

Pier Point Shipping Inc.

Software Design

Contributor: Chun-Ying Wang, Andy Chen, Daniel Yao, Jeffrey Hwang

Manifest

read
write
understand
edit

Symbols

unused → available to use
nan → can't use
[name] → can add stuff, boat structure

Task

↳ reflect the current loading state on the boat

Ways → ① Live update
② Overall update (最後有)

Assumption *as a required idea*

- come with in provided format

① Read in provided format, rows:

- 8x12
- NAN blocks not sure

→ 我猜我要 build up the 2D Array ⇒ ① [x, y], 空

① —.txt

順序: $m \Rightarrow$ standardize
 $m \Rightarrow$ variable

① [x, y], 空

② { ##### }, —

③ XXXXXX

UNUSED

NAN

② 2D Array

- 2D Array (fixed-size)
- class (for data) → open access

- 先建立 8x12 Array

- read location data

- each location ⇒

store (loop)

- weight → editable
- name (type)

③ Update

↳ balance / load unload 之後 · for loop

- 建立新的 OUTBOUND.txt → 用 loop go over, write

④ Output · file output

输出到指定资料夹

Classes

Array.cpp / .h

- Create the array
- able to access by other operation

START: clean 8x12 array

FINISH: stored data from Input

most

only for creating & storing

2D array

ESTIMATE

ARRAY: 1 HR

I/O : 30 mins

Debug : 2 HRC

File I/O .cpp / .h

- Input manifest
 - ① access array
- Output manifest
 - ② Store data into array

OUTBOUND.txt

Task: Input/Output Manifest

[Input]

- Read all text
- Parse weight & names
- Store into in 2D array

[Output]

- Traverse the array
- Write formatted data

[flow] [main.cpp]

INPUT

initialize/generate 2D array



read "manifest.ext"



Since it's definitely by order,
read & store the weight & name



Input done ✓

OUTPUT

traverse thru the array



generate "OUTBOUND.txt" with

- location
- weigh
- txt info



output

Unload: A, B
Load: C, D

		h(35)
1. Offload A	(1,1)	$3+2=5$
2. E to (1,1)		2
3. Offload B		$4+2=6$
4. C to (2,1)		$2+2=4$
5. D to (3,1)		$1+2=3$
		20

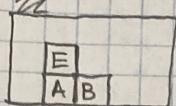
	h(35)
1. F (2,2) \rightarrow (1,3)	2
2. E (2,1) \rightarrow (2,3)	2
3. Off A	$3+2=5$
4. Off B	$4+2=6$
5. C to (1,1)	$3+2=5$
6. D to (2,1)	$2+2=4$
	24

$E(2,1) \rightarrow C$

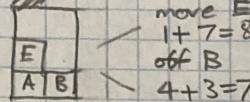
$$\begin{aligned} & \text{ex 1. } h = 3+4+2+1 = 10 \\ & E(2,2) = \\ & \rightarrow (2,1) \rightarrow (1,3) \\ & 3+4+1+3 \quad 3+4+2+1 \end{aligned}$$

★ Heuristic (Offload): Dist. to top left unload square

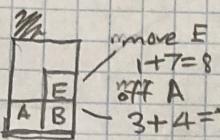
$$\text{ex. } h(A) = 3 + \cancel{2} \quad h(B) = 4 + \cancel{2} \quad \cancel{2} \text{ only matters w/ buffer}$$



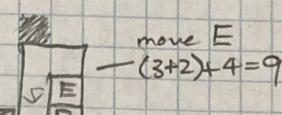
E crates to ?



$$\begin{aligned} & \text{move E} \\ & 1+7=8 \\ & \text{off B} \\ & 4+3=7 \end{aligned}$$



$$\begin{aligned} & \text{move E} \\ & 1+7=8 \\ & \text{off A} \\ & 3+4=7 \end{aligned}$$



(Any col)

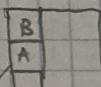
Moves: Choose crate on top or

Move to valid pos

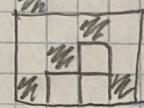
$$\begin{aligned} & \text{Get crate to load} \\ & (8 \times 12) \\ & \text{Max num} = 12 + \# \text{ of load crates} \end{aligned}$$

Max num = 13

→ don't change each move

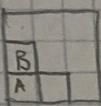


$$\begin{aligned} & 1+2+1 \\ & = 4 \end{aligned}$$

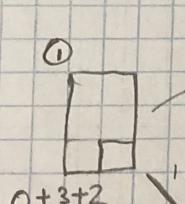


★ (for now)

Heuristic (Onload): 1 for each crate to load or

manhattan dist. to pos. ex. (3, 3, 3, 6)
- scale horizontal movement differently?
 \downarrow
 h will change
each move and
for each
crate
↓
don't count
crates to (so subtract)
unload


