

Swift Playgrounds - Apple

Learn to Code 1

Introduction

Swift Playgrounds is an educational app for iPad that makes coding fun and interactive.

Swift Playgrounds is designed with accessibility in mind and supports VoiceOver, Zoom, and Larger Text, so that students who are blind or have low vision can learn to code. This book contains tactile layouts corresponding to 3D puzzles in the Swift Playgrounds app.

Swift Playground puzzle worlds are based on a 3D grid, which you'll navigate to collect gems and toggle switches while you learn to code in Swift. While most of these puzzles stay the same on each page, some puzzles in the app are dynamic and change - those are noted as "dynamic layout." You will find the overview key in the front of the book.

Explore the key to familiarize yourself with the symbols, textures, and braille abbreviations that will be used in the graphics. Then move on to explore each graphic.

The graphics are tactile, braille, high-contrast ink print, and large print text in order to be universally accessible.

Table of Contents

Introduction..... i

Table of Contents..... ii-iv

Learn to code 1 Key..... v

Commands

Issuing Commands..... 1

Adding a New Command..... 2

Toggle A Switch..... 3

Portal Practice.....4

Finding and Fixing Bugs..... 5

Bug Squash Practice..... 6

The Shortest Route.....7

Functions

Composing a New Behavior..... 8

Creating A New Function..... 9

Collect, Toggle, Repeat..... 10

Across the Board..... 11

Nesting Patterns..... 12

Slotted Stairways..... 13

Treasure Hunt.....14

For Loops

Using Loops..... 15

Looping All the Sides.....16

To the Edge and Back..... 17

Loop Jumper.....18

Branch Out..... 19

Gem Farm.....20

Four Stash Sweep.....21

Conditional Code

Checking for Switches..... 22

Using else if.....23

Looping Conditional Code.....24

Conditional Climb..... 25

Defining Smarter Functions.....26

Boxed In.....27

Decision Tree..... 28

Logical Operators

Using the NOT Operator..... 29

Spiral of NOT.....30

Checking This AND That..... 31

Checking This OR That..... 32

Logical Labyrinth..... 33

While Loops

Running Code While... 34

Creating Smarter While Loops..... 35

Choosing the Correct Tool..... 36

Four by Four.....37

Turned Around.....38

Land of Bounty..... 39

Nesting Loops..... 40

Random Rectangles..... 41

You’re Always Right..... 42

Algorithms

The Right-Hand Rule.....43

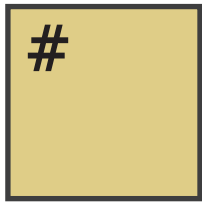
Adjusting Your Algorithm.....44

Conquering a Maze..... 45

Which Way to Turn? 46

Roll Right, Roll Left.....47

Swift Playgrounds Learn to Code 1 Key



Land block with height. Numeric indicator has been omitted.



Unreachable/water block



Player's starting point and direction



Gem



Obstacle (two obstacles together along a grid line can create a wall)



Stairs



Open switch



Closed switch



Portal with color noted in braille:



