

Práctica 3 – 2048 Version 2

Grado en Ingeniería Informática
Grado en Ingeniería del Software
Grado en Ingeniería de Computadores

Virginia Francisco Gilmartín
Raquel Hervás Ballesteros


Facultad de Informática
Universidad Complutense



2048

Entrega: 29/03 (23:55)

Cuatro etapas

1. Versión 1 
 - Inicialización, visualización y cargar tablero (aprox. 06/03)
2. Versión 2
 - Deslizamiento de fichas (aprox. 13/03)
3. Versión 3
 - Juego funcionando (aprox. 20/03)
4. Versión 4
 - Mejores puntuaciones y salvar partida (opcional → 27/03)

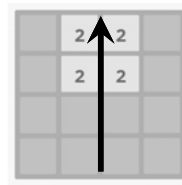
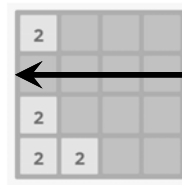
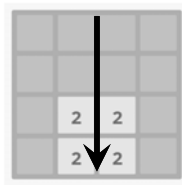
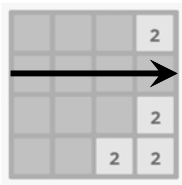


Versión 2: Inclinación del tablero

- ✓ Añade la posibilidad de inclinar el tablero y hacer que las fichas caigan en esa dirección
- ✓ El jugador elige inclinar el tablero en una dirección...



... y todas las fichas se deslizarán en esa dirección



Programa principal

1. Carga / Inicialización tablero
2. Visualización del tablero
3. Solicitamos al usuario la dirección hacia la que inclinar el tablero
4. Si el usuario no ha introducido Esc inclinamos el tablero
5. Volvemos a 2 hasta que el usuario introduzca *Esc*



Lectura teclado del usuario

tAccion leeAccion();

- ✓ Devuelve un tAccion (Arriba, Abajo, Izquierda, Derecha o Salir). El valor Nada sólo se usa internamente (si queréis).

```
...
cin.sync();
key = _getch();
if (key == 0xe0){ //Comprobamos si es una tecla especial
    key = _getch();
    if (key == 72)
        accion = Arriba;
    else if (. . .)
        ...
}
else if (key == 27) //Comprobamos si es Esc
    accion = Salir;
...
```

- ✓ Hay que repetir hasta que se introduzca una tecla válida



mueveFichas: Descripción

- ✓ Desplaza las fichas del tablero en la dirección indicada por accion

tablero



accion

Derecha



tablero



tablero



accion

Arriba



tablero



mueveFichas: Implementación

- ✓ Vamos recorriendo el tablero por filas (Derecha o Izquierda) o columnas (Arriba o Abajo)
- ✓ Desde el extremo hacia el que se inclina hasta el extremo opuesto buscamos las celdas ocupadas y las vamos moviendo hacia el extremo de inclinación

