Thu15j

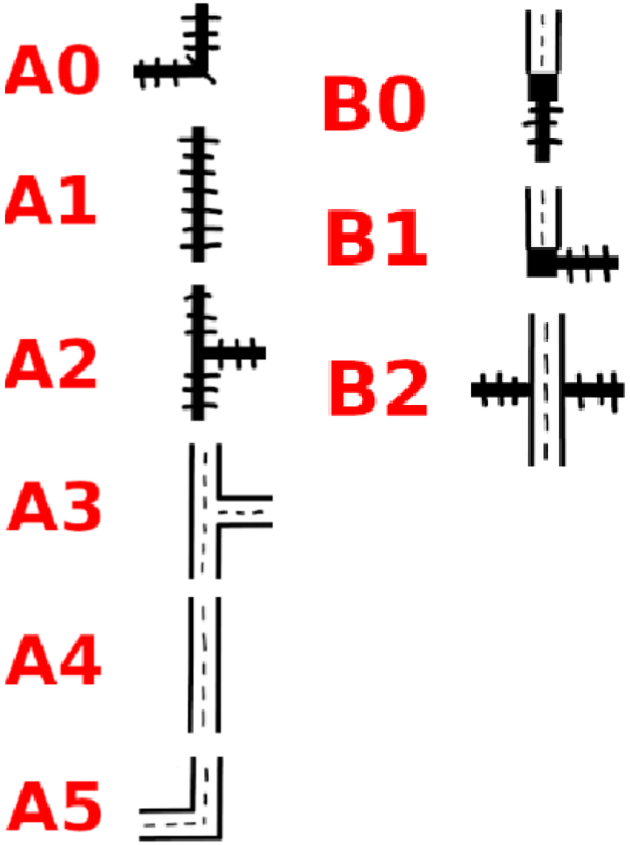
JINGYU HAN – U6764688

TIANJUN PENG – U6761542

HENGJIA ZHANG - U6658734

Dice

Encode a die



Faces: imaging a die with n faces, each face represents a tile shape.

Die A is of 6 faces (0 to 5).

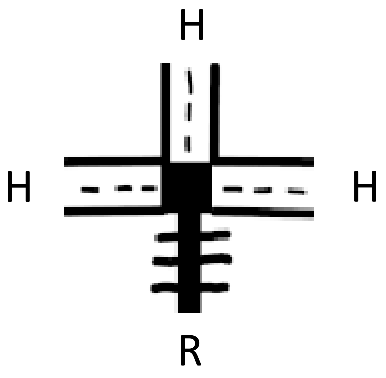
Die b is of 3 faces (0 to 2).

Times: "Each round, die **A** will be rolled three times and die **B** will be rolled once"

Use For loop and Math.random() to roll a die n times

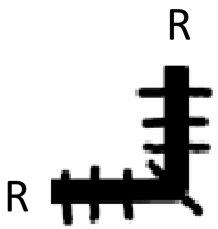
Tile shape

How to define a Tile type?



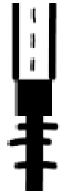
= "HHRH"

- Anticlockwise
- From the top port to the right port

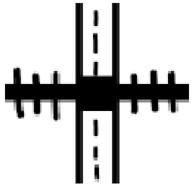


= "RR00"

More examples:



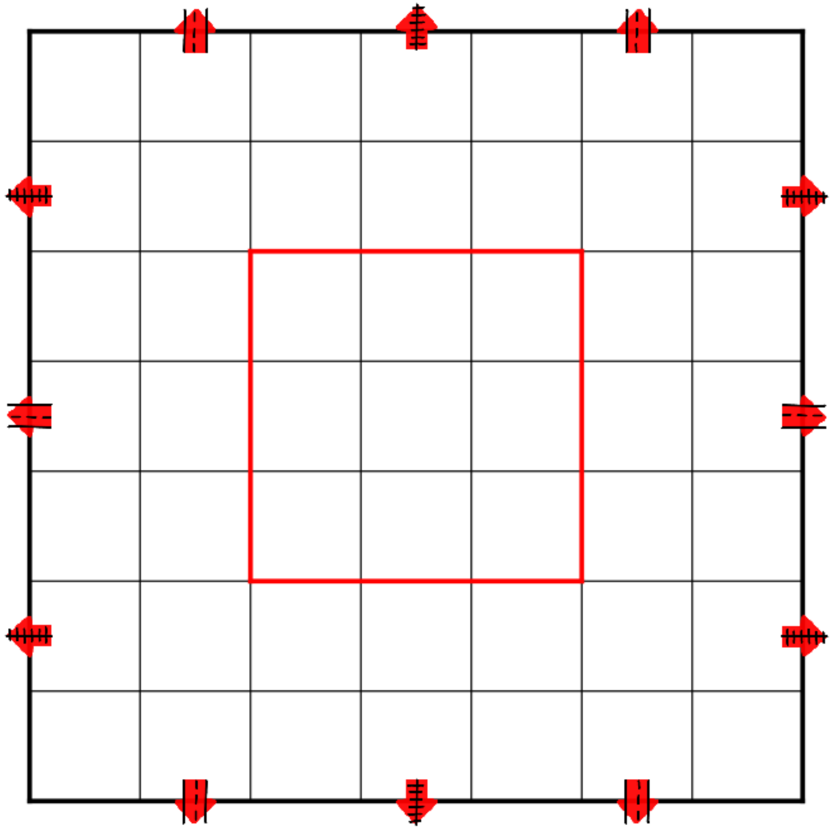
= "H0R0"

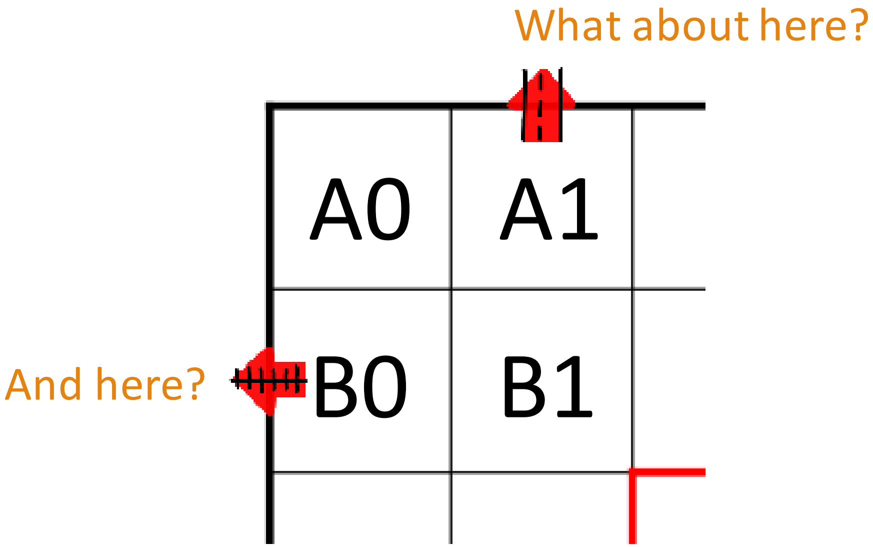


= "HRHR"

Board

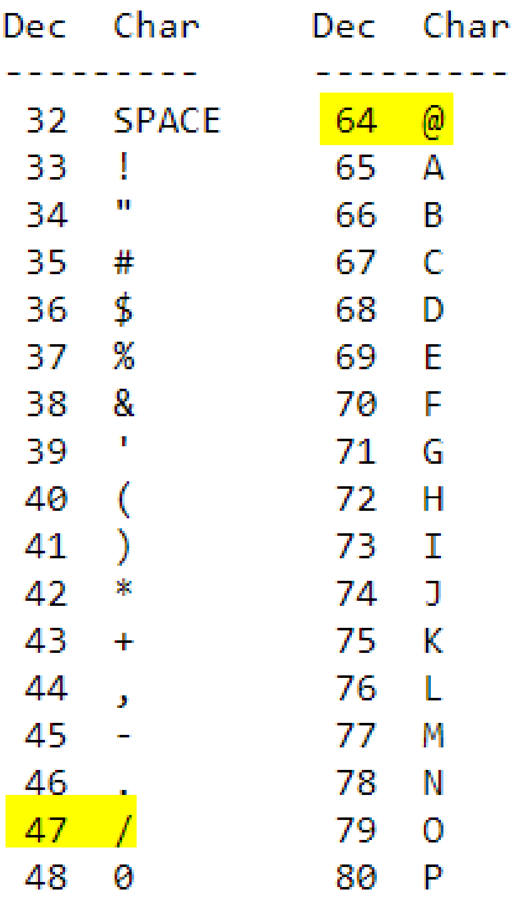
How to encode an exit?