Blue



Green

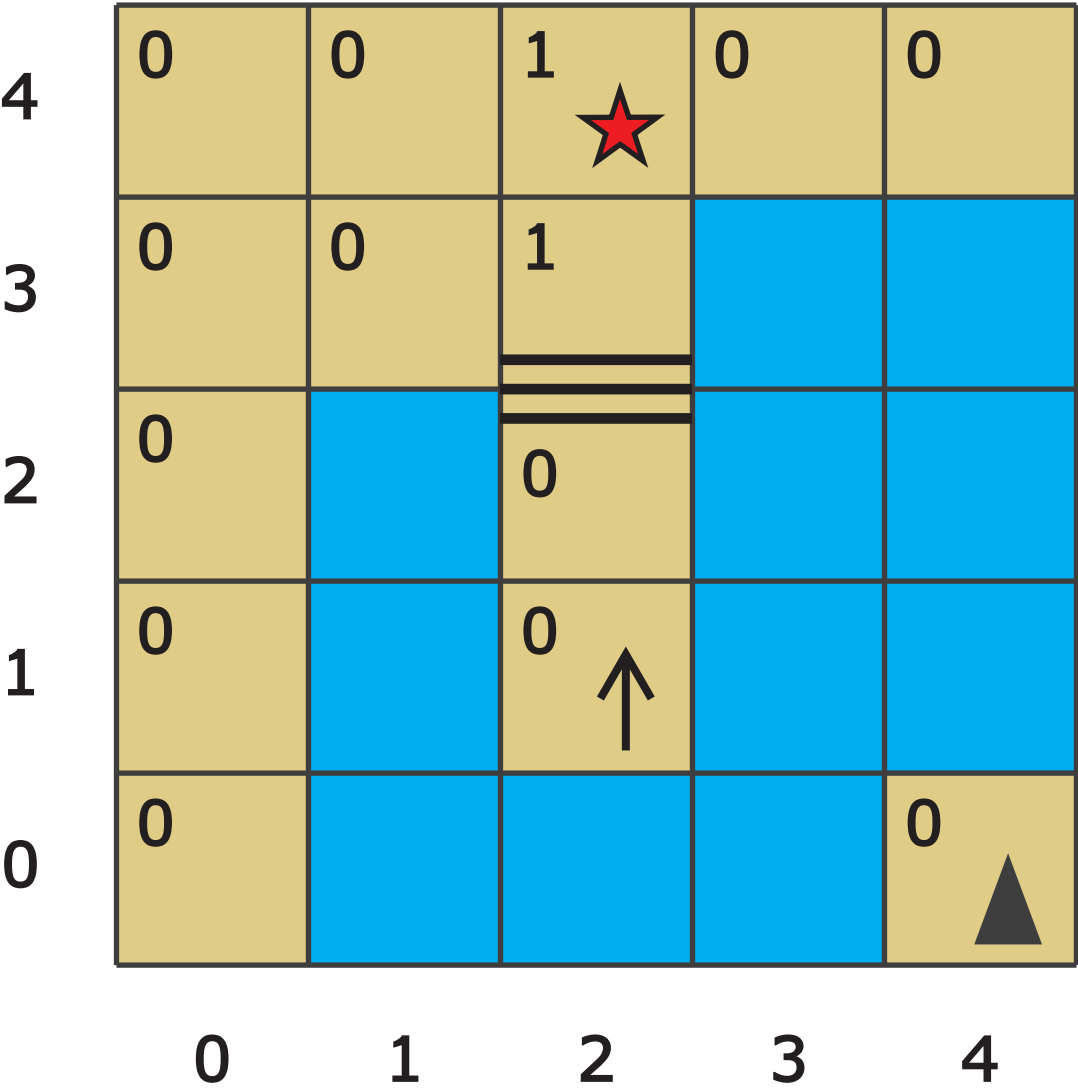


Pink

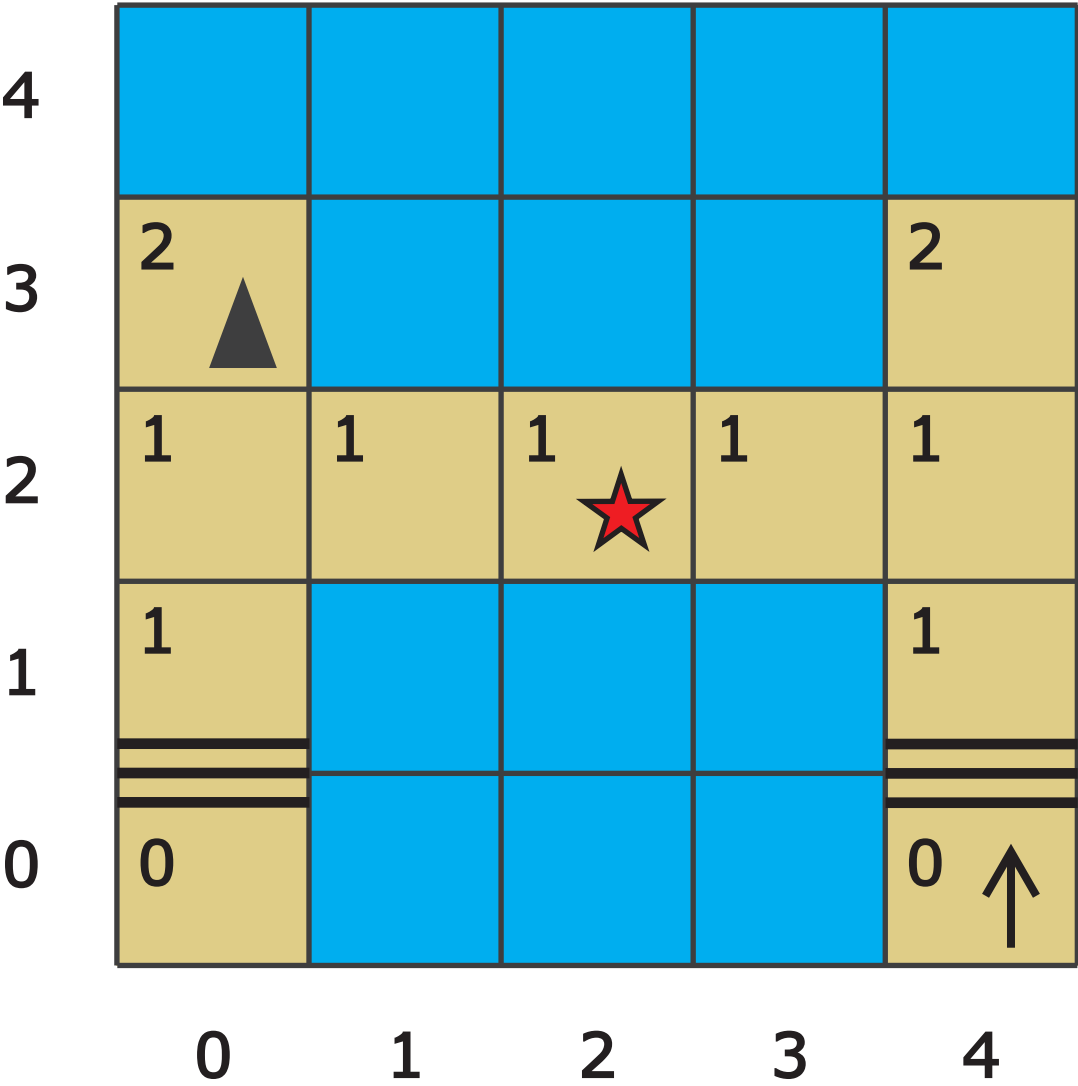


Yellow

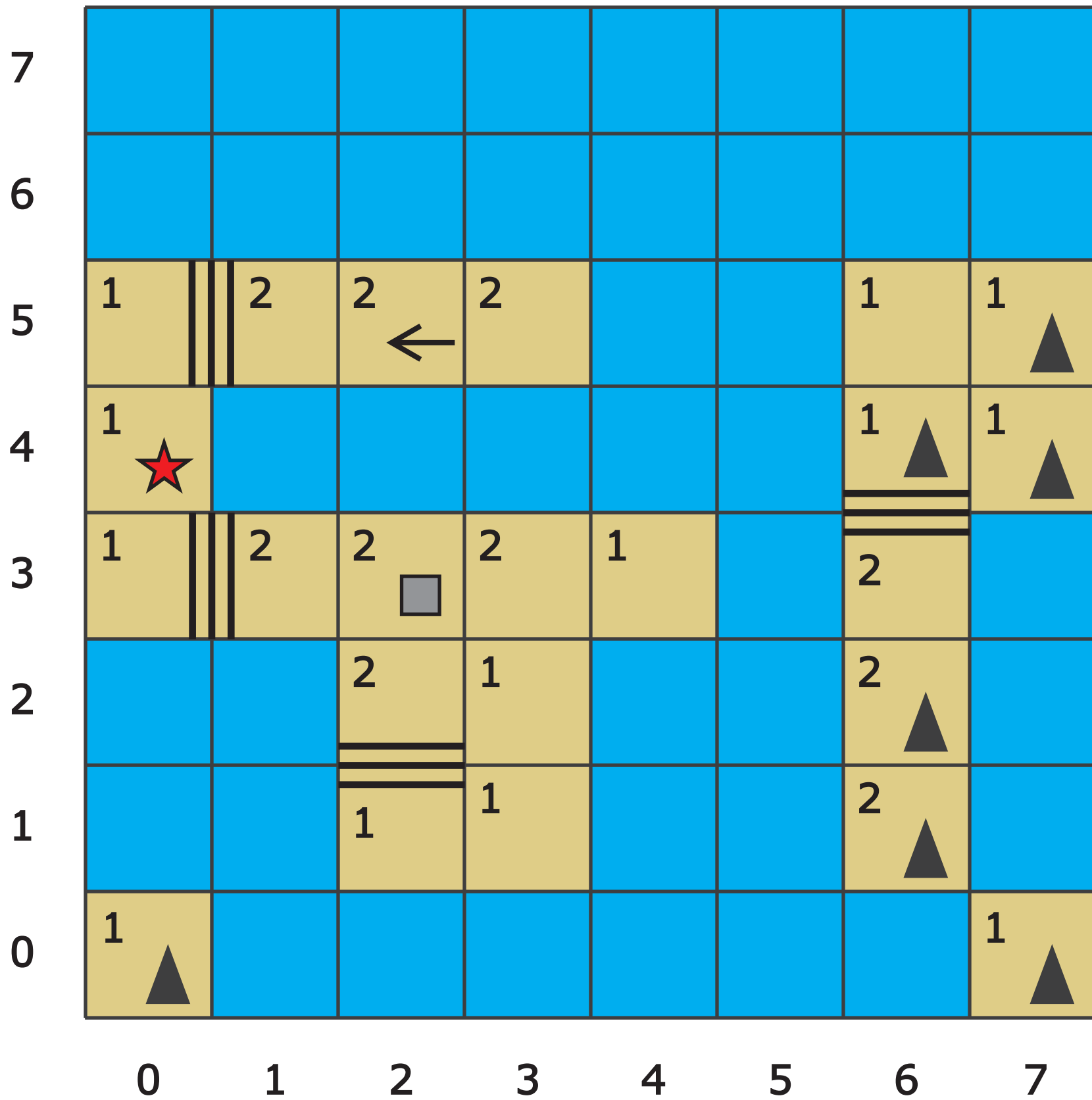
Issuing Commands



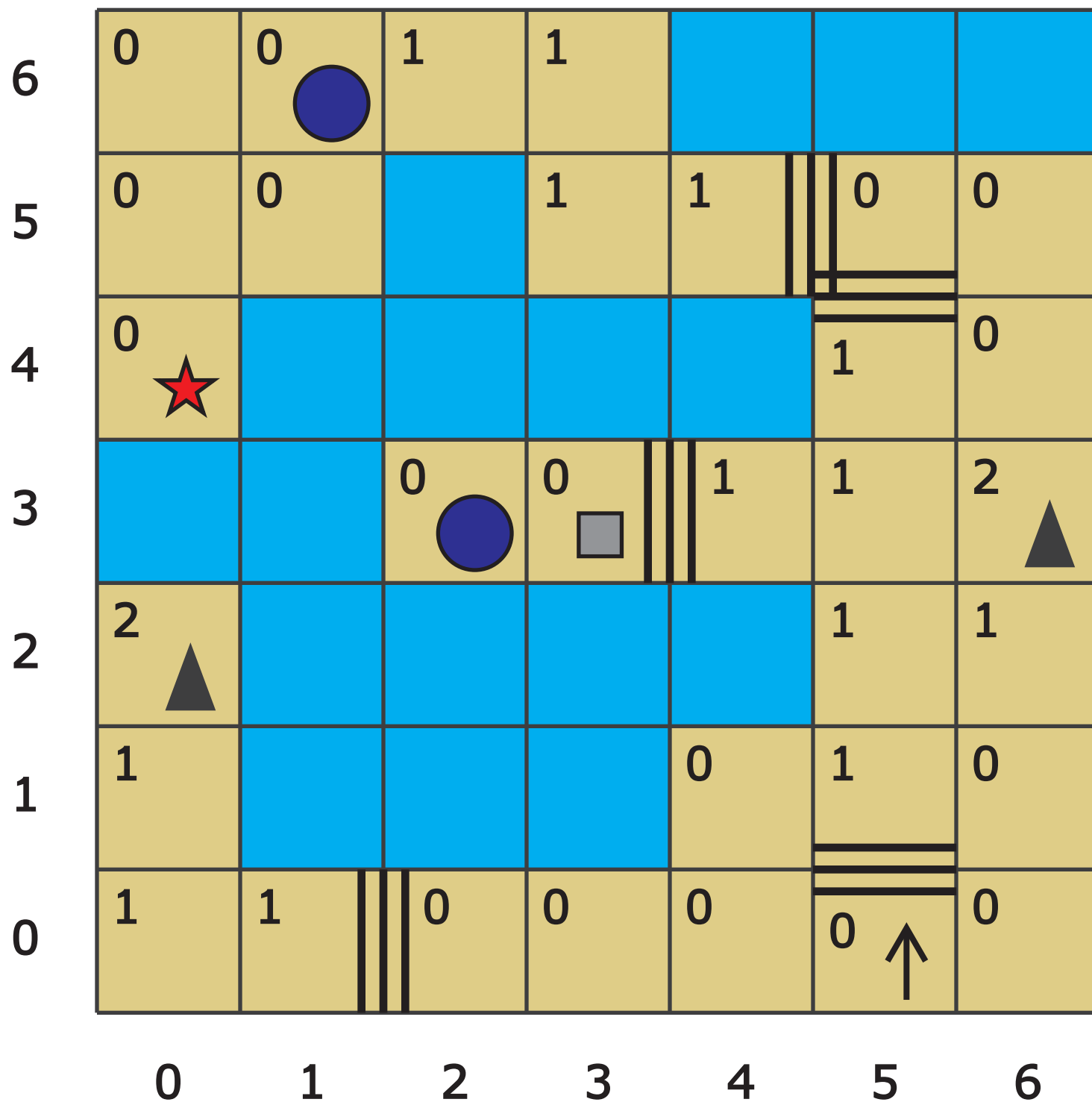
Adding a New Command



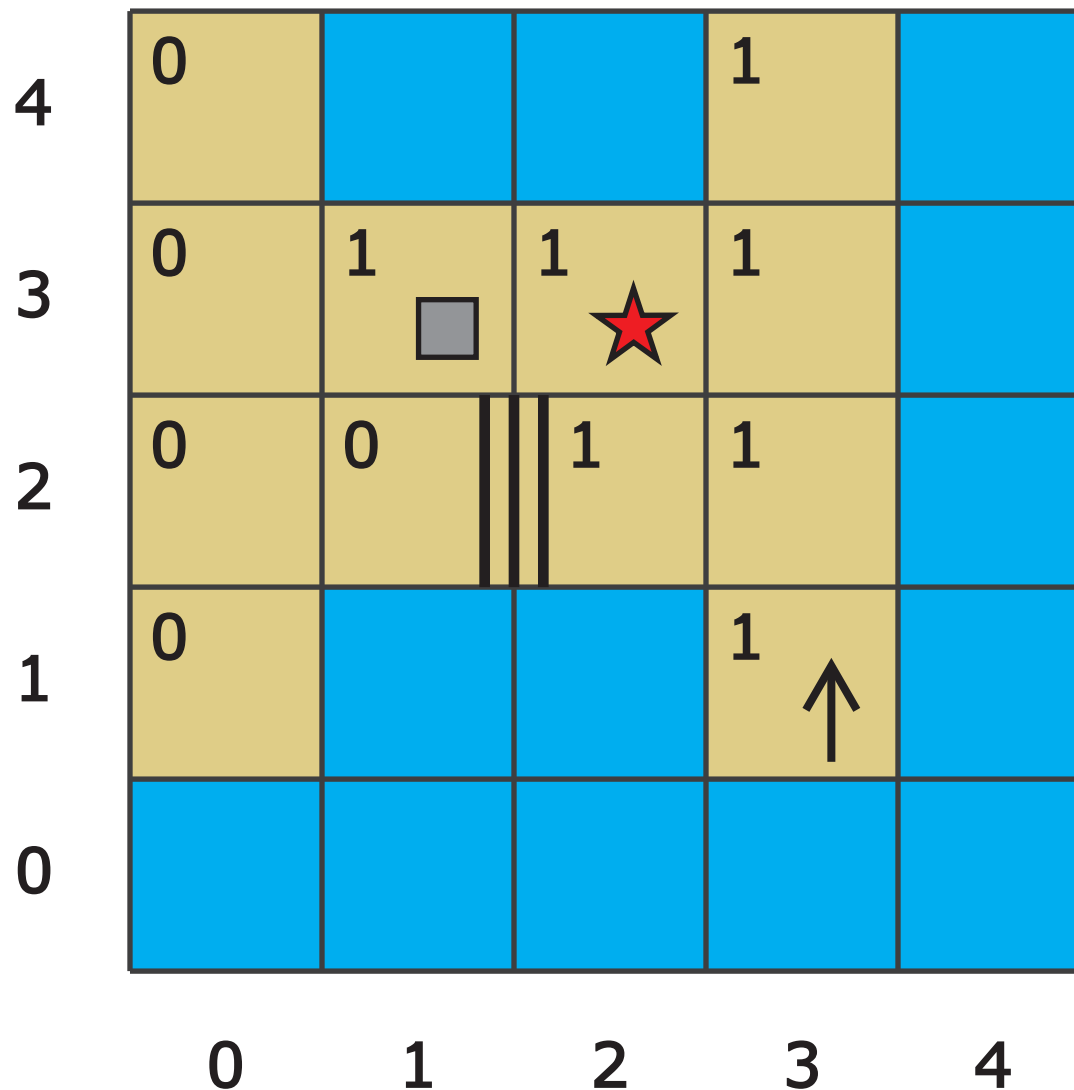
Toggling a Switch



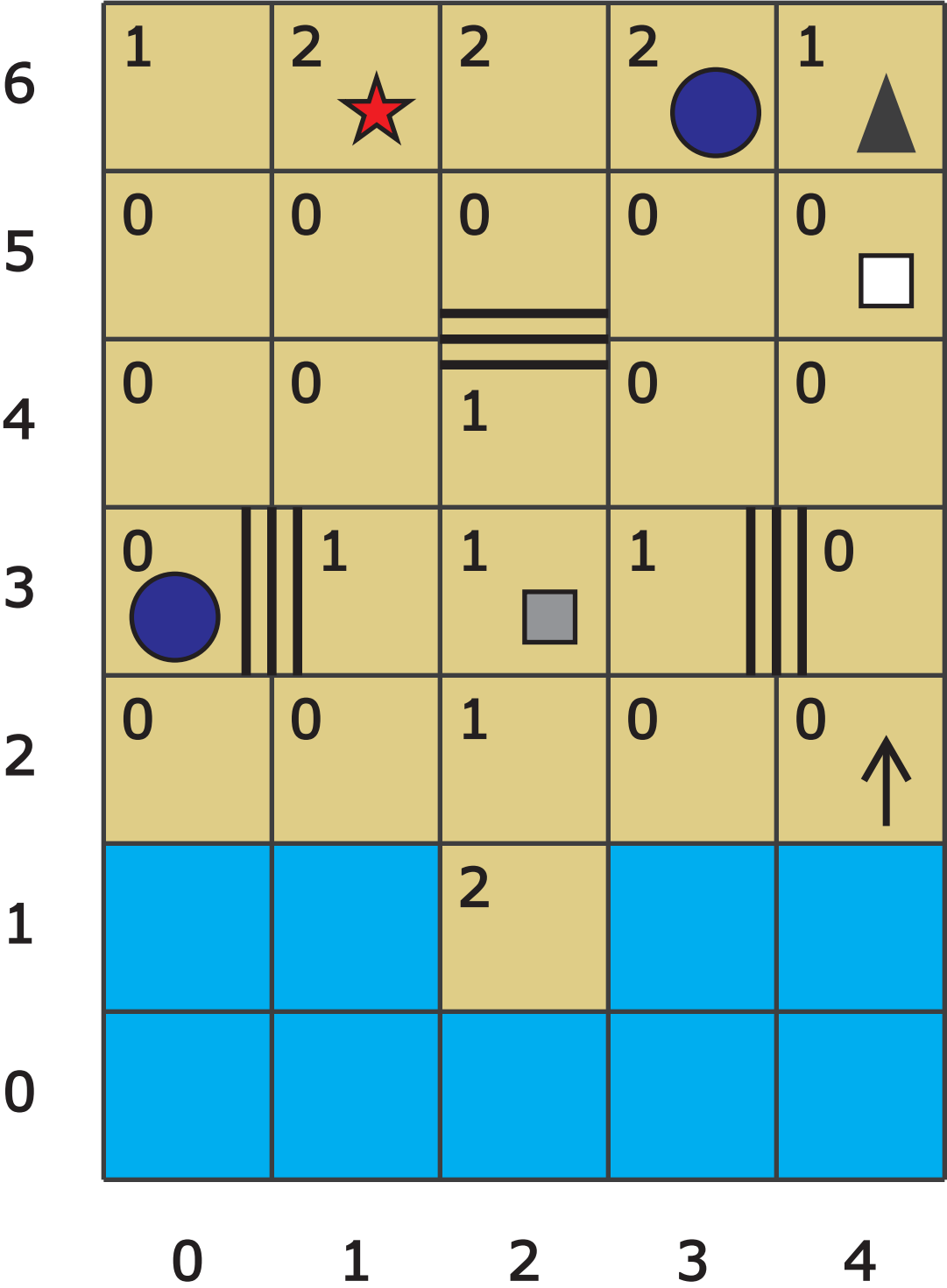