3+2=5



$$\begin{aligned} & 1+2+1 \\ & \textcircled{2} \end{aligned}$$

$$\begin{aligned} & 3+1 \\ & \textcircled{3} \end{aligned}$$

$$\begin{aligned} & 4 \\ & \textcircled{4} \checkmark \end{aligned}$$

$$\begin{aligned} & \text{A E} \\ & 6 + (3+2+4) = 12 \end{aligned}$$

0+3+2

3+2

$$\begin{aligned} & \text{ex.} \\ & \text{or} \end{aligned}$$

$$\begin{aligned} & \text{C, D} \\ & 2+1 \end{aligned}$$

①

1+2+1

3+2

4+3

5+4

6+5

7+6

8+7

9+8

②

1+2+1

3+2

4+3

5+4

6+5

7+6

8+7

9+8

③

1+2+1

3+2

4+3

5+4

6+5

7+6

8+7

9+8

④

1+2+1

3+2

4+3

5+4

6+5

7+6

8+7

9+8

⑤

1+2+1

3+2

4+3

5+4

6+5

7+6

8+7

9+8

⑥

1+2+1

3+2

4+3

5+4

6+5

7+6

8+7

9+8

⑦

1+2+1

3+2

4+3

5+4

6+5

7+6

8+7

9+8

⑧

1+2+1

3+2

4+3

5+4

6+5

7+6

8+7

9+8

⑨

1+2+1

3+2

4+3

5+4

6+5

7+6

8+7

9+8

⑩

1+2+1

3+2

4+3

5+4

6+5

7+6

8+7

9+8

⑪

1+2+1

3+2

4+3

5+4

6+5

7+6

8+7

9+8

⑫

1+2+1

3+2

4+3

5+4

6+5

7+6

8+7

9+8

⑬

1+2+1

3+2

4+3

5+4

6+5

7+6

8+7

9+8

⑭

1+2+1

3+2

4+3

5+4

6+5

7+6

8+7

9+8

⑮

$$\begin{array}{c}
 \text{E} - \textcircled{1} \\
 \text{A, B}
 \end{array}
 \xrightarrow{0+(4+4+1+3)=12}
 \begin{array}{c}
 \text{F} - \textcircled{2} \\
 \text{A, B}
 \end{array}
 \xrightarrow{1+(3+5+2+1)=12}
 \begin{array}{c}
 \text{E} \\
 \text{B}
 \end{array}
 \xrightarrow{4+(5+3+2)=14}$$

$$\begin{array}{c}
 \text{A, B, E} \\
 3+(3+4+2+1)=13
 \end{array}
 \quad
 \begin{array}{c}
 \text{E} \\
 \text{A}
 \end{array}
 \quad
 \begin{array}{c}
 \text{E} \\
 4+(4+1+3)=12
 \end{array}$$

$$\begin{array}{c}
 \text{A, B} \\
 0+(3+5+1+1)=10
 \end{array}
 \xrightarrow{1+(4+4+2)=11}
 \begin{array}{c}
 \text{E} \\
 \text{B}
 \end{array}
 \xrightarrow{3+(5+2)=10}
 \begin{array}{c}
 \text{E, B} \\
 5+(4+2)=11
 \end{array}
 \quad
 \begin{array}{c}
 \text{A, B, E} \\
 2+(3+4+2)=11
 \end{array}$$

A* Star
 - Check goal state ✓
 - Map (already explored state) ✓
 - Frontier += all moves
 - Heuristic
 ↗ $f = g + h$ (cost heuristic)
 ↗ Could update this

off: E
 on: F
 $\xrightarrow{\text{ops.}}$
 A $\rightarrow (2, 2)$ $(3, 3)(1, 4)(2, 5)$, off
 B $\rightarrow (2, 1)(3, 3), (1, 4)(2, 5)$, off
 E $\rightarrow (2, 1), (3, 2), (1, 4)(2, 5)$, off
 D $\rightarrow (2, 1)(2, 2)(3, 3)(1, 4)$, off
 (Or) F $\rightarrow (2, 1)(2, 2)(3, 3)(1, 4)(2, 5)$, off
 25 moves

implicitly move objects and then
→ $\left[\left(\text{container}_i, 1\text{-path} \right)^{[4,0]}, \left(\text{Container } 2, 2\text{-path} \right) \right] + (\text{target}, \text{target-path})$ $\nearrow \text{last element}$

return:

movesMade $\left[\text{start}_{\text{container}.row}, \text{start}_{\text{container}.col}, \text{end}_{\text{target}.row}, \text{end}_{\text{target}.col}, (\text{Container } 2, \dots) \right]$

1. findBestMove \rightarrow target.position
2. findObstacles \rightarrow $\left[(\text{row}_1, \text{col}(1)), (\text{row}_2, \text{col}(2)), \dots \right]$, element number = how many obstacles
3. For r in range ObstacleNum
FindObPath(\checkmark)
Animation $\left[\text{ex: 3} \right]$