Board

How to encode an exit?

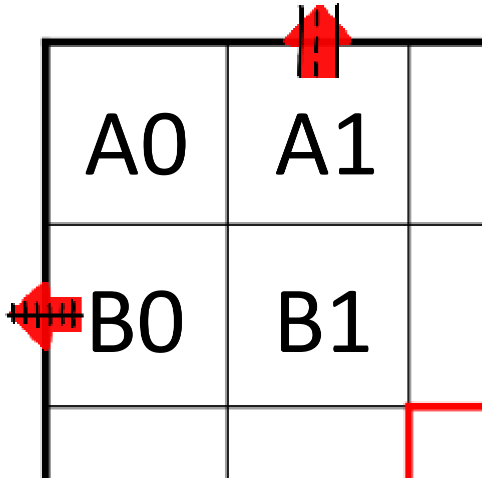


Dec	Char	Dec	Char
32	SPACE	64	@
33	!	65	A
34	"	66	B
35	#	67	C
36	$	68	D
37	%	69	E
38	&	70	F
39	'	71	G
40	(	72	H
41	)	73	I
42	*	74	J
43	+	75	K
44	,	76	L
45	-	77	M
46	.	78	N
47	/	79	O
48	0	80	P

Tile + Placement + Orientation

Highway exit;
Above row A

A4@10

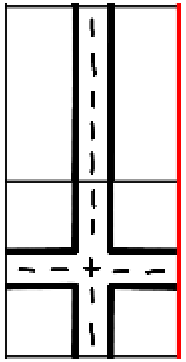


Railway exit;
On the left of column 0

A1B/1

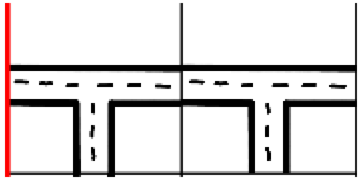
Tile Connection

Whether neighbouring tiles are connected



"H0H0"

"HHHH"

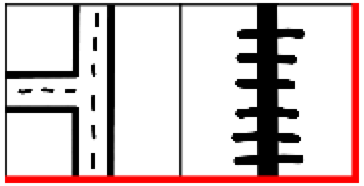


"0HHH"

"0HHH"

Same column: charAt 0 and 3 must be the same
Same row: charAt 2 and 4 must be the same

Not connected example:

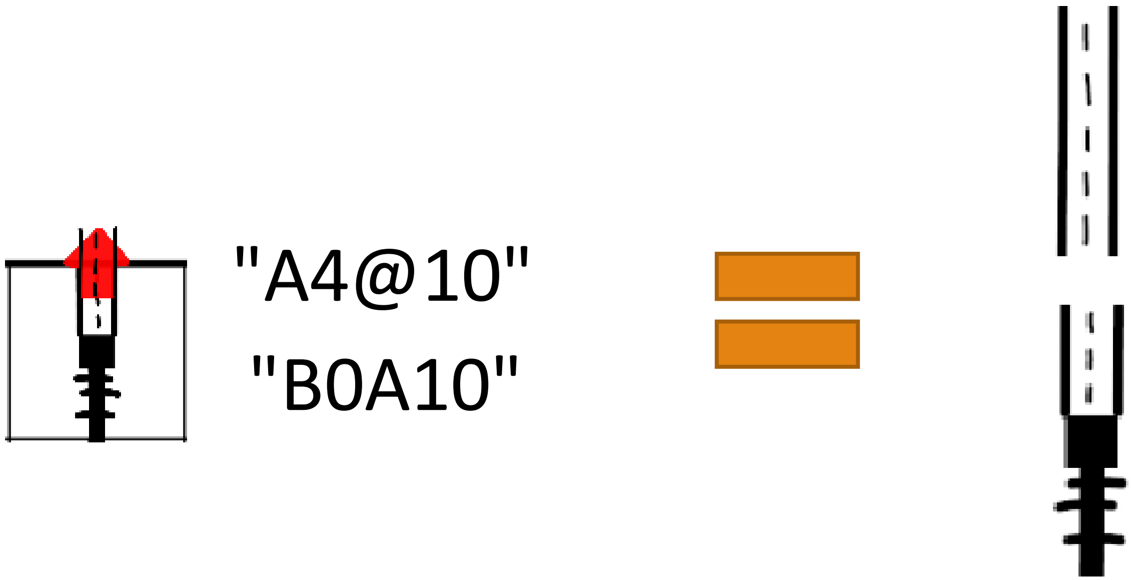


~~"HHH0"~~

~~"R0R0"~~

Tile Connection

Connected to exits?



"A4@10"

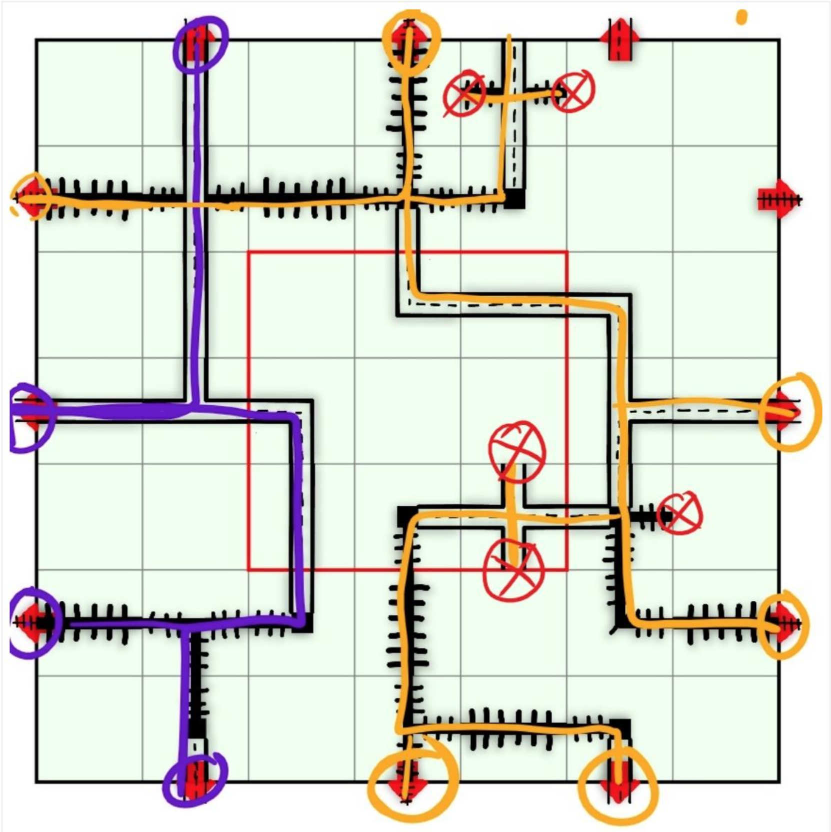
"B0A10"

=

"H0H0"

"H0R0"