Portal Practice



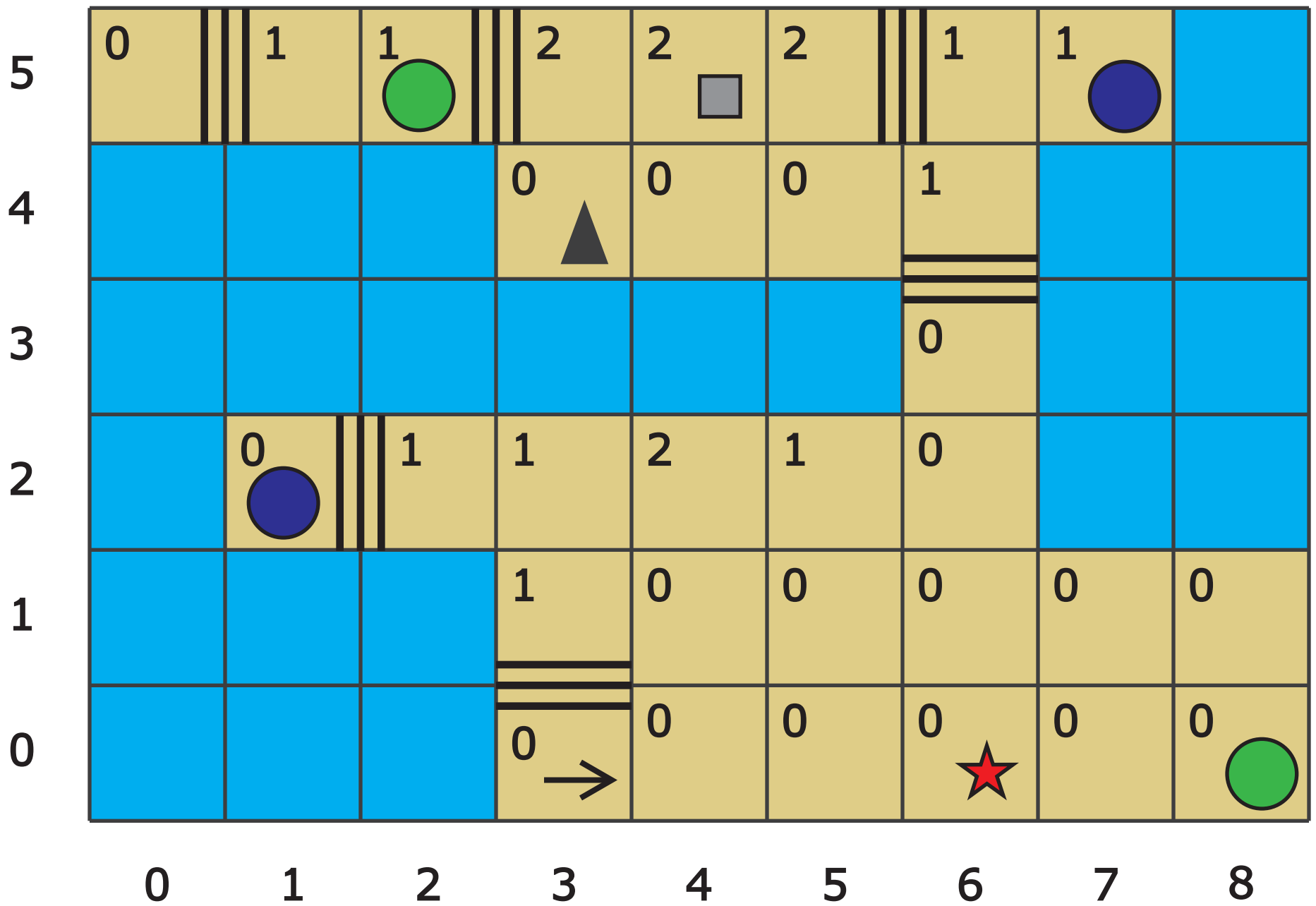
Finding and Fixing Bugs



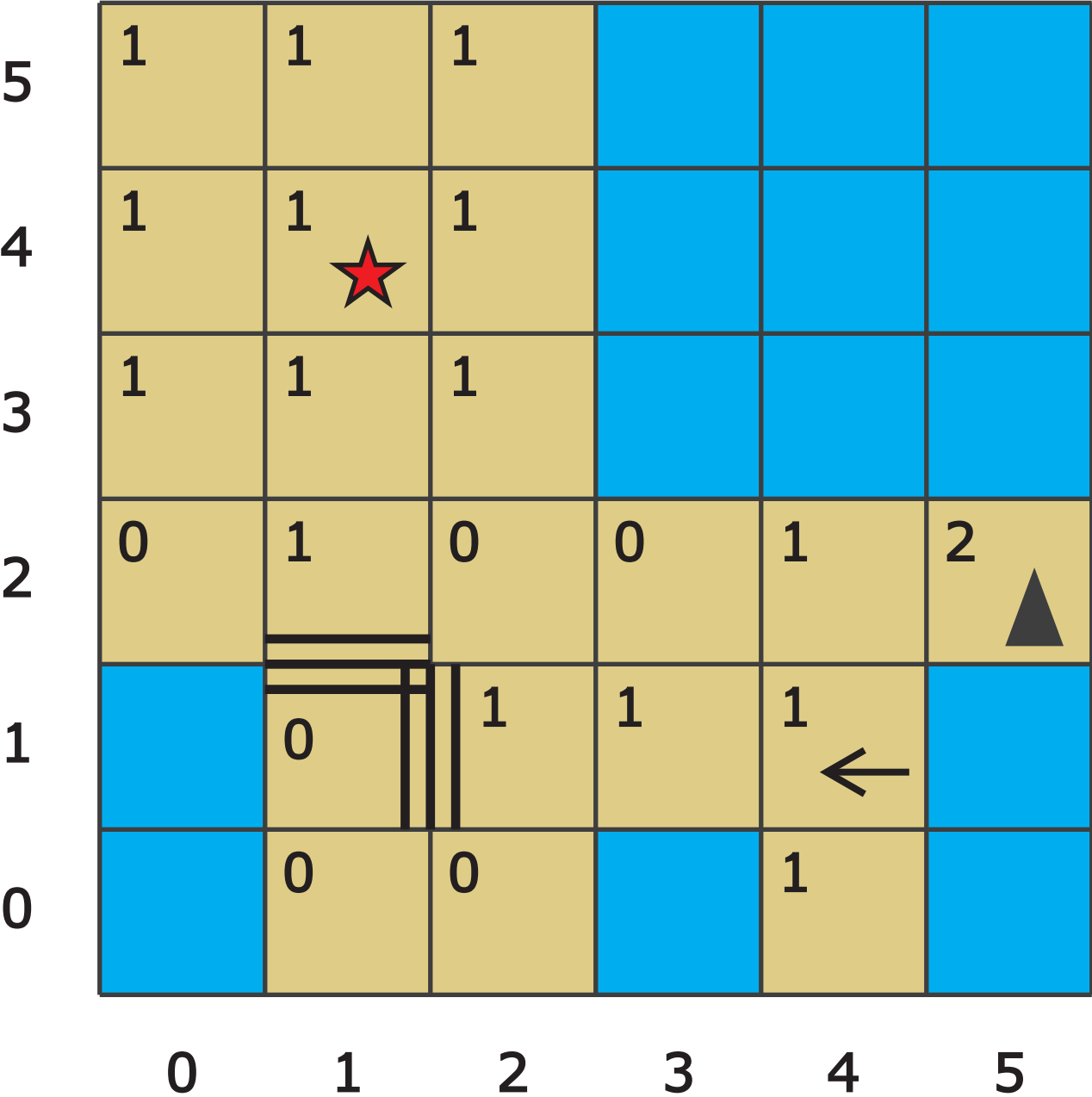
Bug Squash Practice



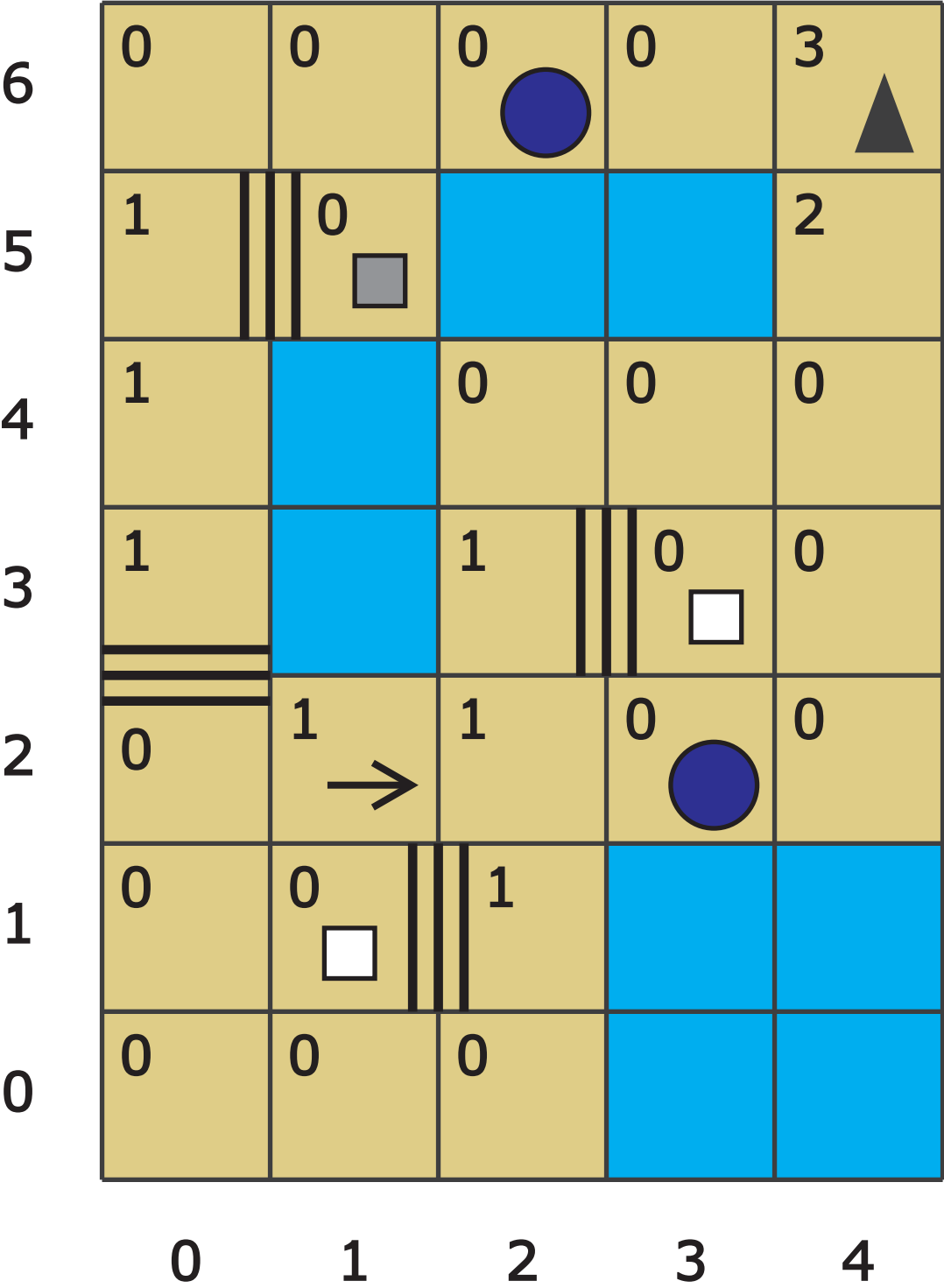
The Shortest Route



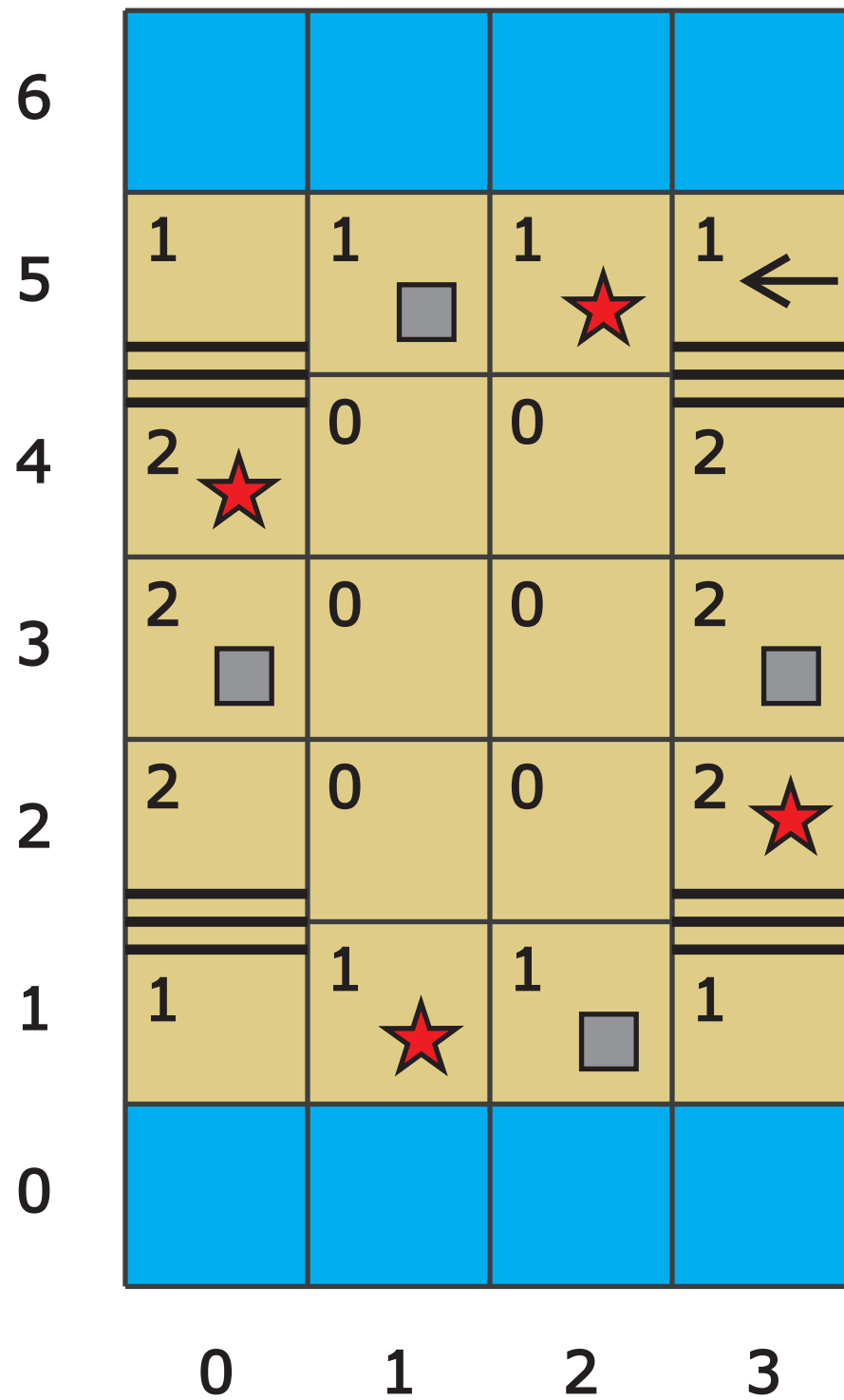
Composing a New Behavior



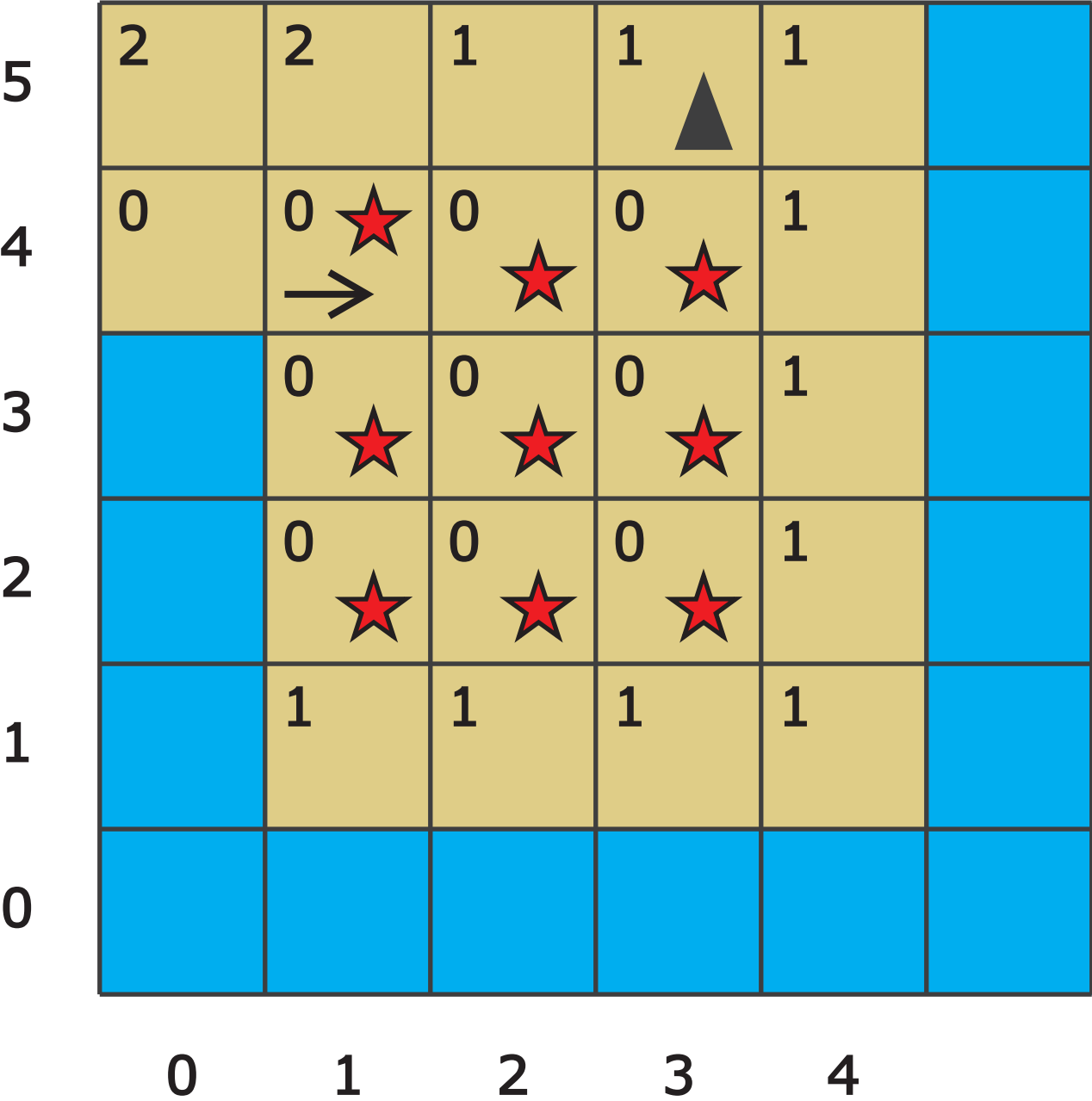
