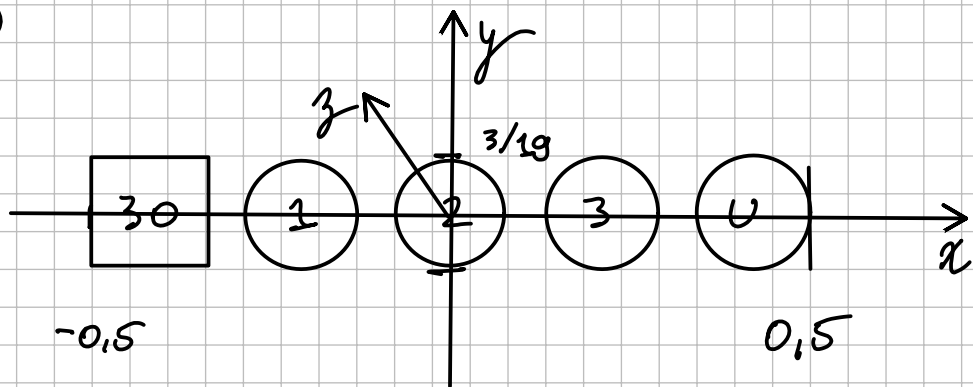
$$P_3 = (-0,5; -0,5; 0)$$

$$P_2 = (0,5; 0,5; 0)$$

$$P_4 = (0,5; -0,5; 0)$$

$$P_{i,j} = (-0,5 + jdw, 0,5 - idh, 0)$$

HVD



# UML