Get Basic Score



- Central Area + : check board String how many in center

- Connected Exit +

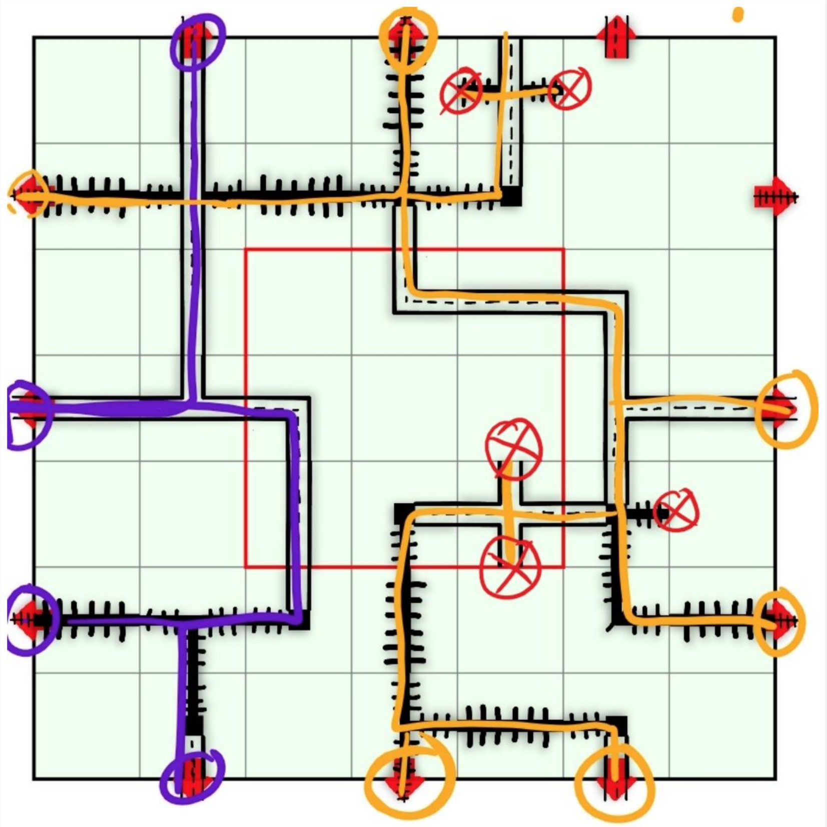
- Connectable but bare edge -

: for tile, first check ID we get edges of it, then check the rest tile get whether all edge are connected

Finding the Network

What is network?

Example of Network



We define : network is constructed by one tile connected to the other and extended as long as possible.

Two network on the left : Yellow and Purple

Finding the number of network is crucial :
because the basic score for exit is the (number of all connected exit in the board – network) *4

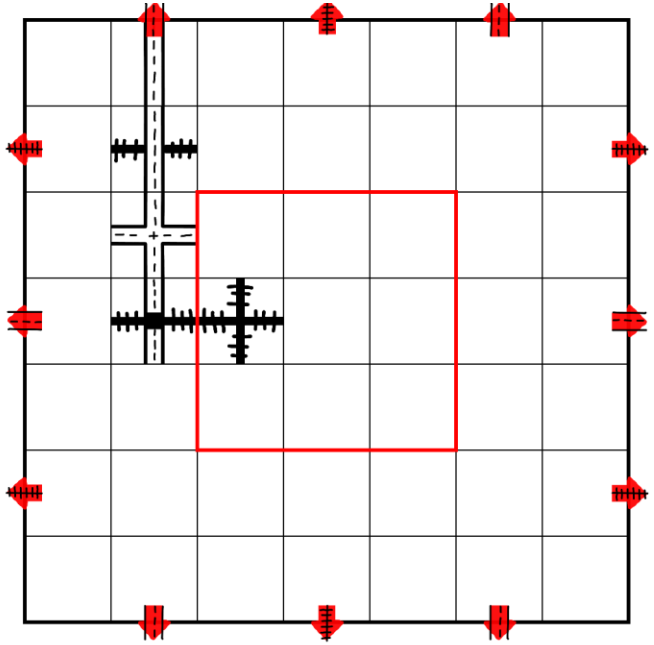
That is (9 –2) * 4 in this case.

Finding the Network

Adjacency matrix

Example of Network

Board String: **A4A16B2B14S2C13S5D12S3D25**



We have 5 tile : A4, B2, S2, S5, S3 in this case

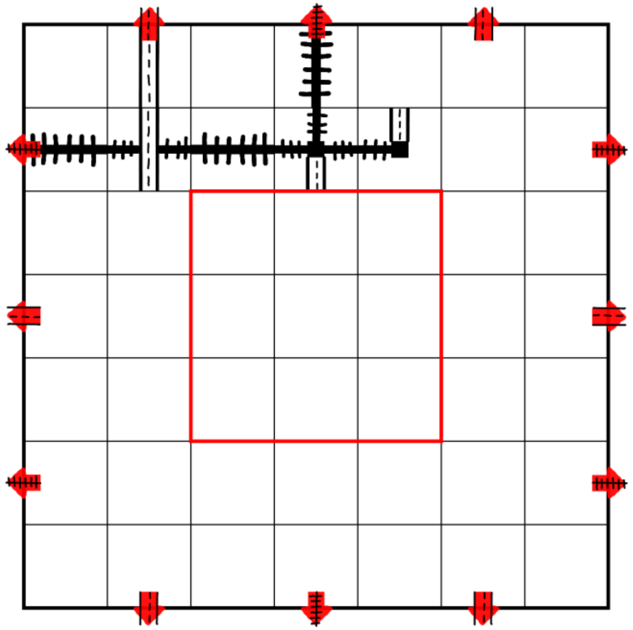
	A4A16	B2B14	S2C13	S5D12	S3D25
A4A16	0	1	0	0	0
B2B14	1	0	1	0	0
S2C13	0	1	0	1	0
S5D12	0	0	1	0	1
S3D25	0	0	0	1	0

Finding the Network

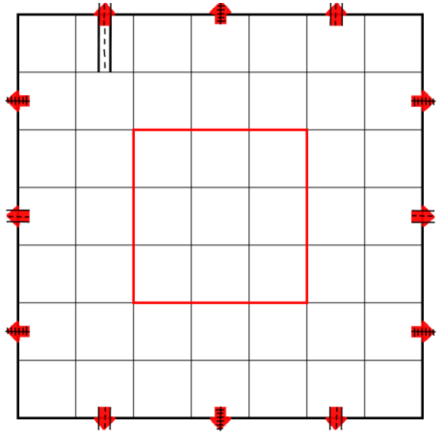
Depth first search algorithm

Example of Network

Board String: A4A12B2B16A1B01A1B23S1B32A1A32B1B44



Stage 1:



First, check adjacency matrix of tile, A4A12B, then I see it is connected with B2B16

Second, add the finding to stack

Third, the stack will pop the latest finding can we put the latest finding on the map

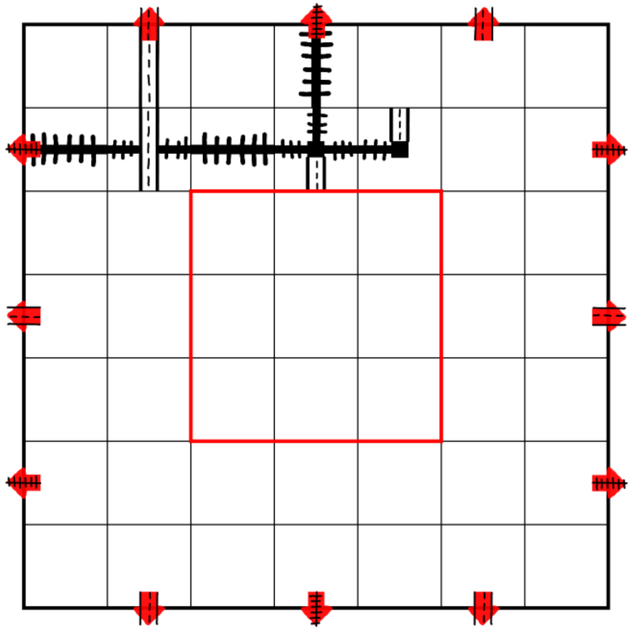
Why stack and what is latest?