Creating a New Function



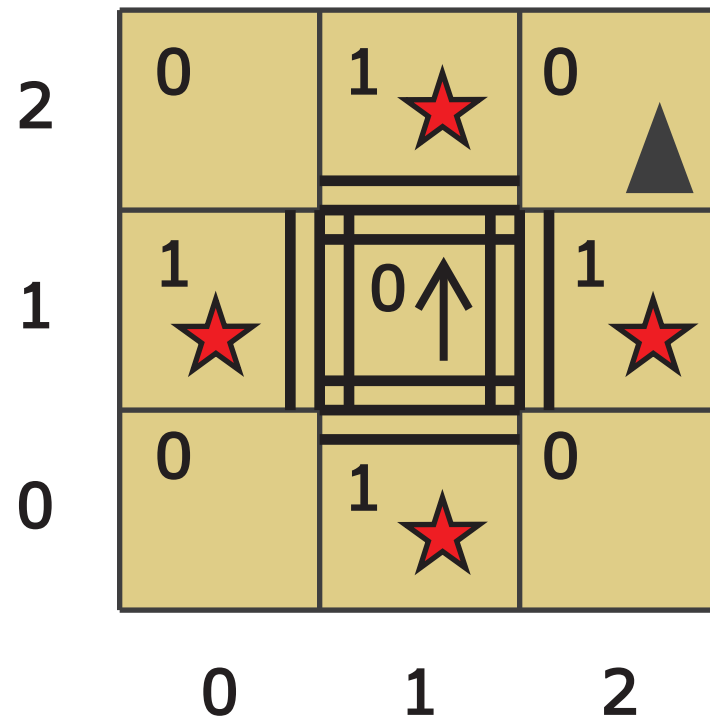
Collect, Toggle, Repeat



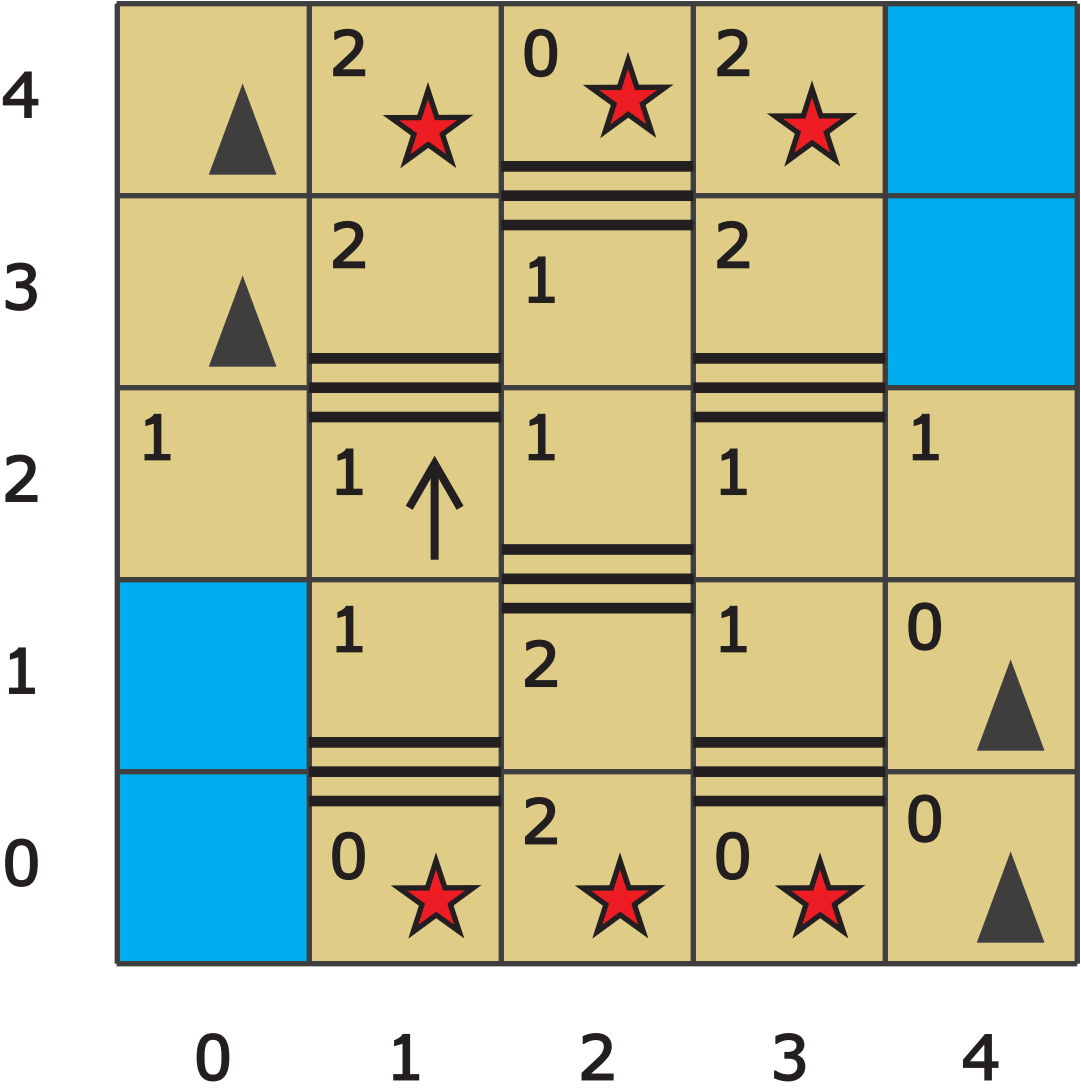
Across the Board



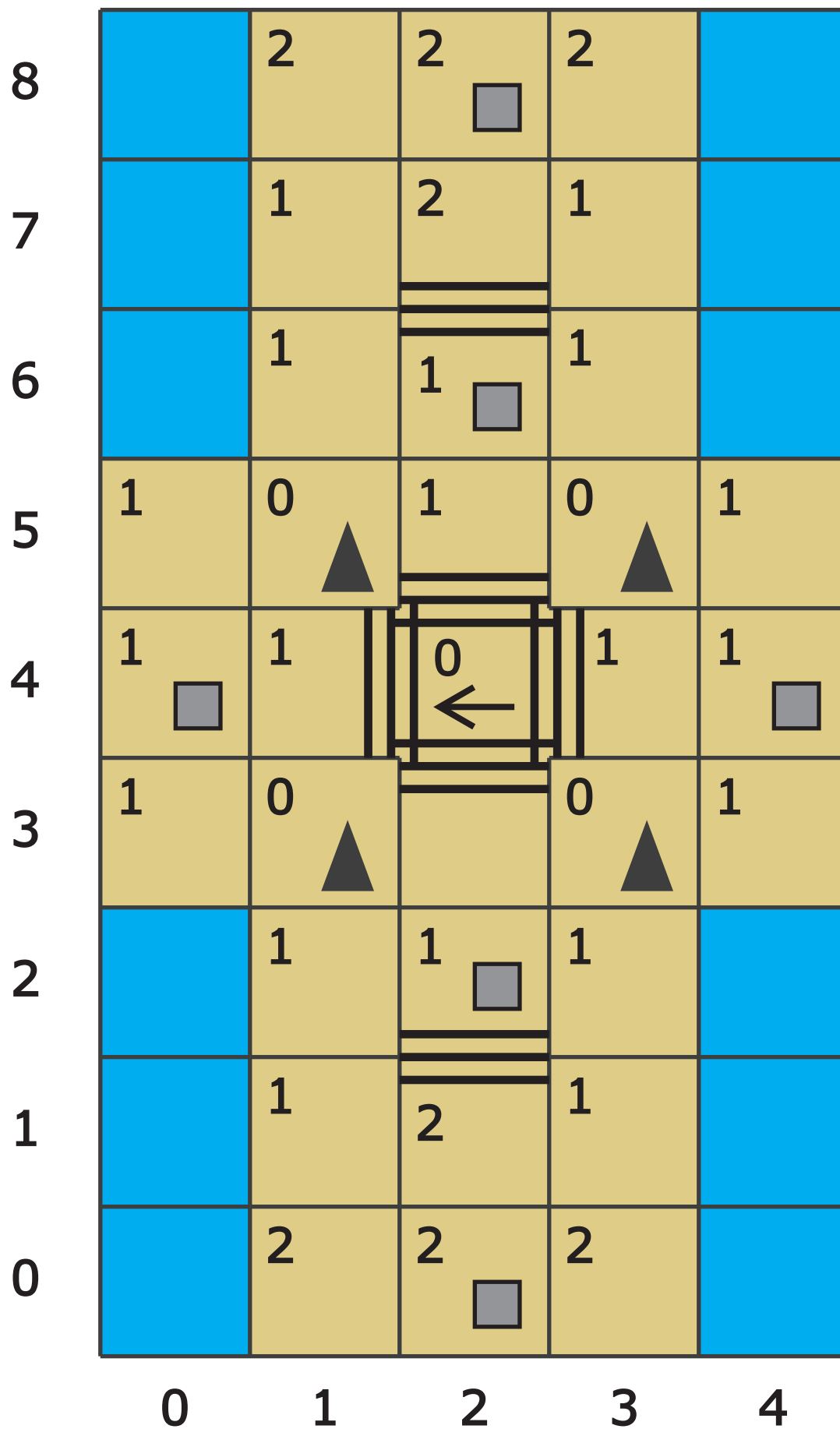
Nesting Patterns



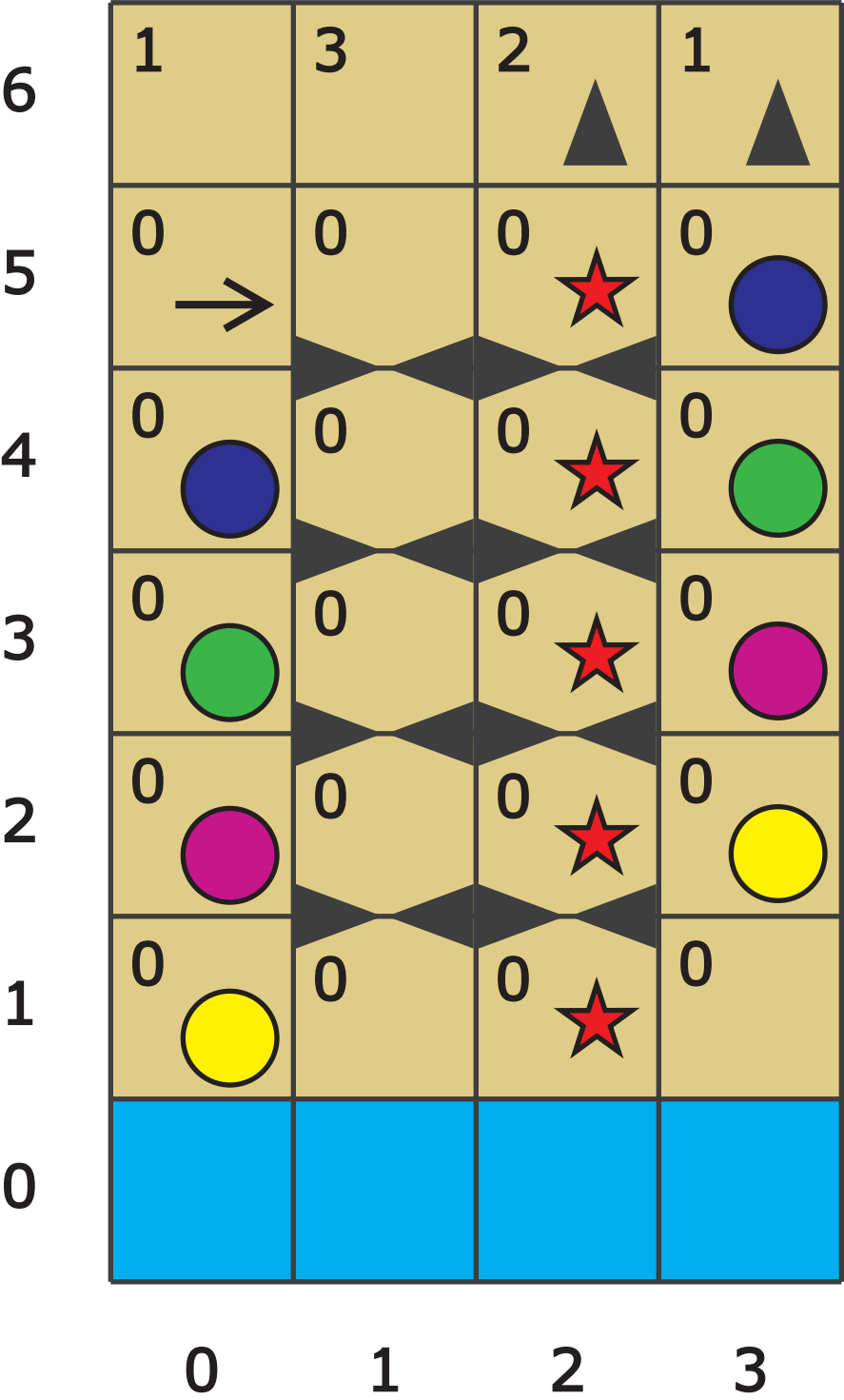
Slotted Stairways



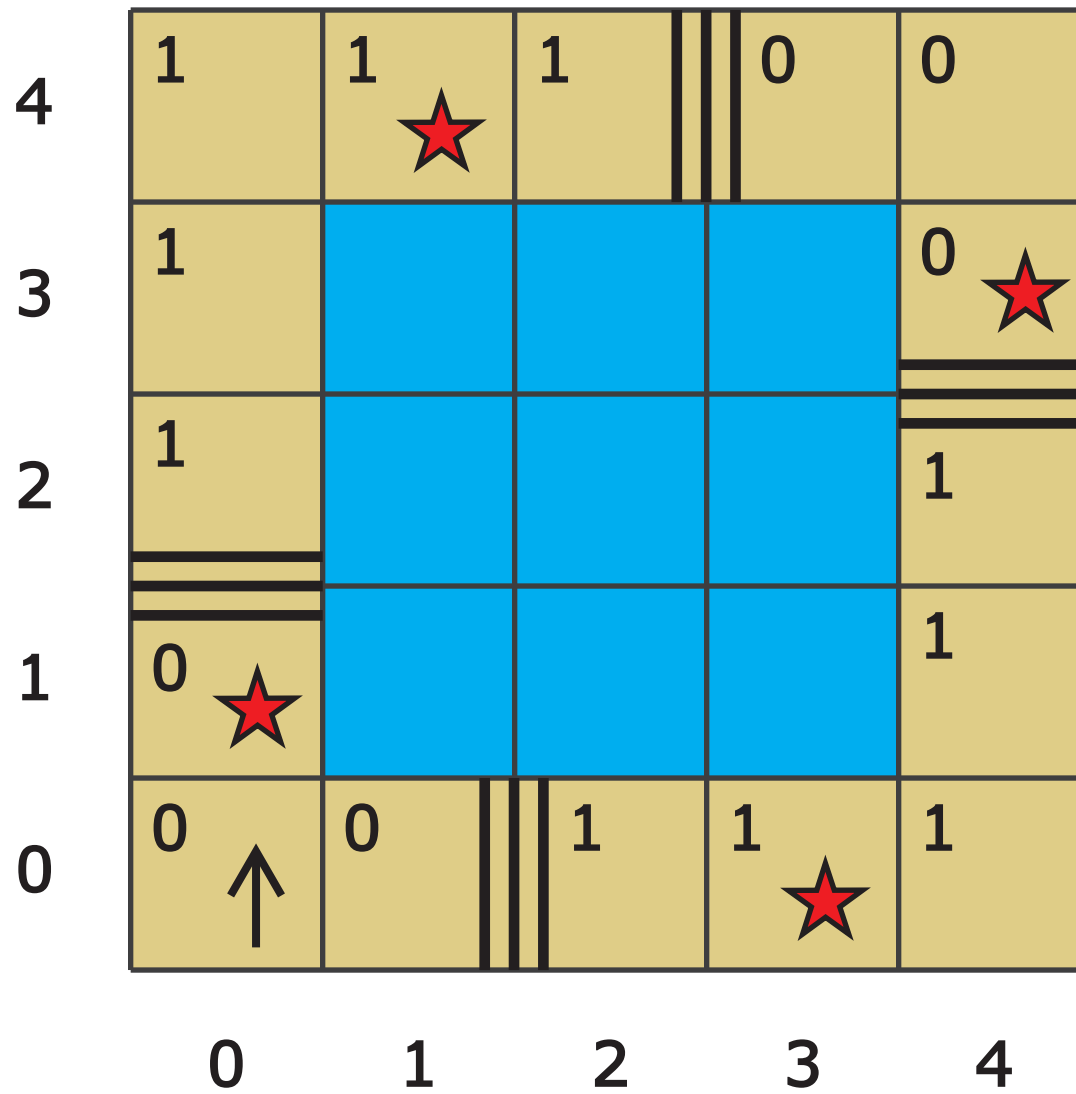