tablero

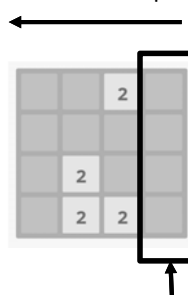


accion

Derecha



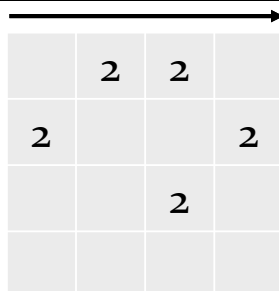
Dirección de búsqueda



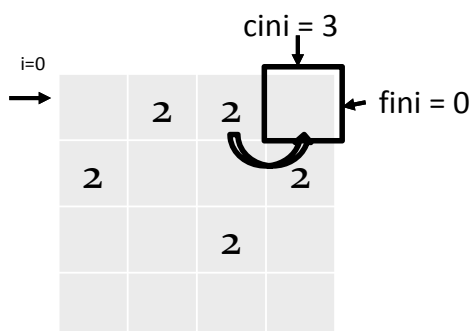
Extremo de inclinación



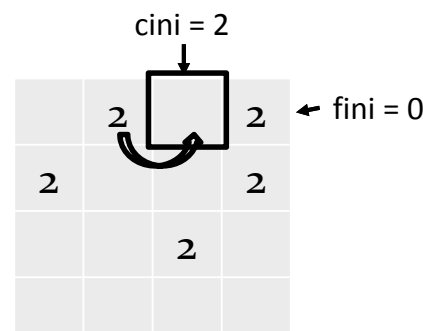
mueveFichas: Implementación



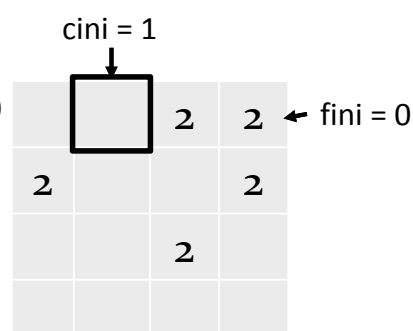
Recorremos las filas con la i



Buscamos casillas ocupadas



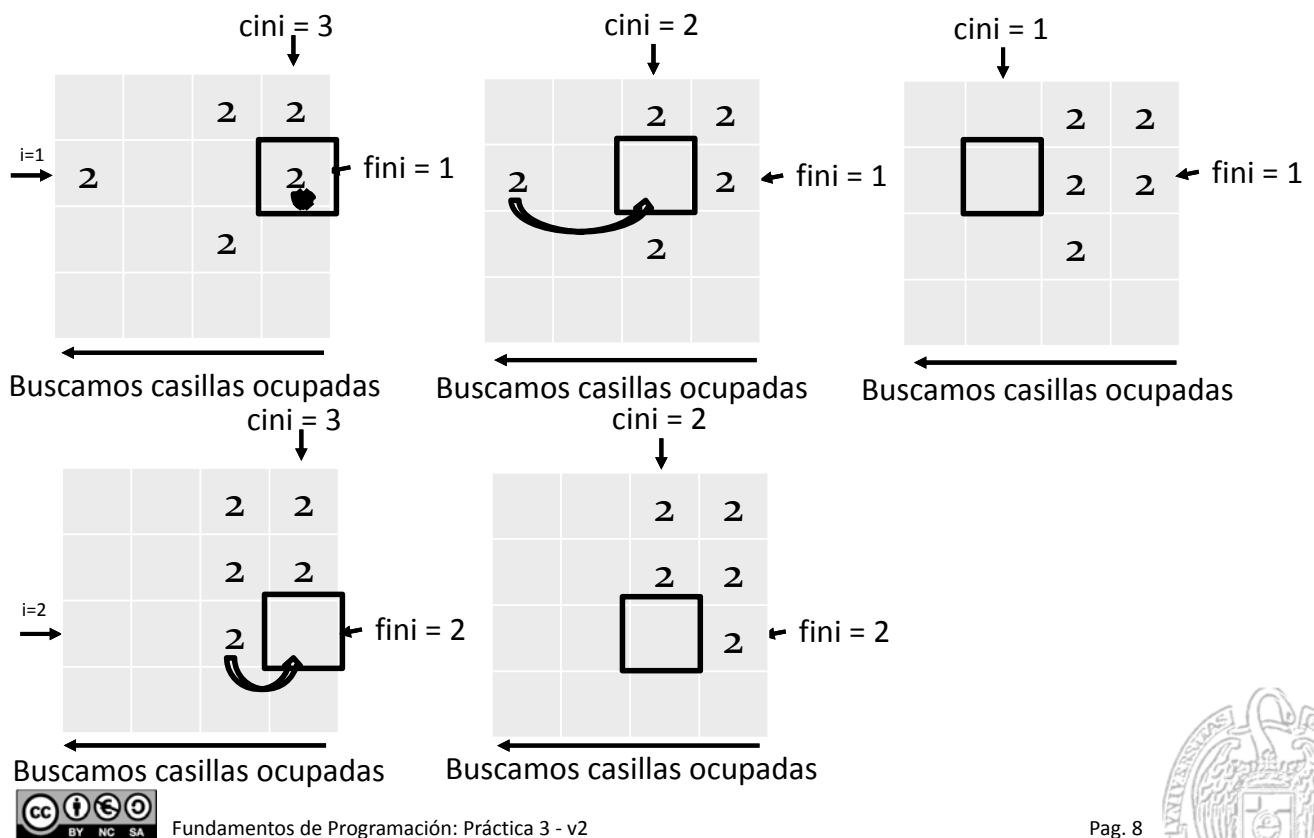
Buscamos casillas ocupadas



Buscamos casillas ocupadas



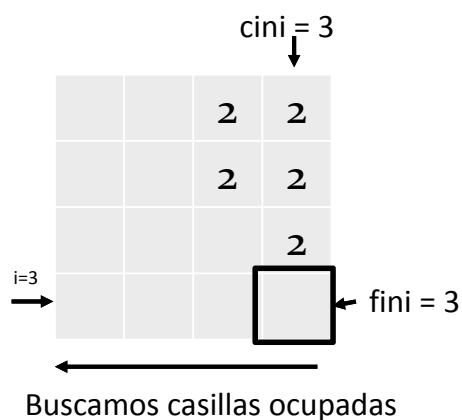
mueveFichas: Implementación



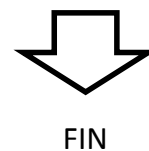
Pag. 8



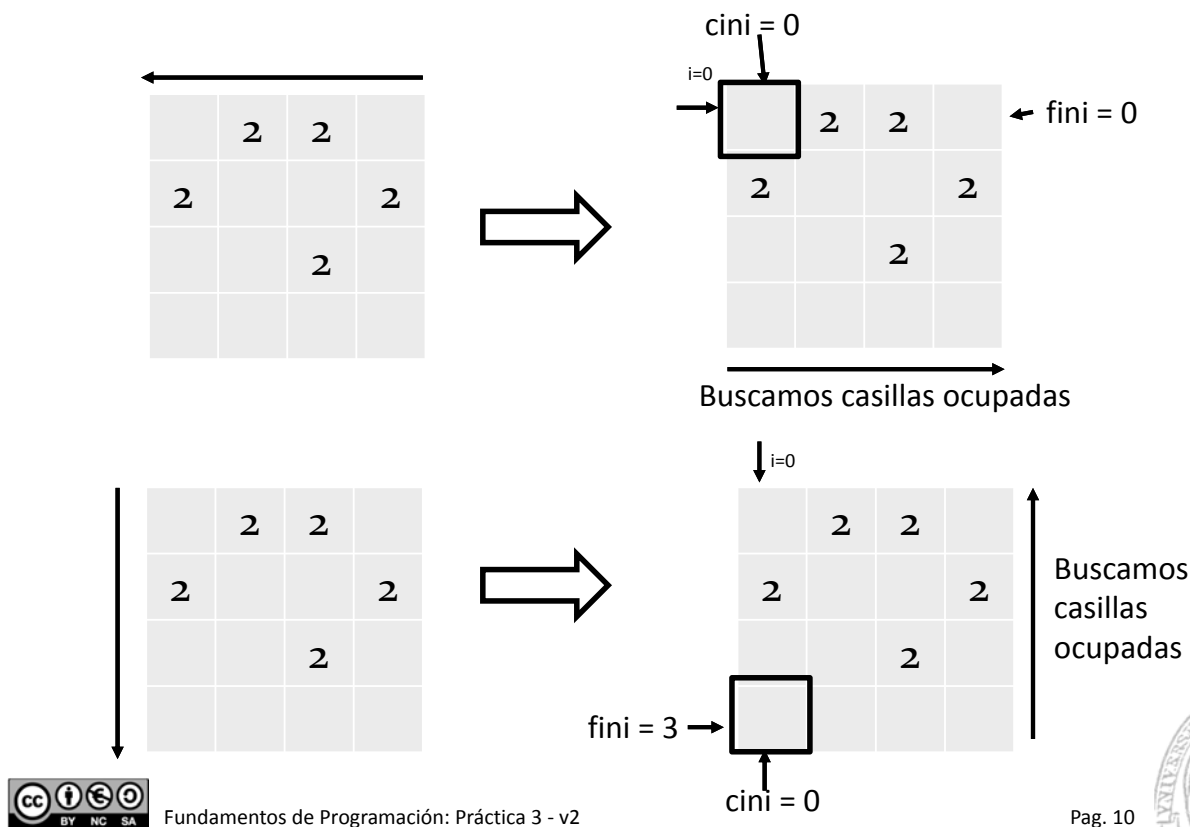
mueveFichas: Implementación



Cuando se termina con la fila $i == N-1$ (3)



mueveFichas: Implementación



Cuatro posibles movimientos

Según la dirección:

- ✓ Los valores de $fini$ e $cini$ serán diferentes
- ✓ Las direcciones para movernos por el tablero serán diferentes
- ✓ Por ejemplo, si Derecha:
 - $fini = \text{fila en la que estamos}$
 - $cini = N - 1$
 - Al movernos, la fila se queda igual y las columnas se decrementan
- ✓ Por ejemplo, si Arriba:
 - $fini = 0$
 - $cini = \text{columna en la que estamos}$
 - Al movernos, las filas se incrementan y la columna se queda igual

Subprograma auxiliar para mover fichas

```
void obtenParam(tAccion accion, int i, int &fini, int &cini, int &incr f, int &incr c);