Finding the Network

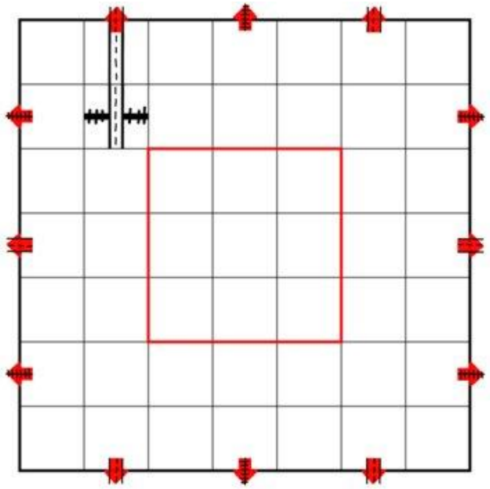
Depth first search algorithm

Example of Network

Board String: A4A12B2B16A1B01A1B23S1B32A1A32B1B44



Stage 2:



First, check adjacency matrix of latest finding tile, B2B16, then I see it is connected with A1B01 and A1B23

Second, add the finding s to stack

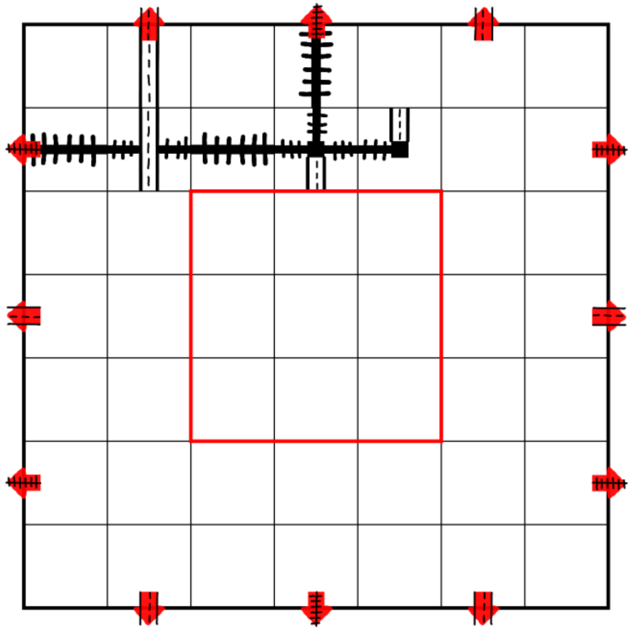
Third, the stack will pop the latest finding can we put the latest finding on the map , which is A1B23

Finding the Network

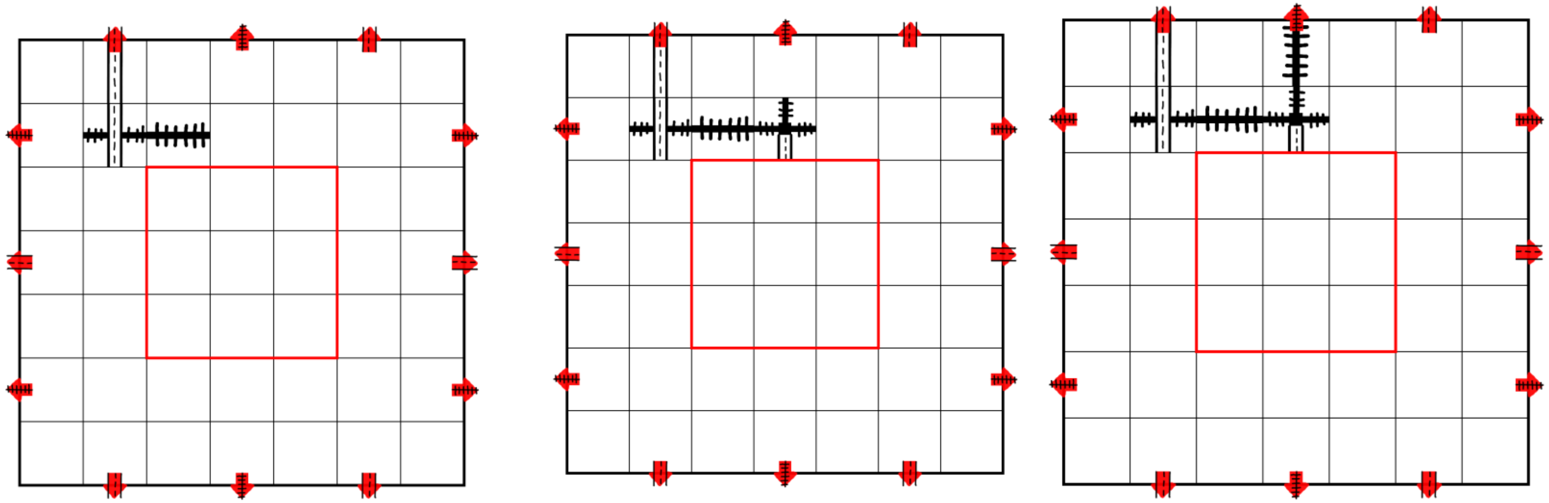
Depth first search algorithm

Example of Network

Board String: A4A12B2B16A1B01A1B23S1B32A1A32B1B44



Following Stages

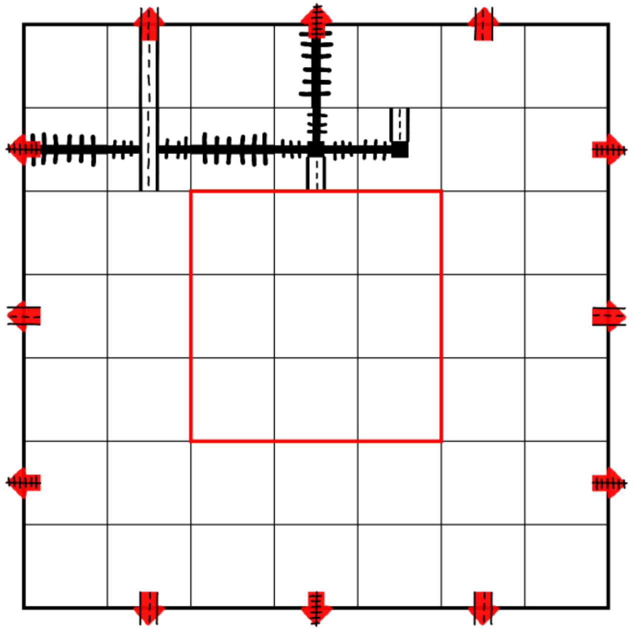


Finding the Network

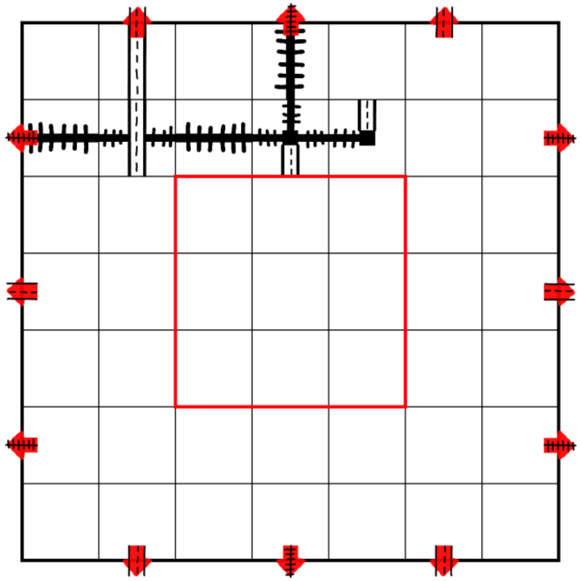
Depth first search algorithm

Example of Network

Board String: A4A12B2B16A1B01A1B23S1B32A1A32B1B44



Last Stages



There is only one tile left in stack ! So it finally pops A1B01! The one we find in stage two.