Treasure Hunt



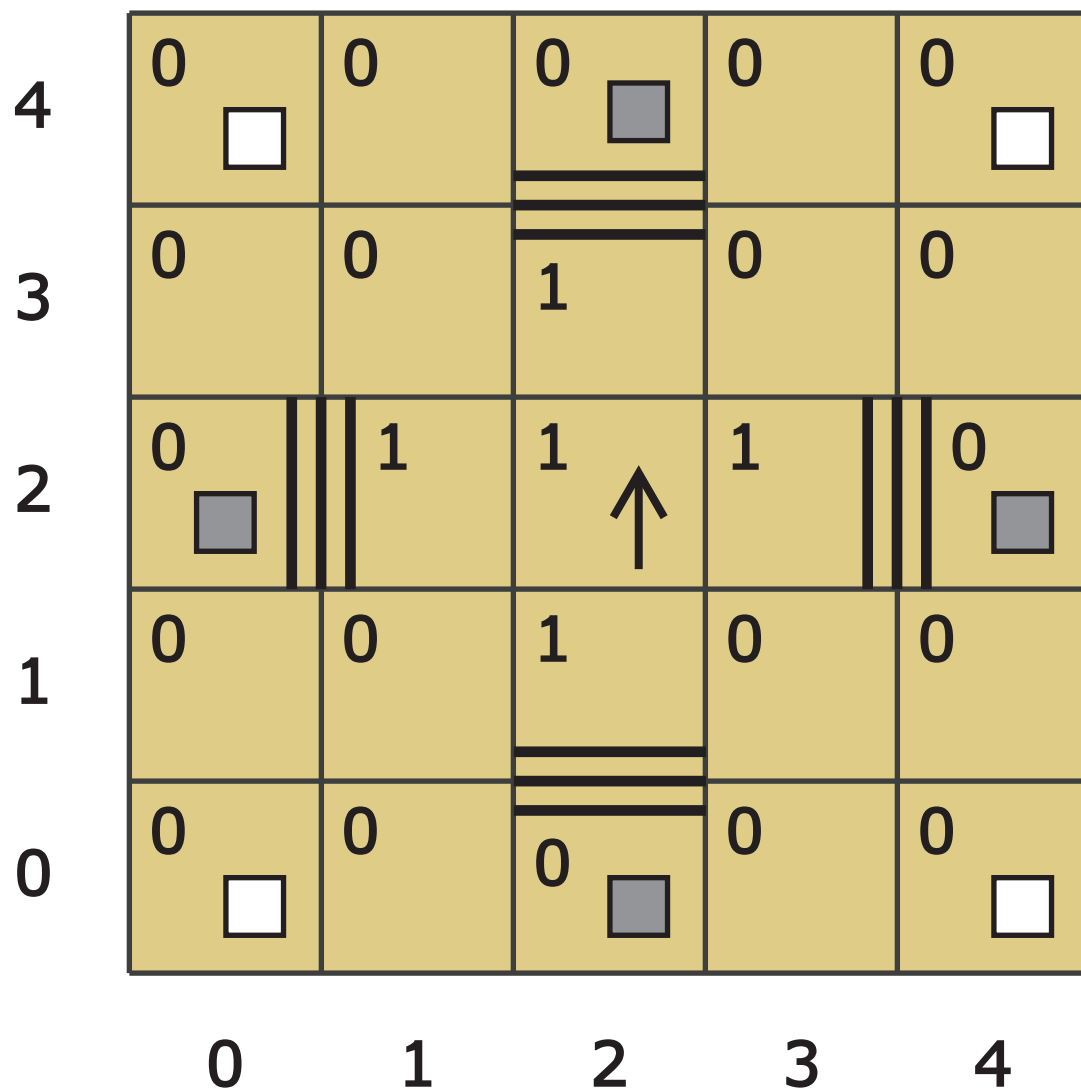
Using Loops



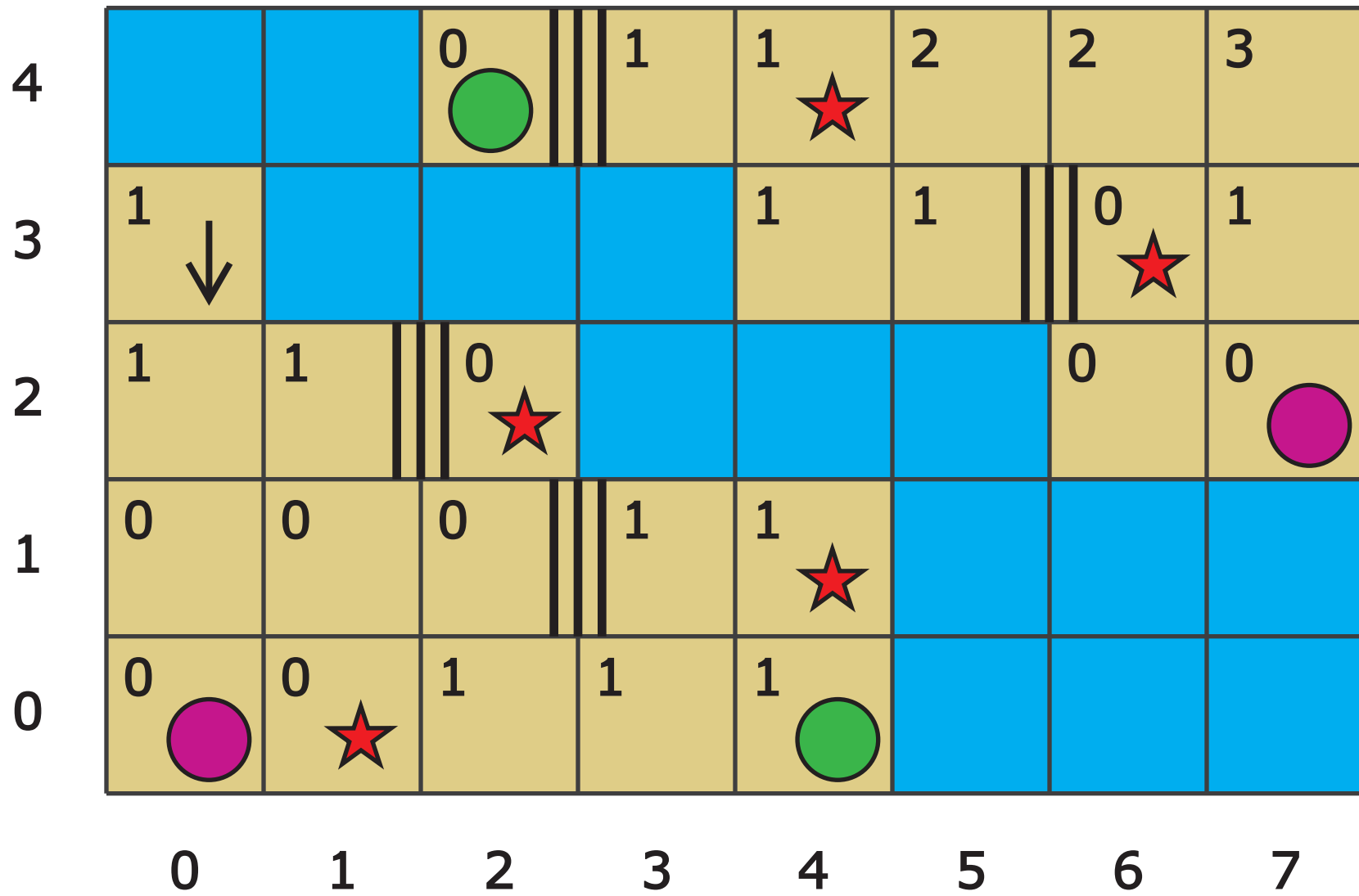
Looping All the Sides



To the Edge and Back



Loop Jumper

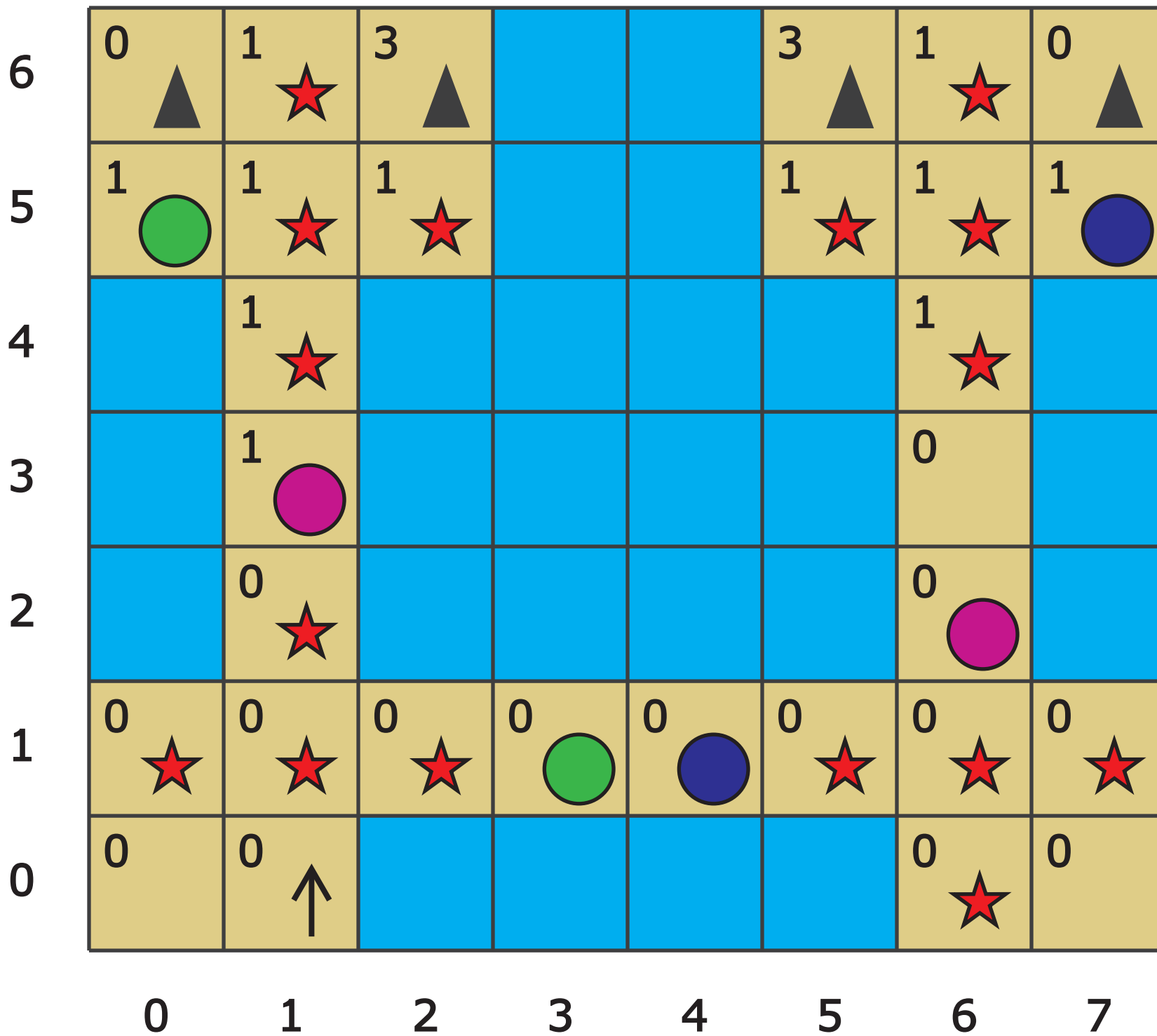


0	0	1	1	1	2	2	3	3	
0	0								
0	0	1	1	1	2	2	3	3	
0	0								
0	0	1	1	1	2	2	3	3	
0	0								
0	0								