```

Según la dirección y el contador del recorrido establece:

- ✓ Casilla inicial en la que comenzar a buscar las casillas ocupadas (**fini** e **cini**)
- ✓ Incrementos para movernos en el tablero buscando las casillas ocupadas (**incr f** e **incr c**)



Subprograma que mueve todas las fichas

```
void mueveFichas(tTablero tablero, tAccion accion)
```

Recorrido de **i** de 0 a N-1

1. Obtenemos los valores de **fini**, **cini**, **incr f** e **incr c**
2. Búsqueda para encontrar una celda ocupada en la fila o columna
 - ✓ Comenzamos la búsqueda en $(f, c) = (fini, cini)$
 - ✓ Avanzamos con **incr f** e **incr c**
 - ✓ $(f, c) = (f + incr f, c + incr c)$
3. Si encontramos una celda ocupada:
 - a. La movemos a la celda **(fini, cini)**
 - b. Avanzamos **fini** e **cini** para volver a iniciar la búsqueda (volvemos al paso 2)
 - ✓ Avanzamos con **incr f** e **incr c**



Depuración

¡Prueba las demos!

¡Prueba tu programa con distintos casos de uso!

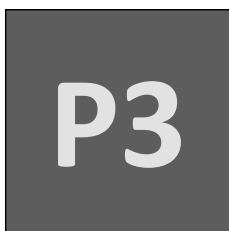


Fundamentos de Programación: Práctica 3 - v2

Pag. 14



Fundamentos de Programación



Práctica 3 – 2048 Version 2

Grado en Ingeniería Informática
Grado en Ingeniería del Software
Grado en Ingeniería de Computadores

Virginia Francisco Gilmartín
Raquel Hervás Ballesteros
Facultad de Informática
Universidad Complutense