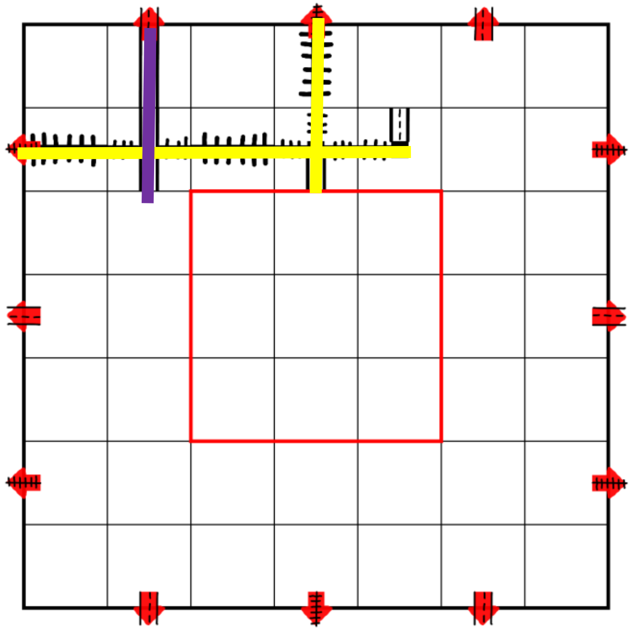
Rule of stack: first in last out.

One tricky tile: B2

Replacement

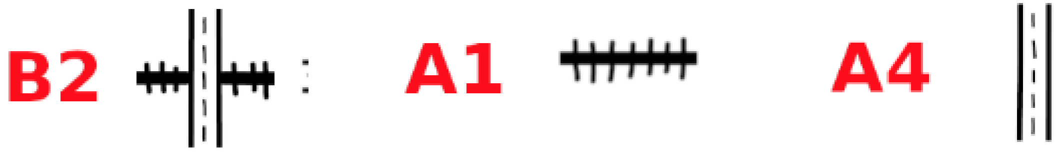
Example of Network

Board String: A4A12B2B16A1B01A1B23S1B32A1A32B1B44



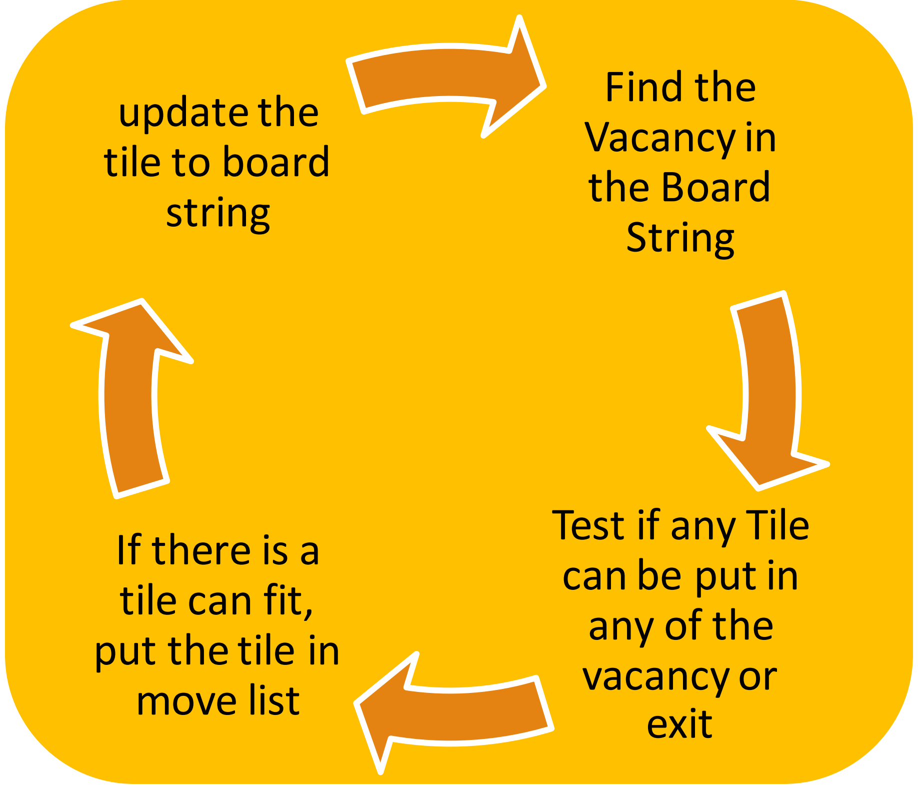
There is actually two network in this case!

Solution : Replace B2 with A1 and A4.



Generate Move

for int i = 0 ; i < worstcase; i++



Find the Vacancy in the Board String

Test if any Tile can be put in any of the vacancy or exit

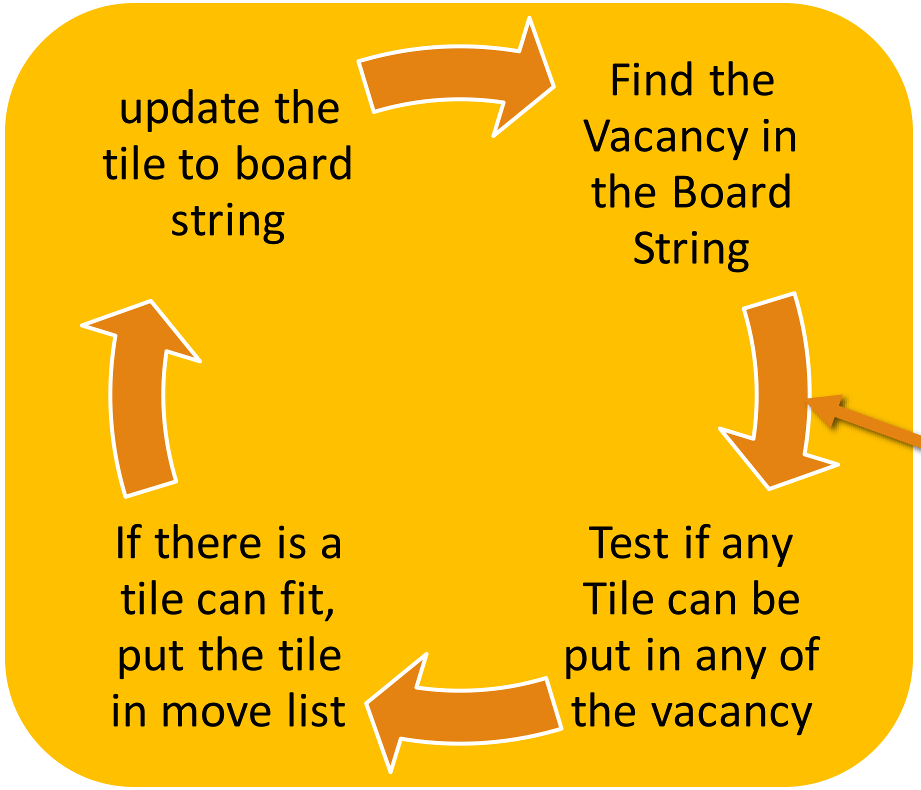
If there is a tile can fit, put the tile in move list

update the tile to board string

Worst case : 40

First loop : check 10 tile – 4 from dice roll , 6 from special
Second loop: check 9 tile
Third loop : check 8 tile
Fourth loop : check 7 tile
Fifth loop : check 6 tile

Generate Move



Find the Vacancy in the Board String

Test if any Tile can be put in any of the vacancy

If there is a tile can fit, put the tile in move list

update the tile to board string

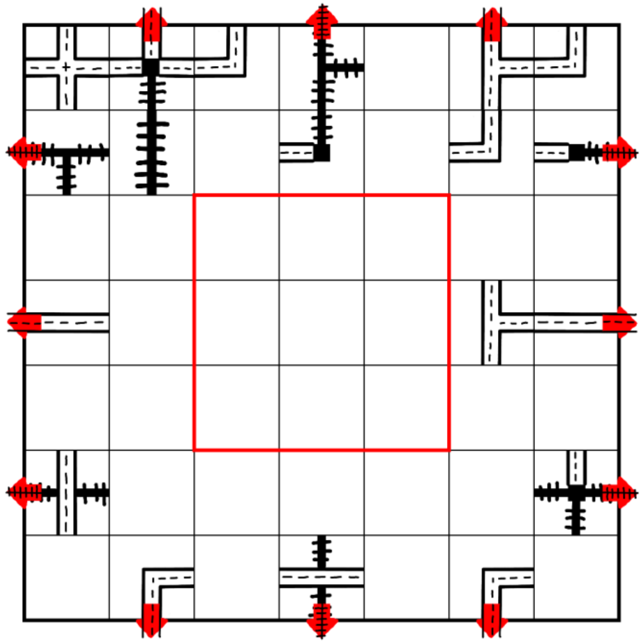
- Note : Special tile can only be used for three time
- Hence add a Boolean to check to check it have been used for three time
- If it did, the worst case will be 4 + 3 + 2 + 1 = 10

Whether should add Special Tile for the loop?