Gem Farm

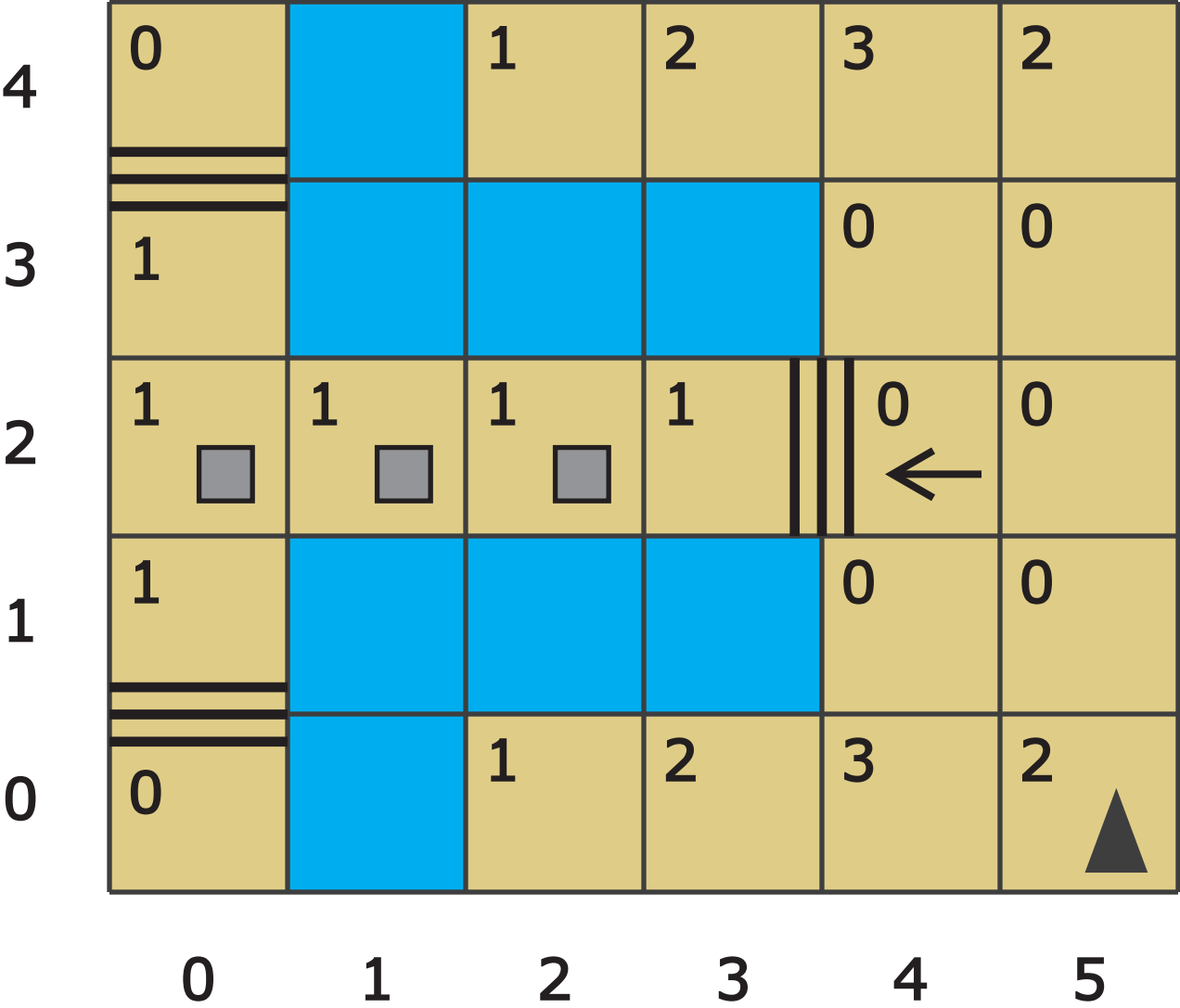


Four Stash Sweep



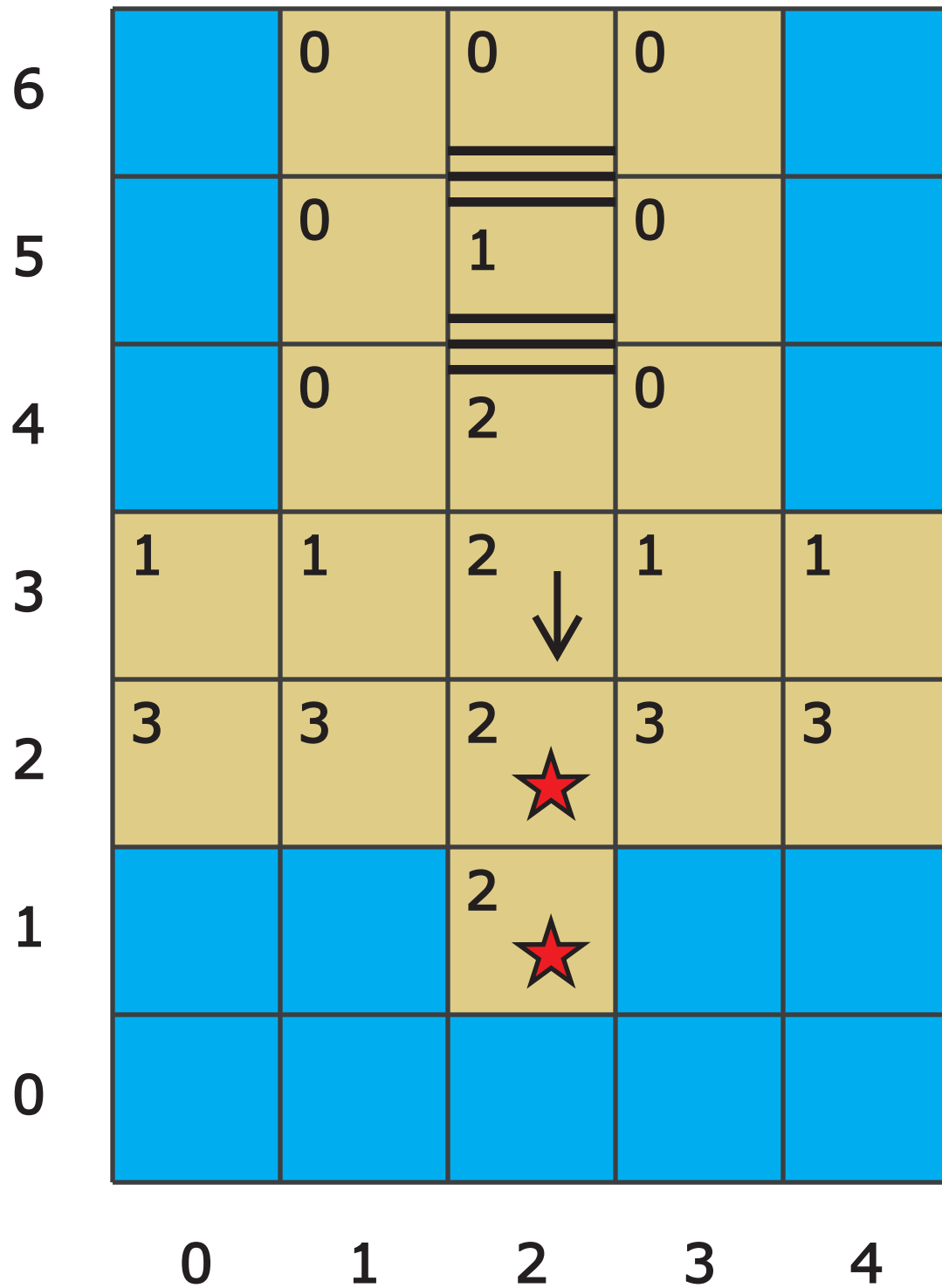
Checking for Switches

Dynamic Layout



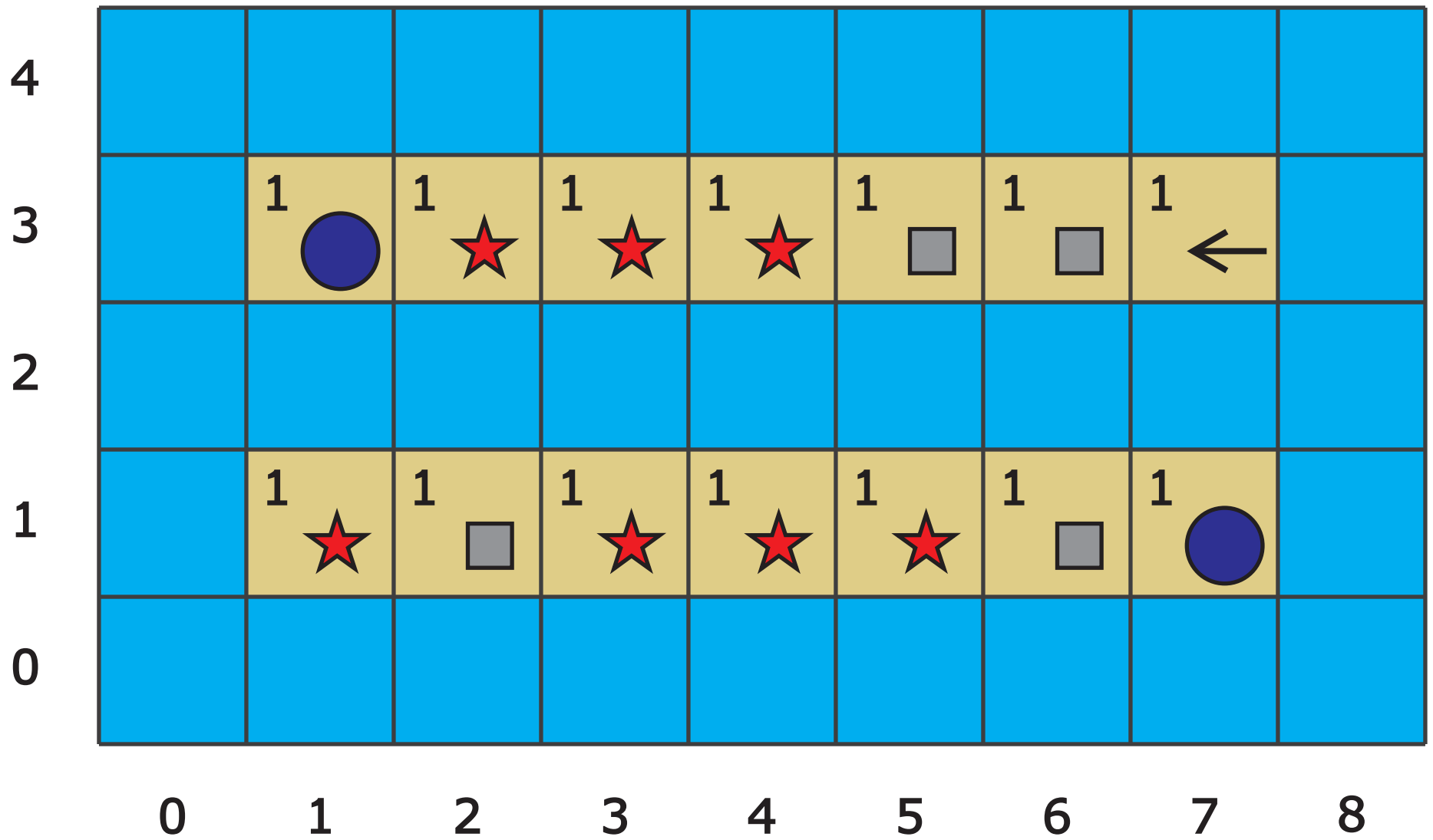
Using else if

Dynamic Layout

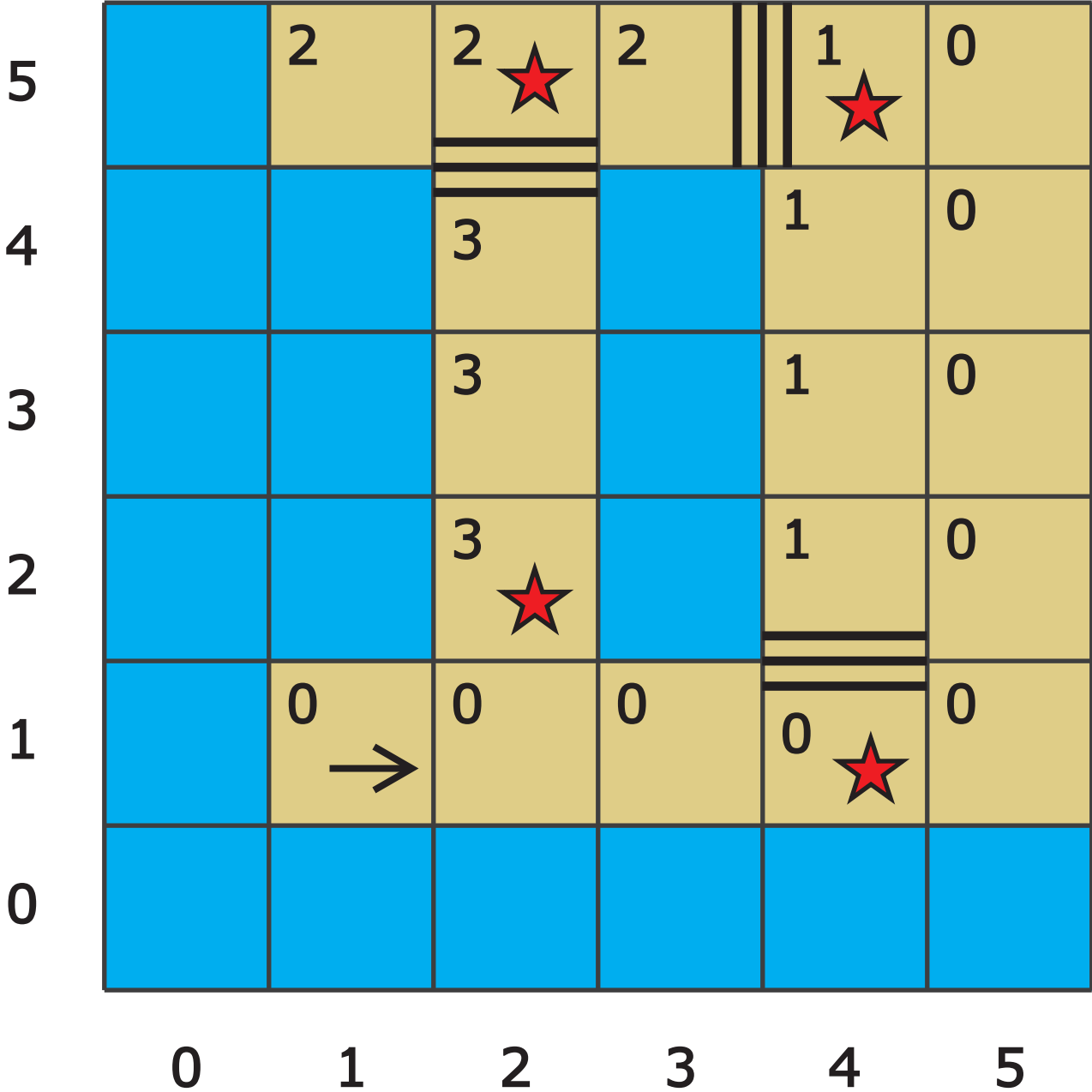


Looping Conditional Code

Dynamic Layout

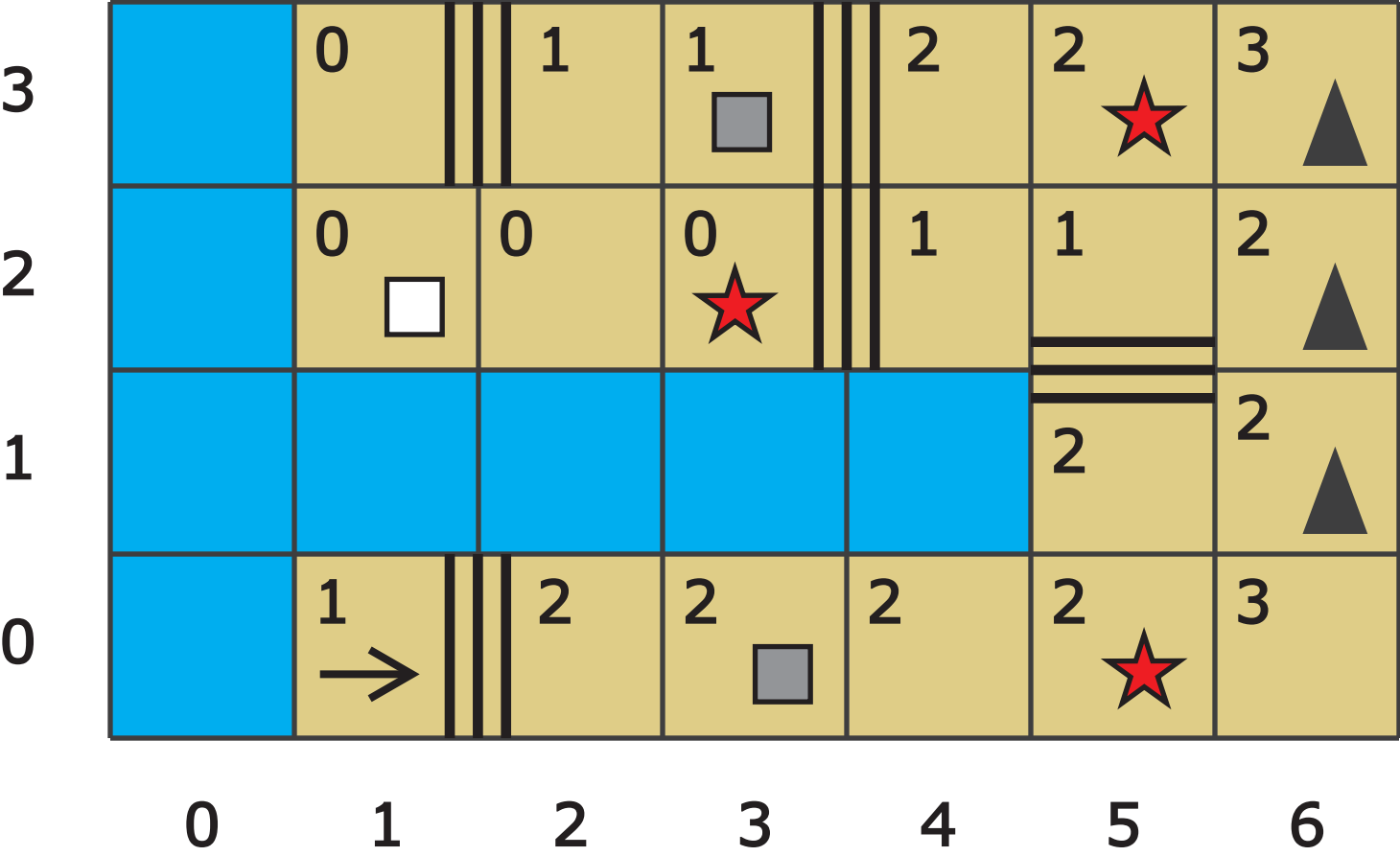


Conditional Climb



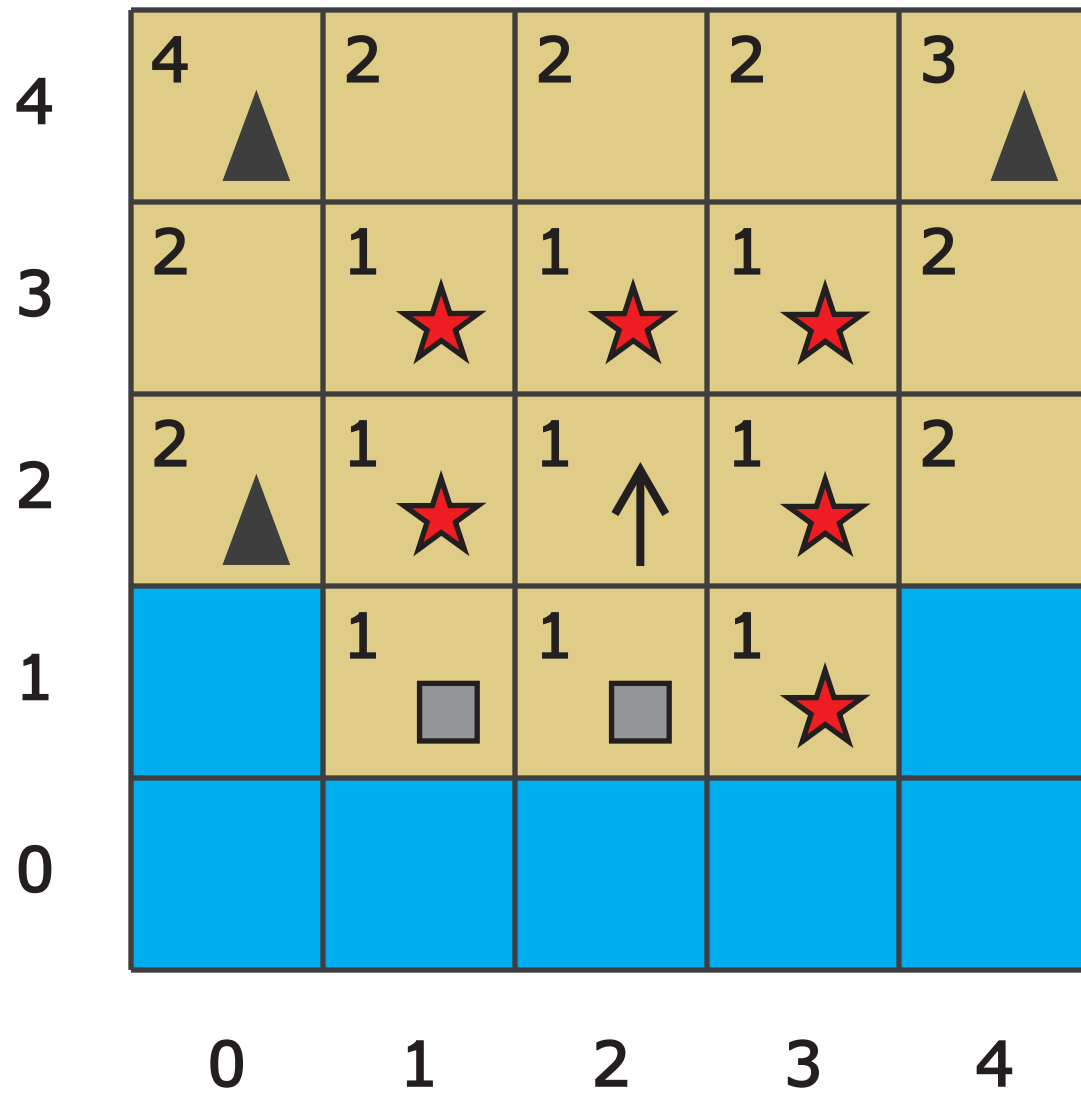
Defining Smarter Functions

Dynamic Layout

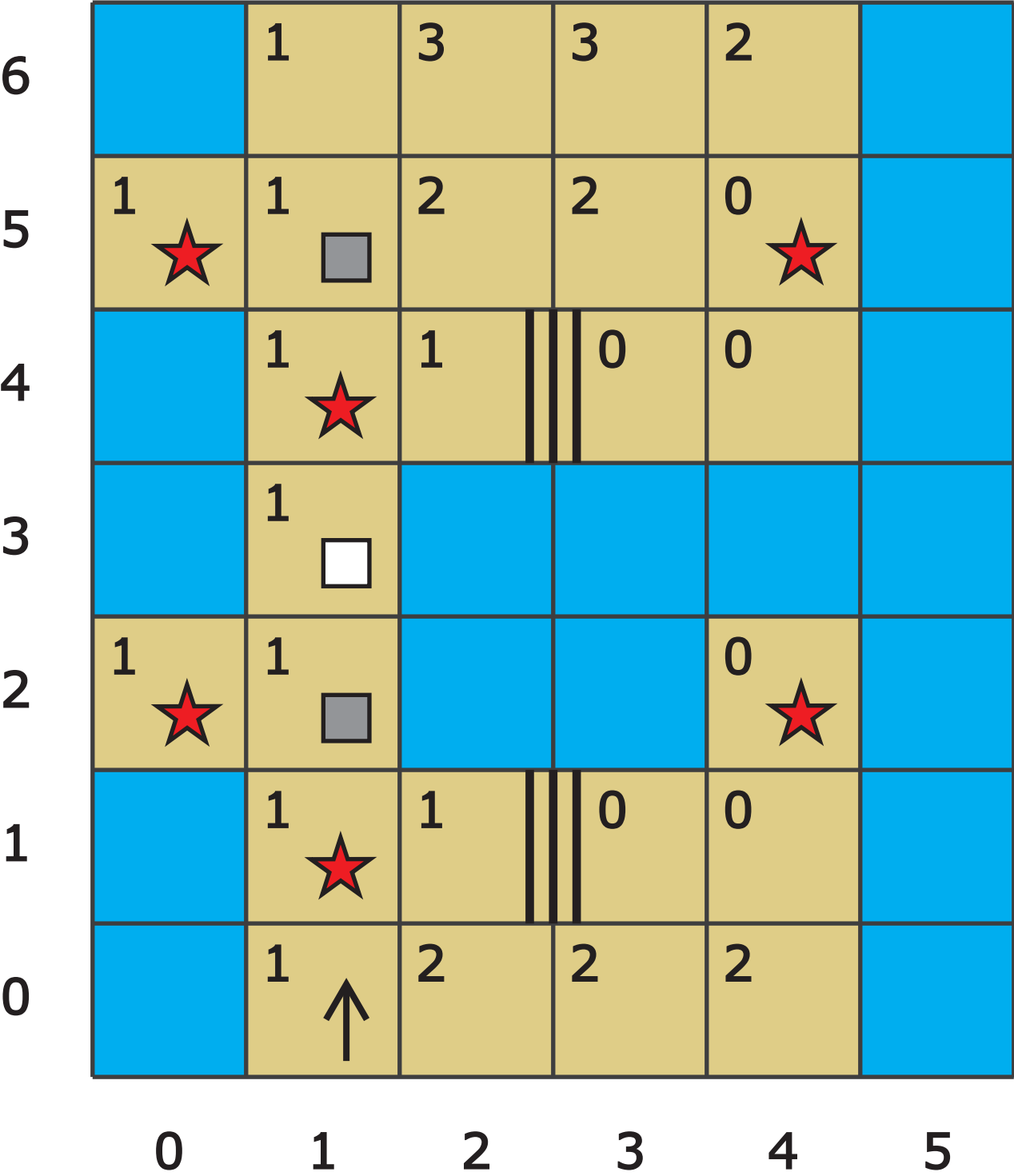


Boxed In

Dynamic Layout

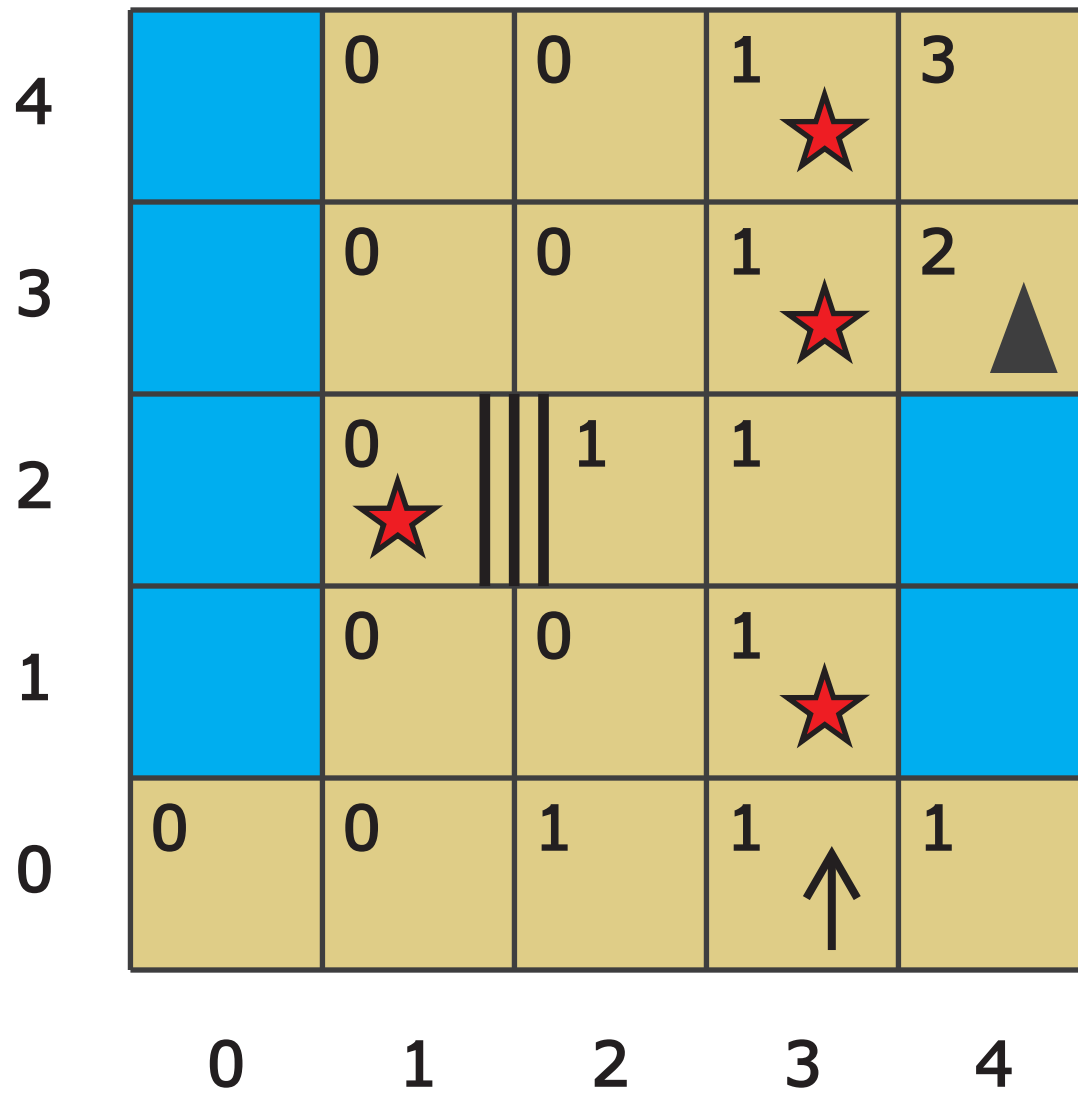


Decision Tree



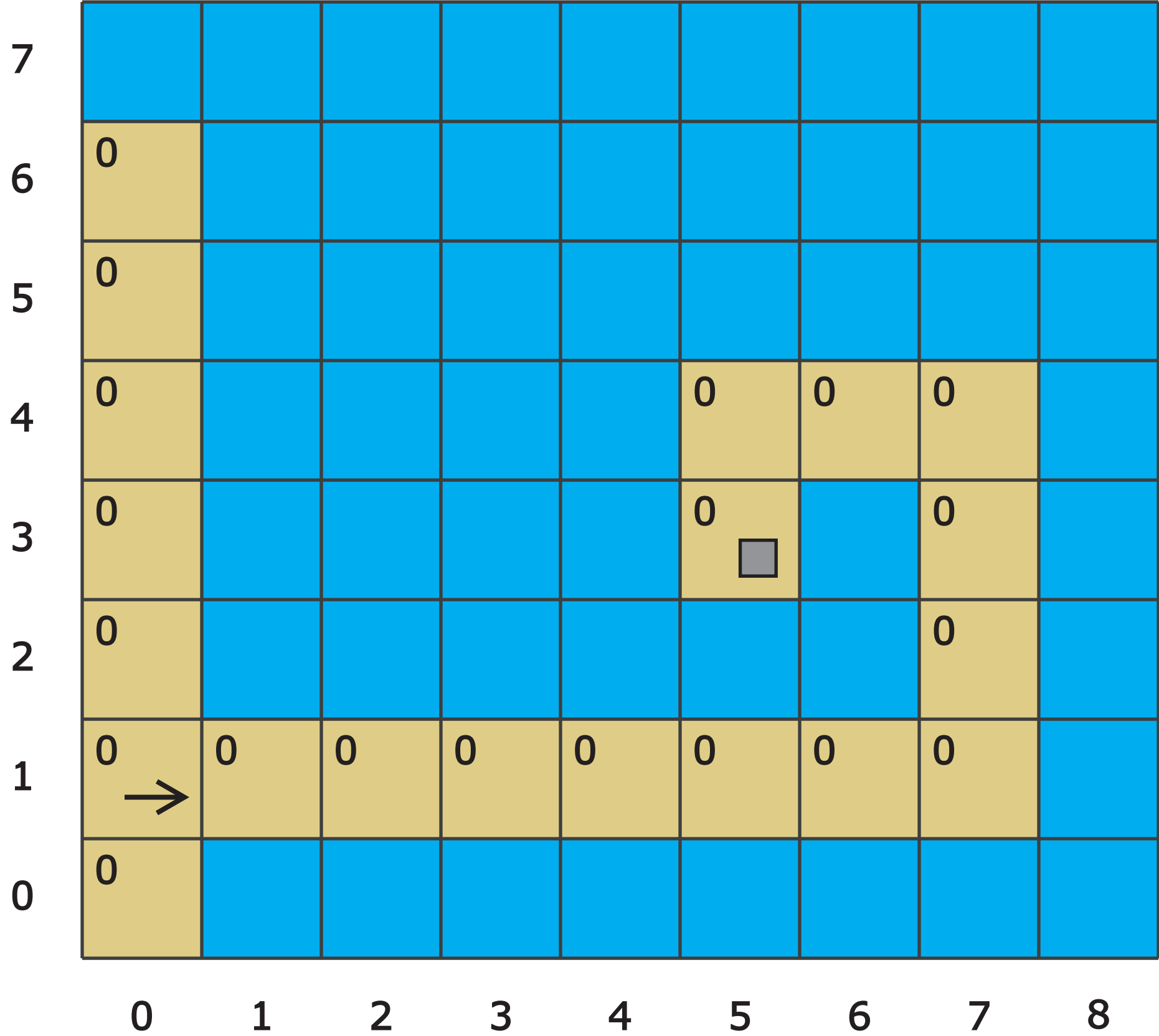
Using the NOT Operator

Dynamic Layout

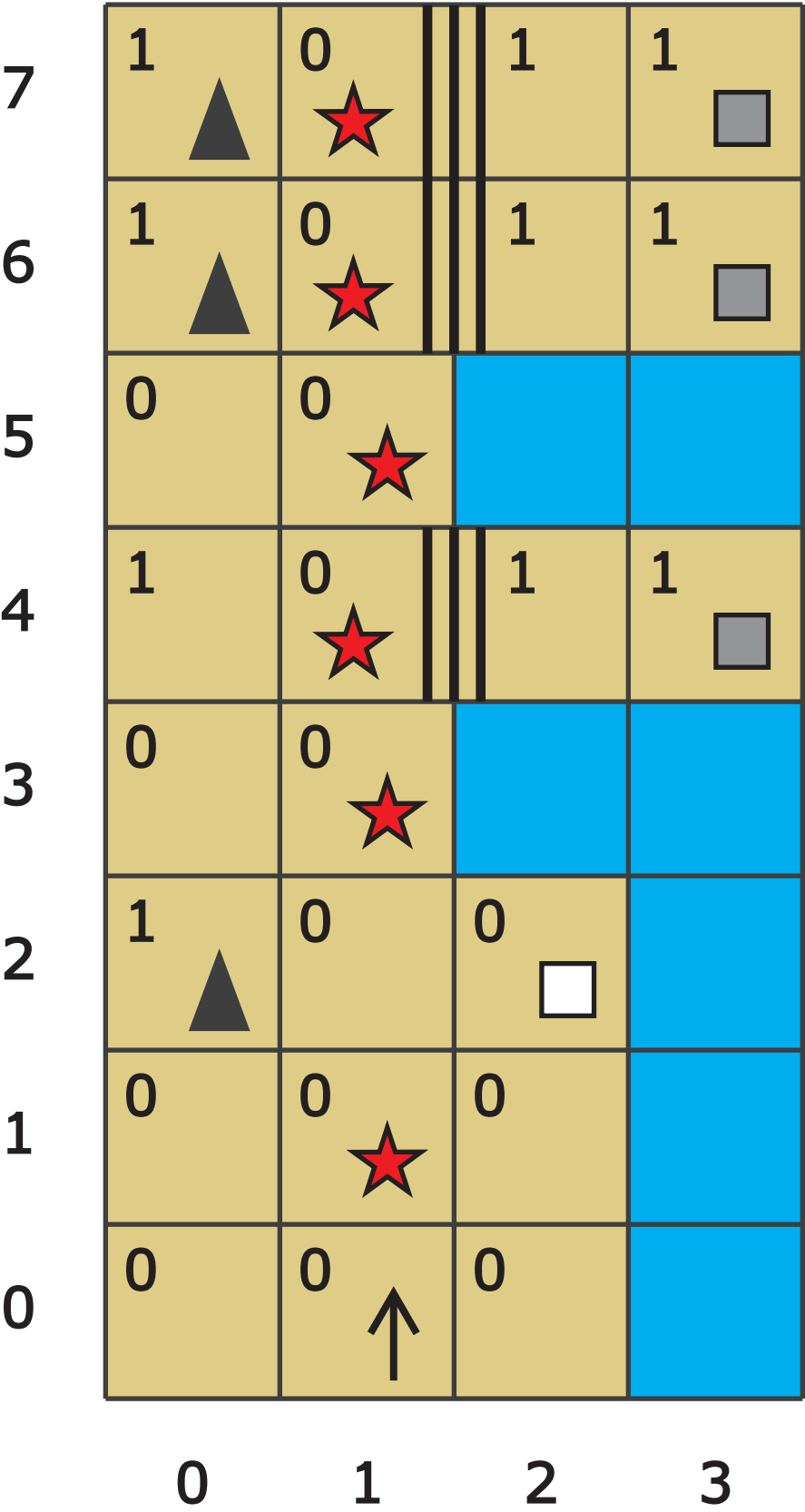


Spiral of NOT

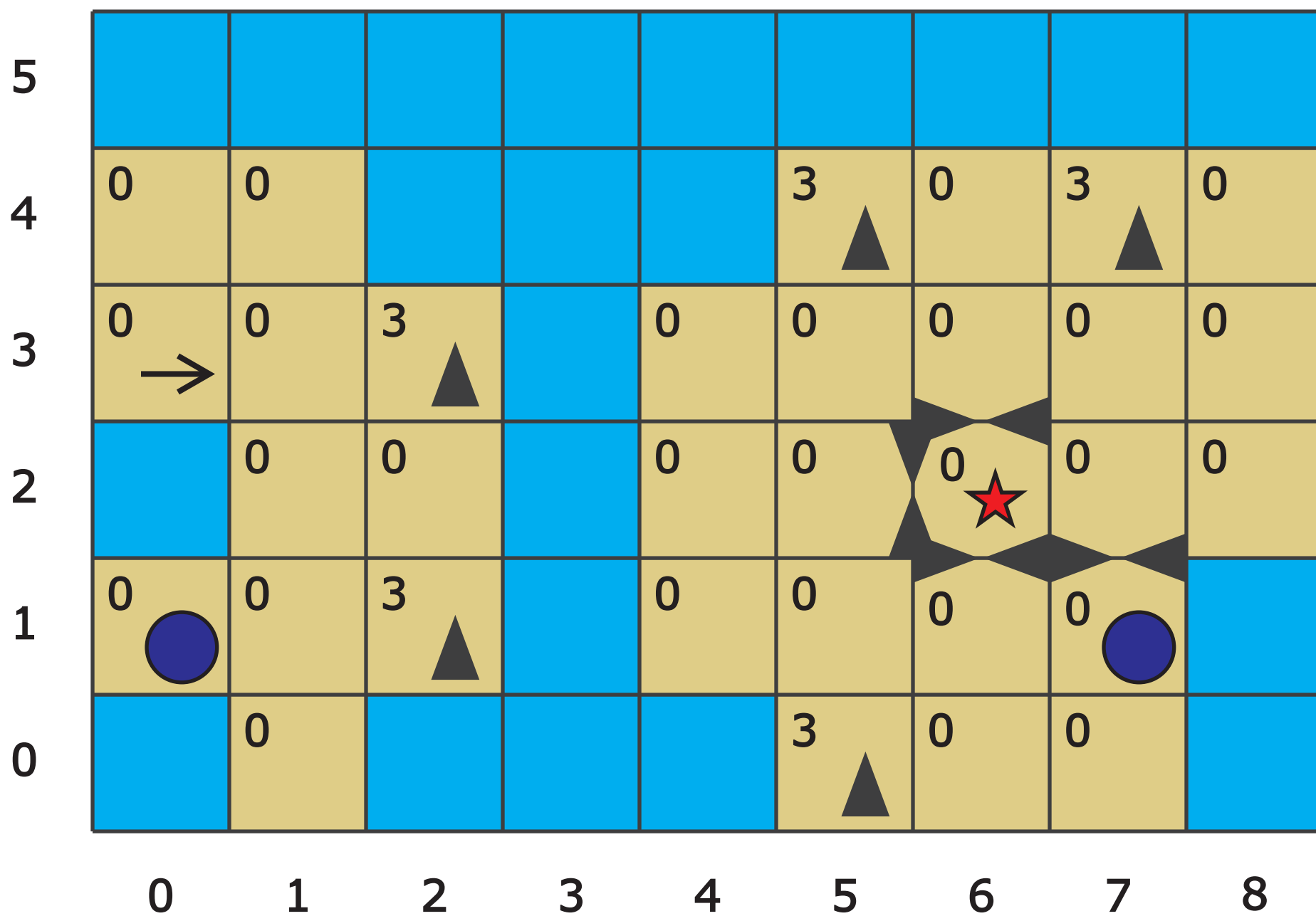
Dynamic Layout