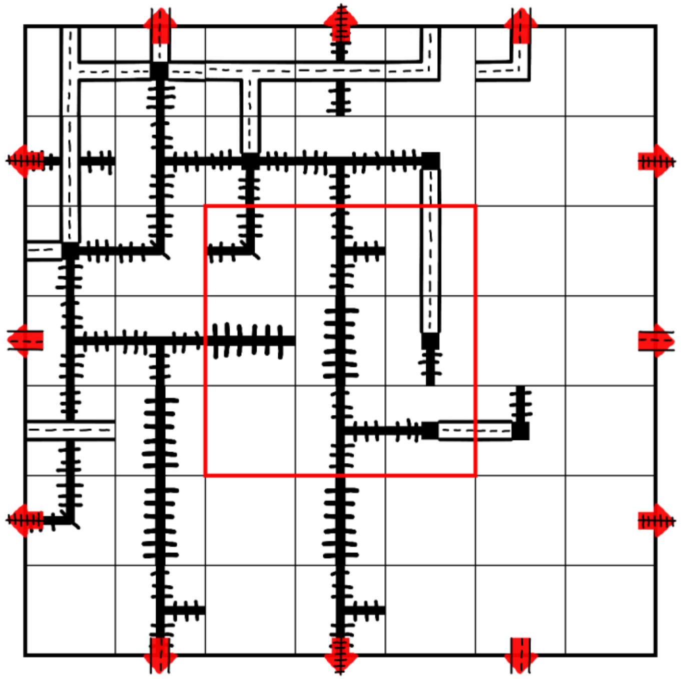
- Boolean 1: if we have used special tile this round
- Boolean 2: if there is already 3 special tile in the board String

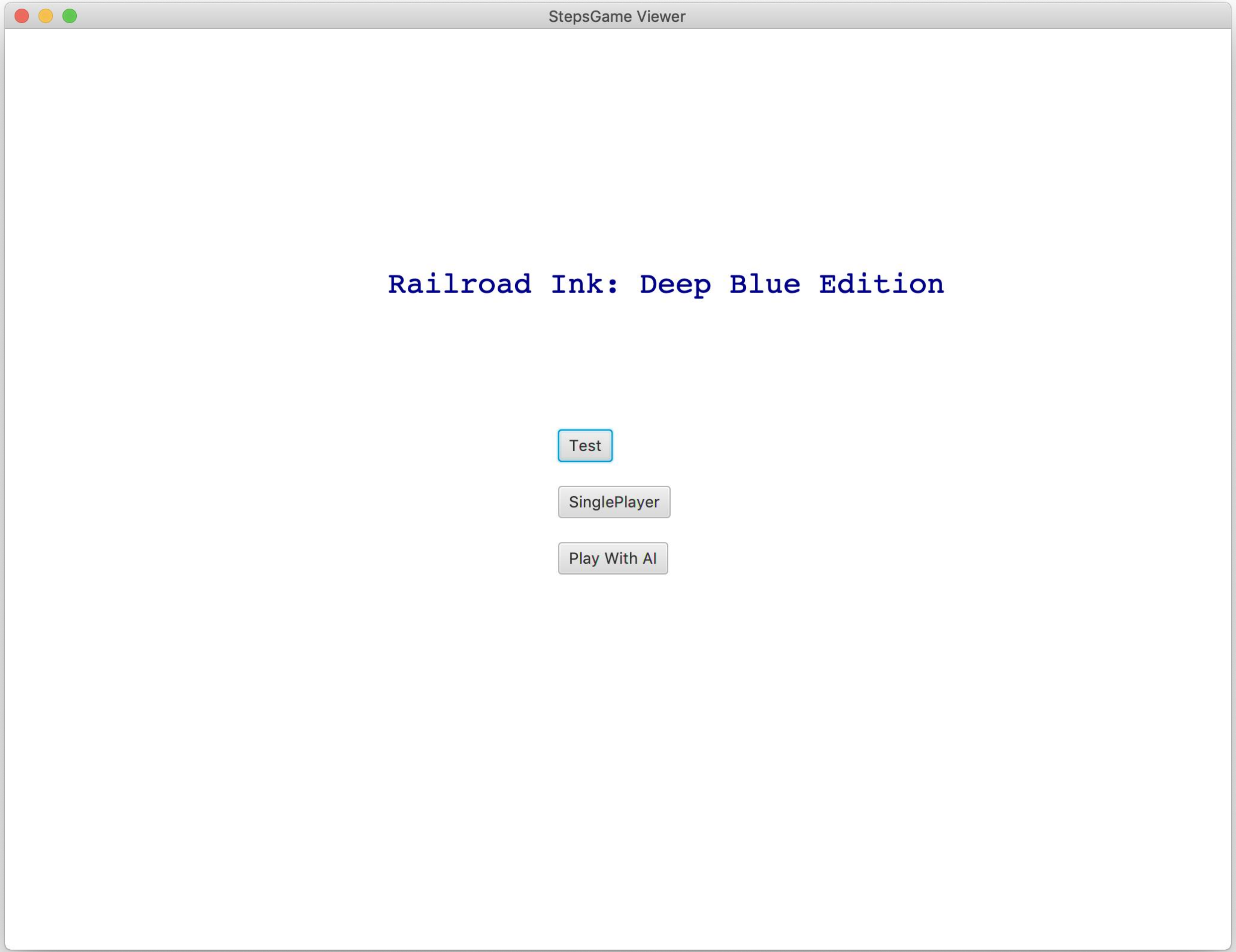
Tile First or Exit First?

Exit First

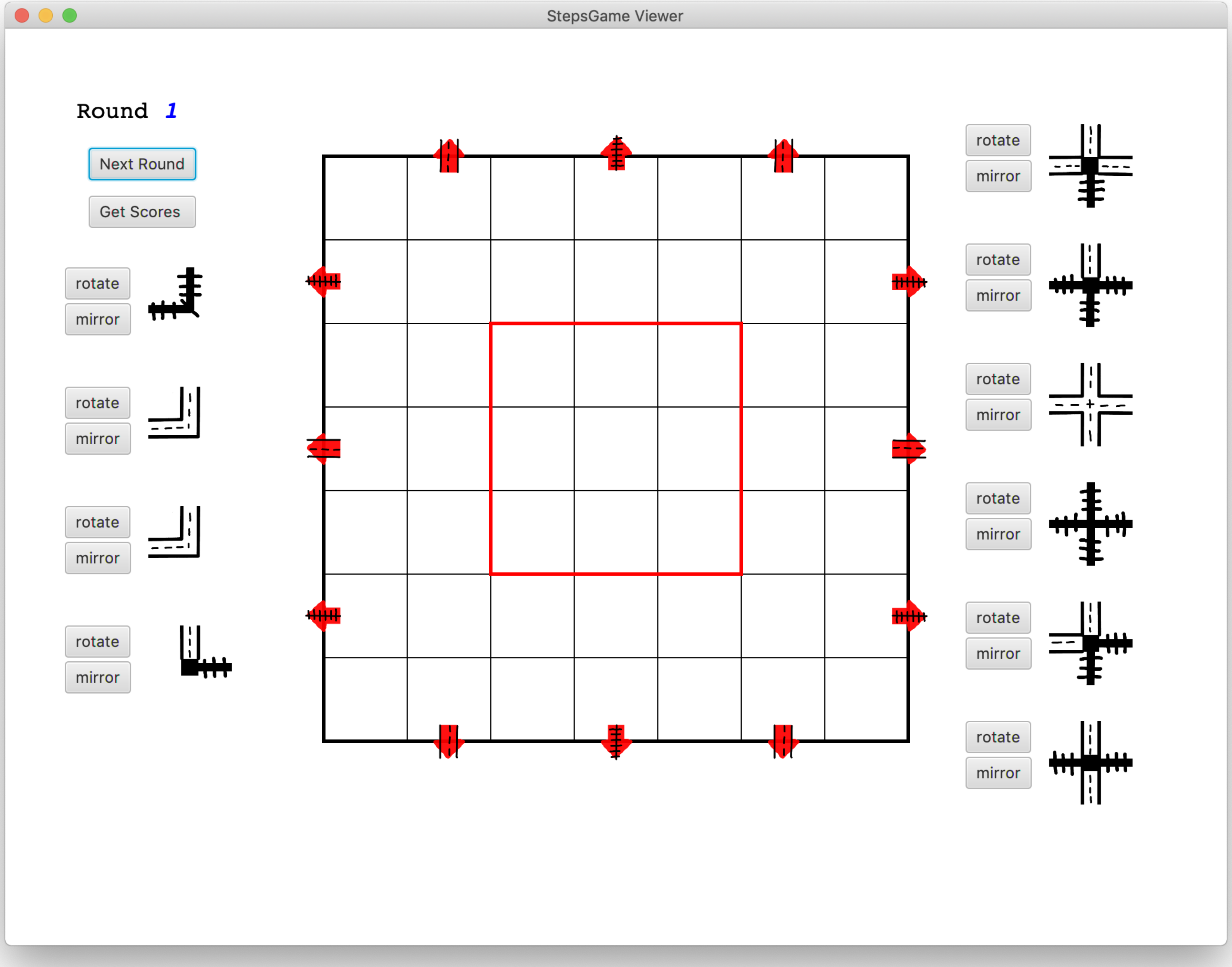


Tile First

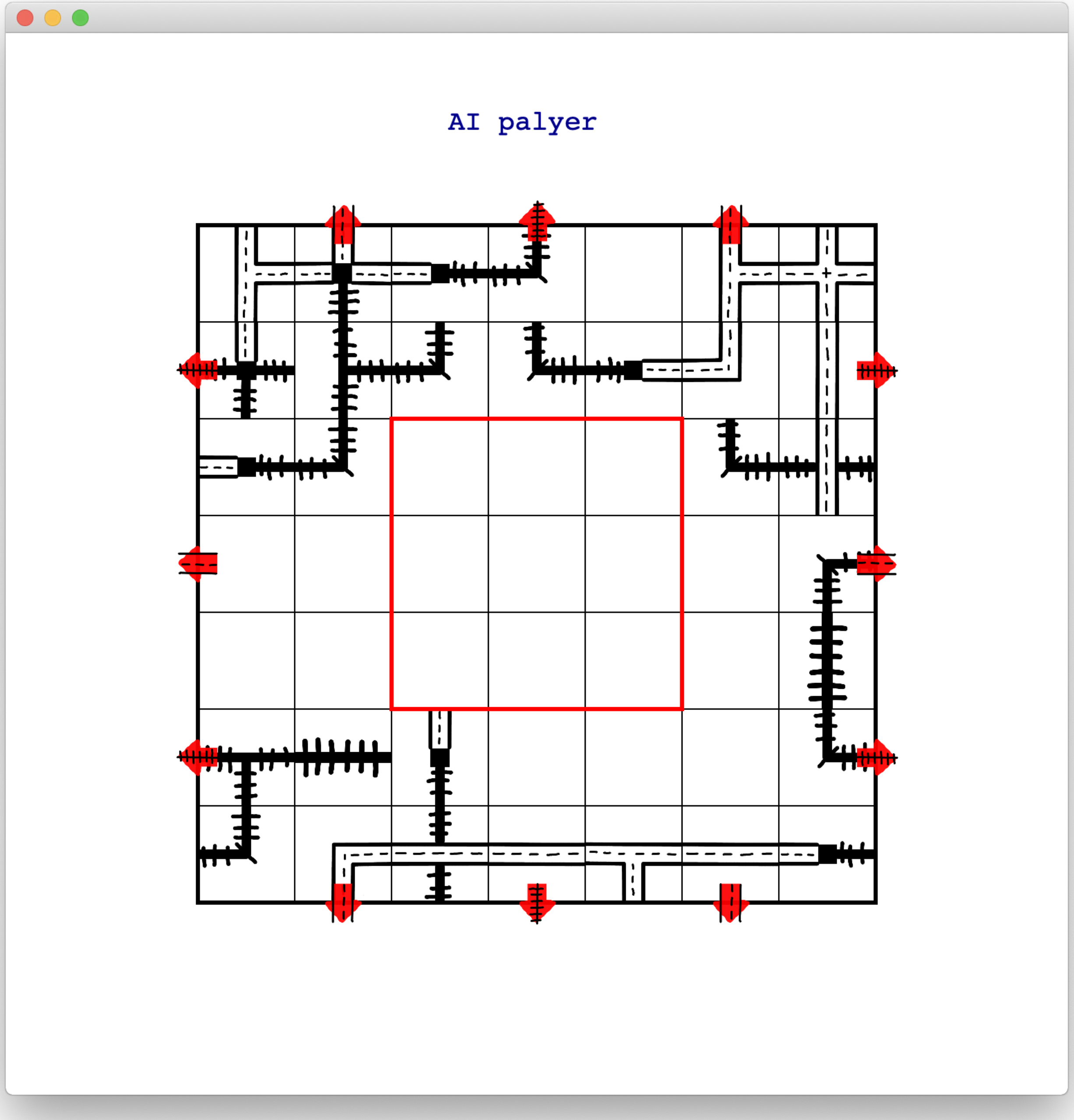




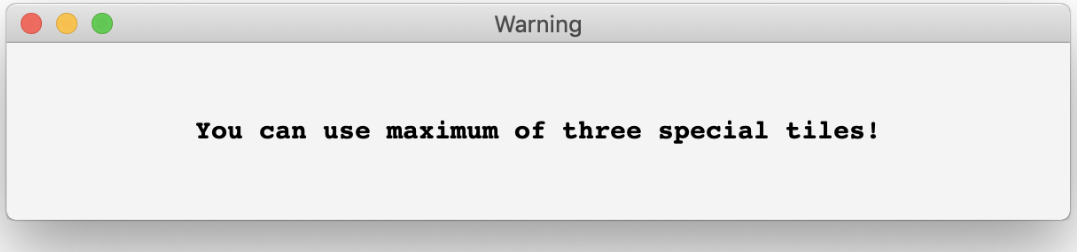
More than Simple



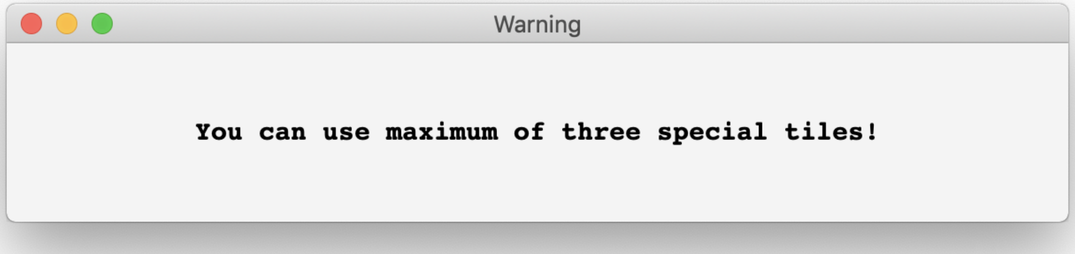
Left Mouse Click is enough



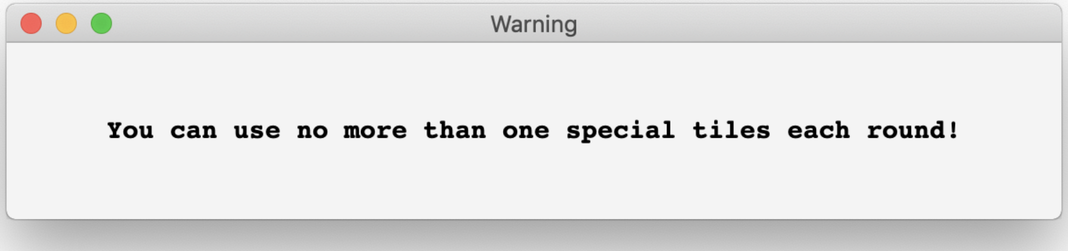
AI palyer



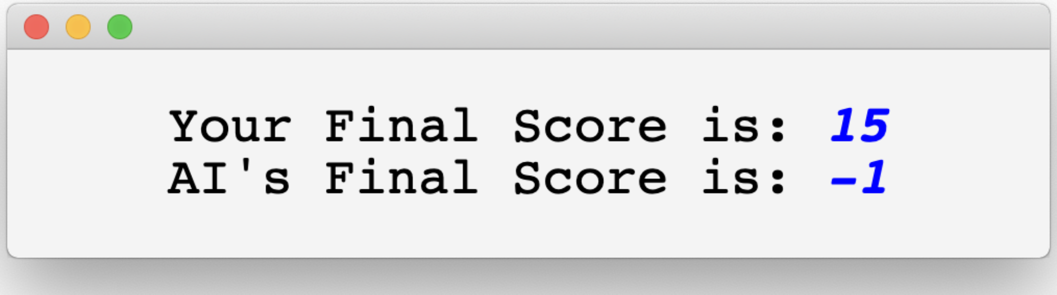
You can use maximum of three special tiles!



You can use maximum of three special tiles!

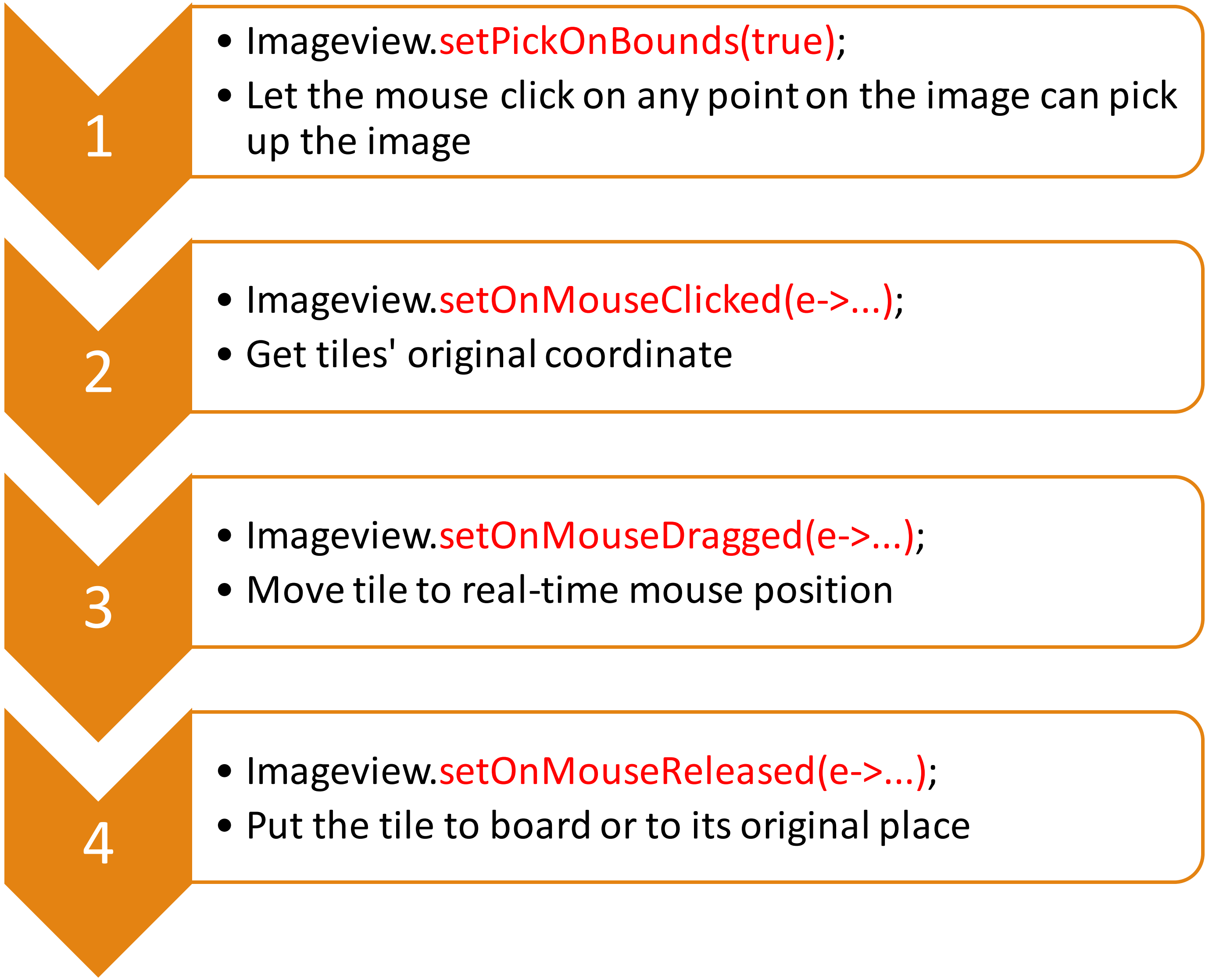


You can use no more than one special tiles each round!



Your Final Score is: *15*
AI's Final Score is: *-1*

Drag and Put



1

- Imageview.setPickOnBounds(true);
- Let the mouse click on any point on the image can pick up the image

2

- Imageview.setOnMouseClicked(e->...);
- Get tiles' original coordinate

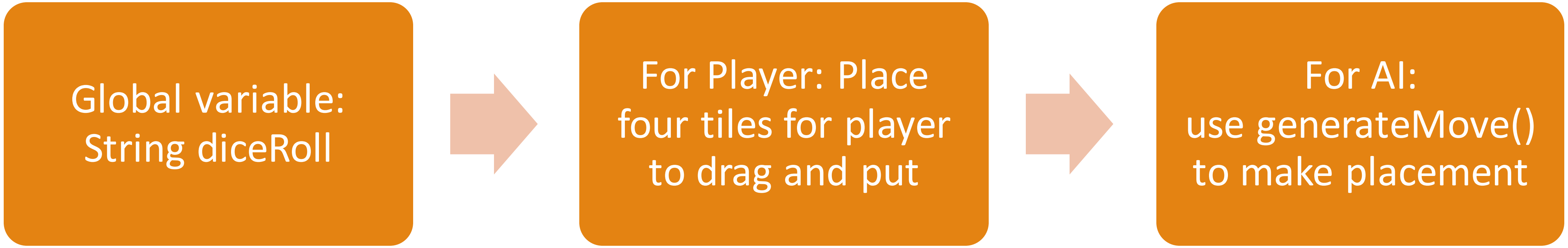
3

- Imageview.setOnMouseDragged(e->...);
- Move tile to real-time mouse position

4

- Imageview.setOnMouseReleased(e->...);
- Put the tile to board or to its original place

Connection between Single Player and AI



Global variable:
String diceRoll

For Player: Place four tiles for player to drag and put

For AI:
use generateMove() to make placement