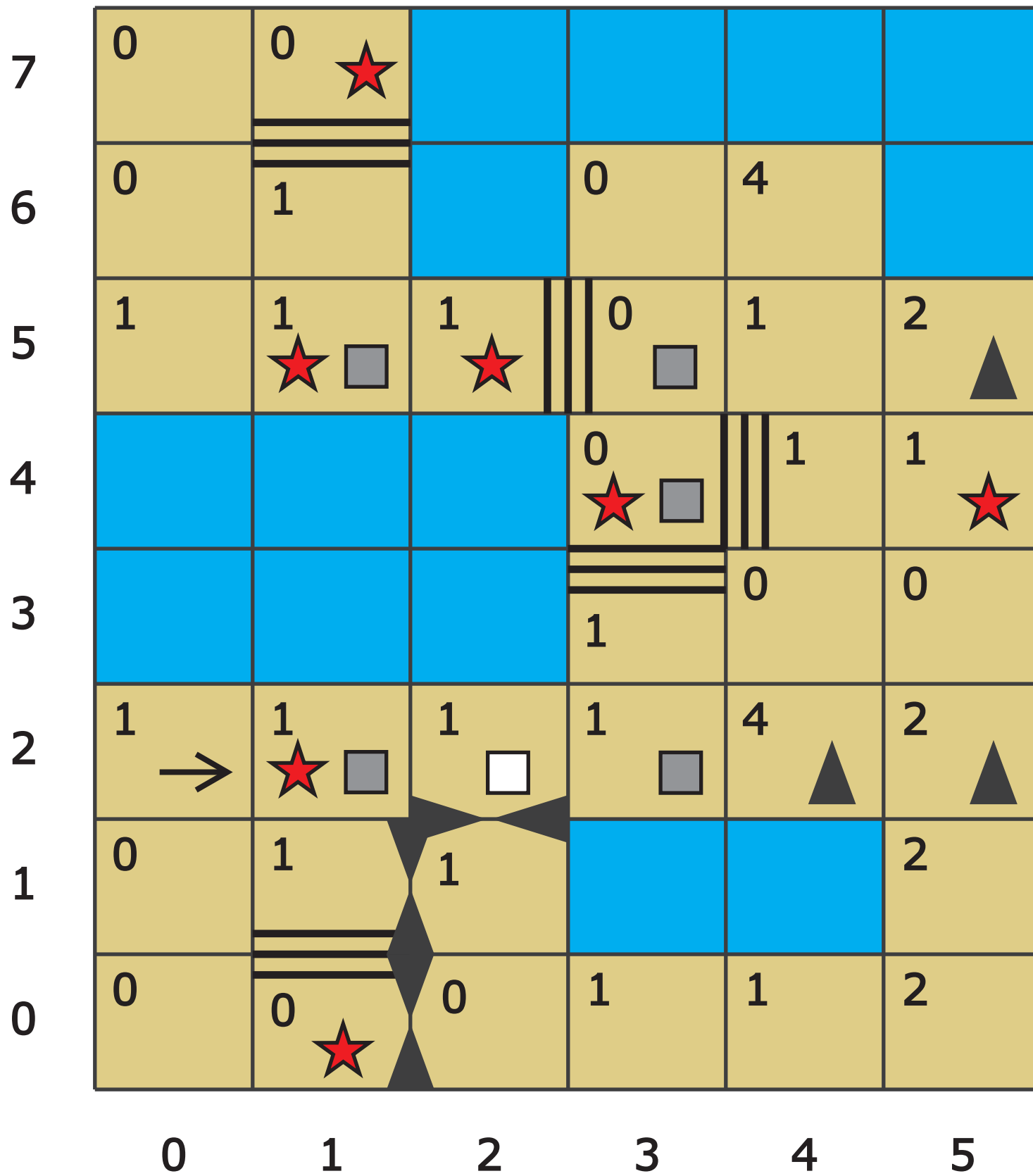
Checking This AND That



Checking This OR That

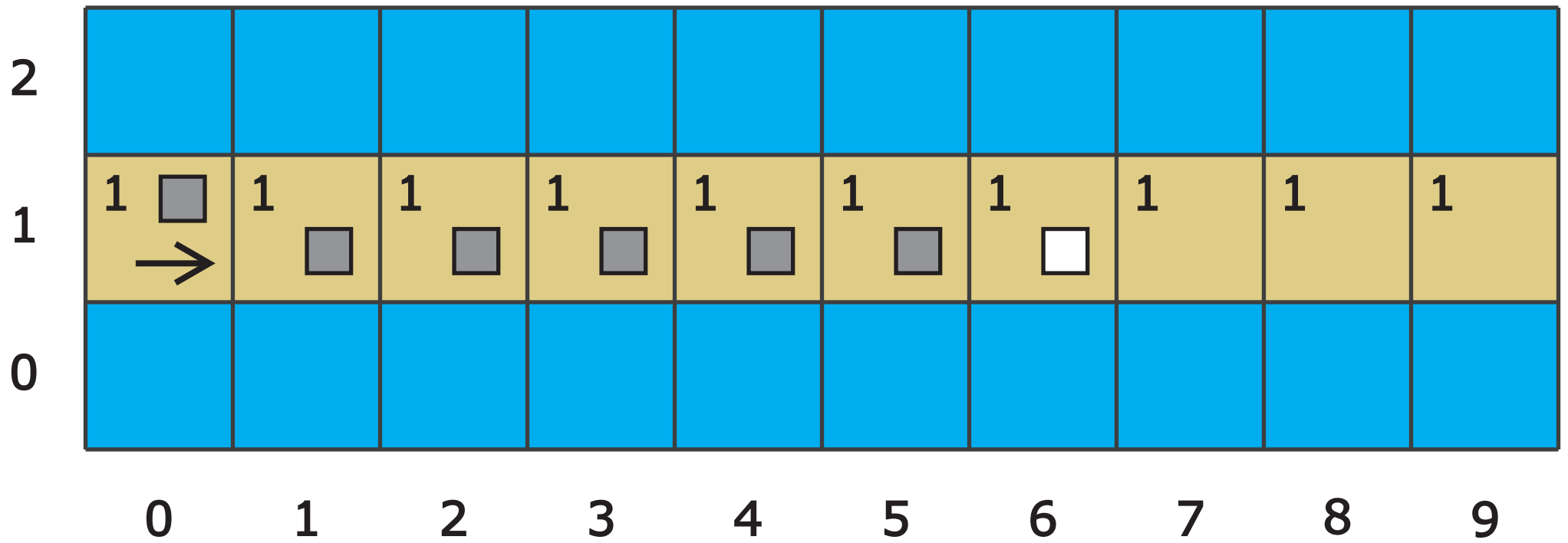


Logical Labyrinth



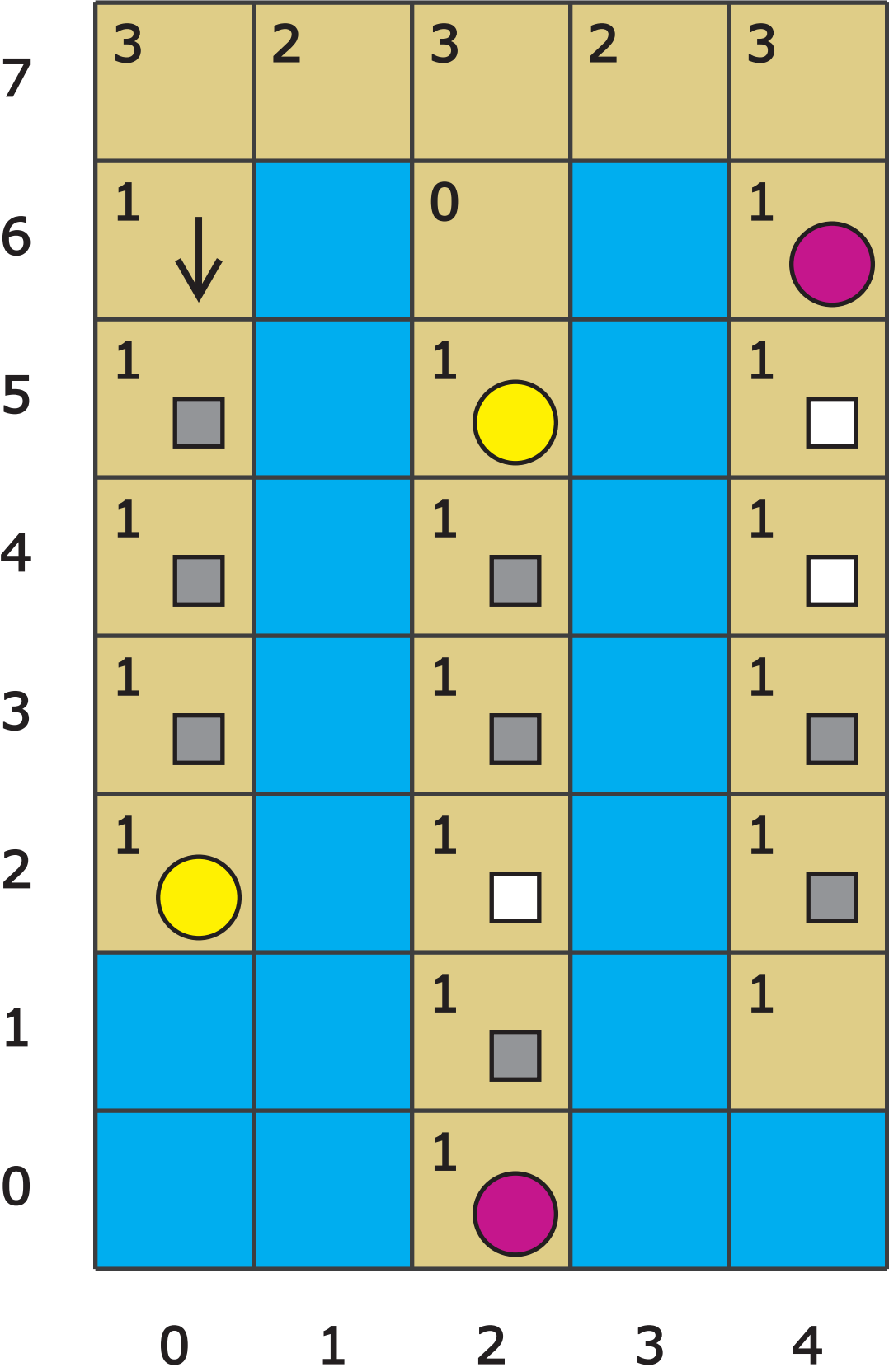
Running Code While...

Dynamic Layout

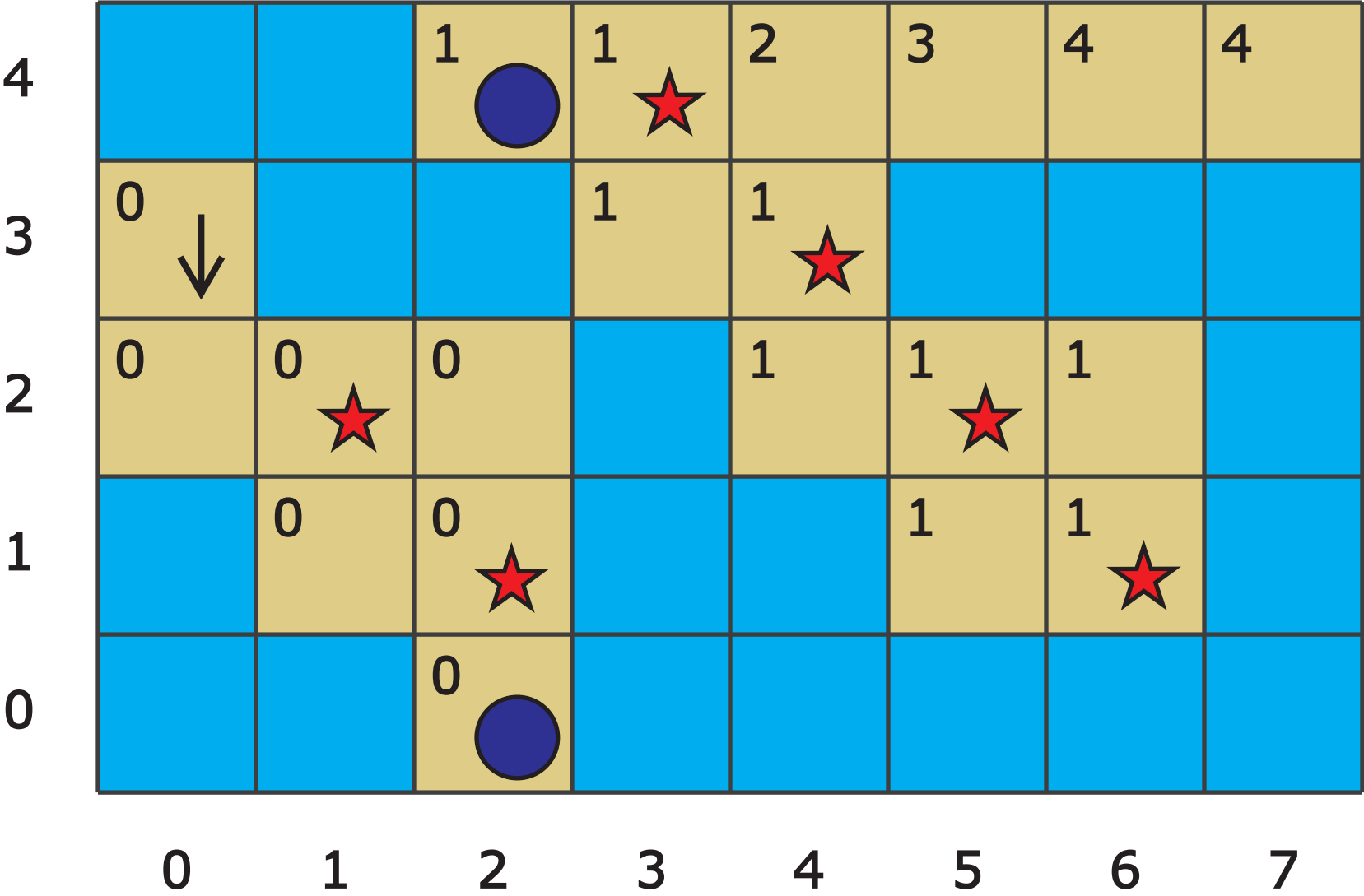


Creating Smarter While Loops

Dynamic Layout

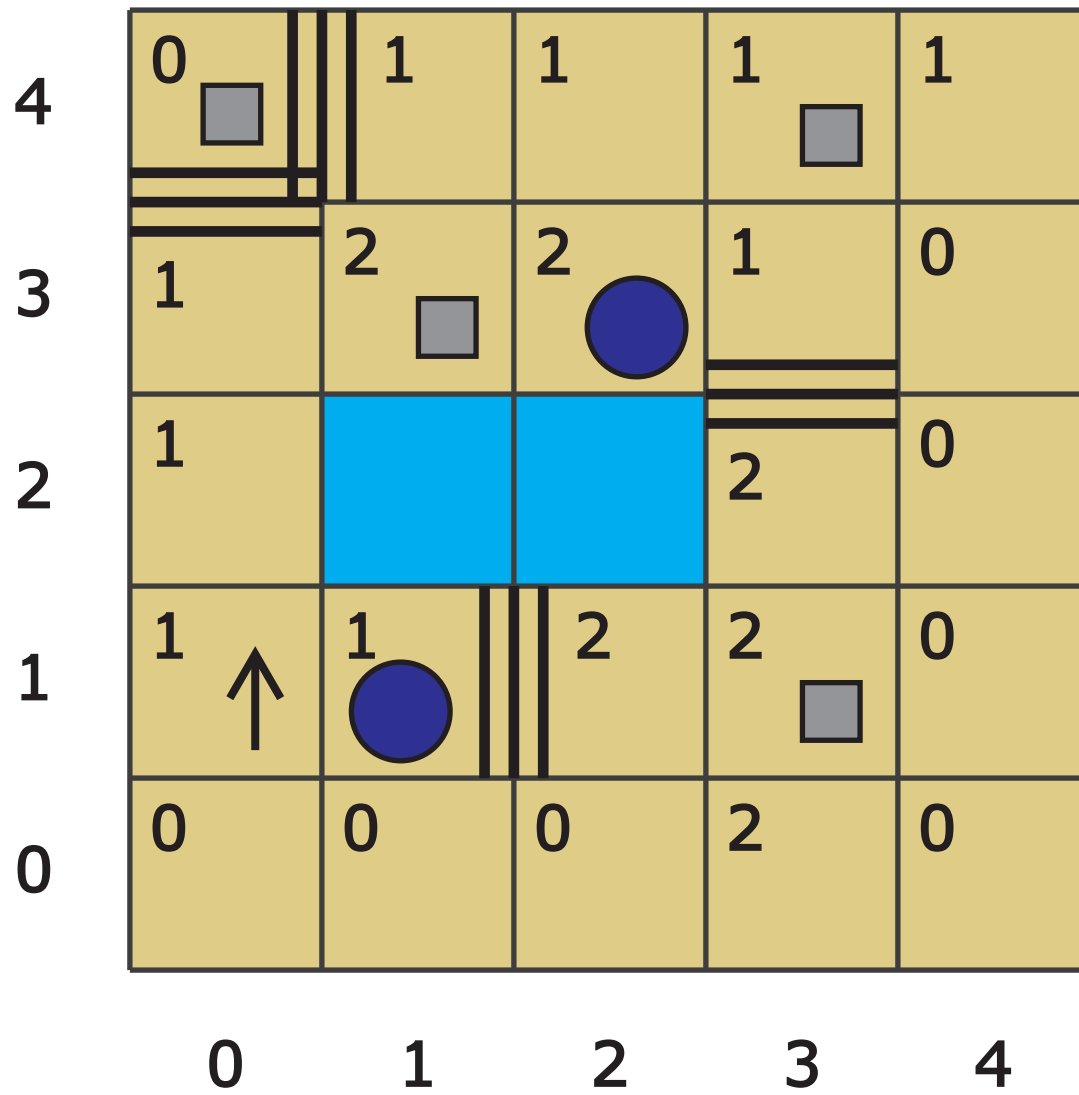


Choosing the Correct Tool

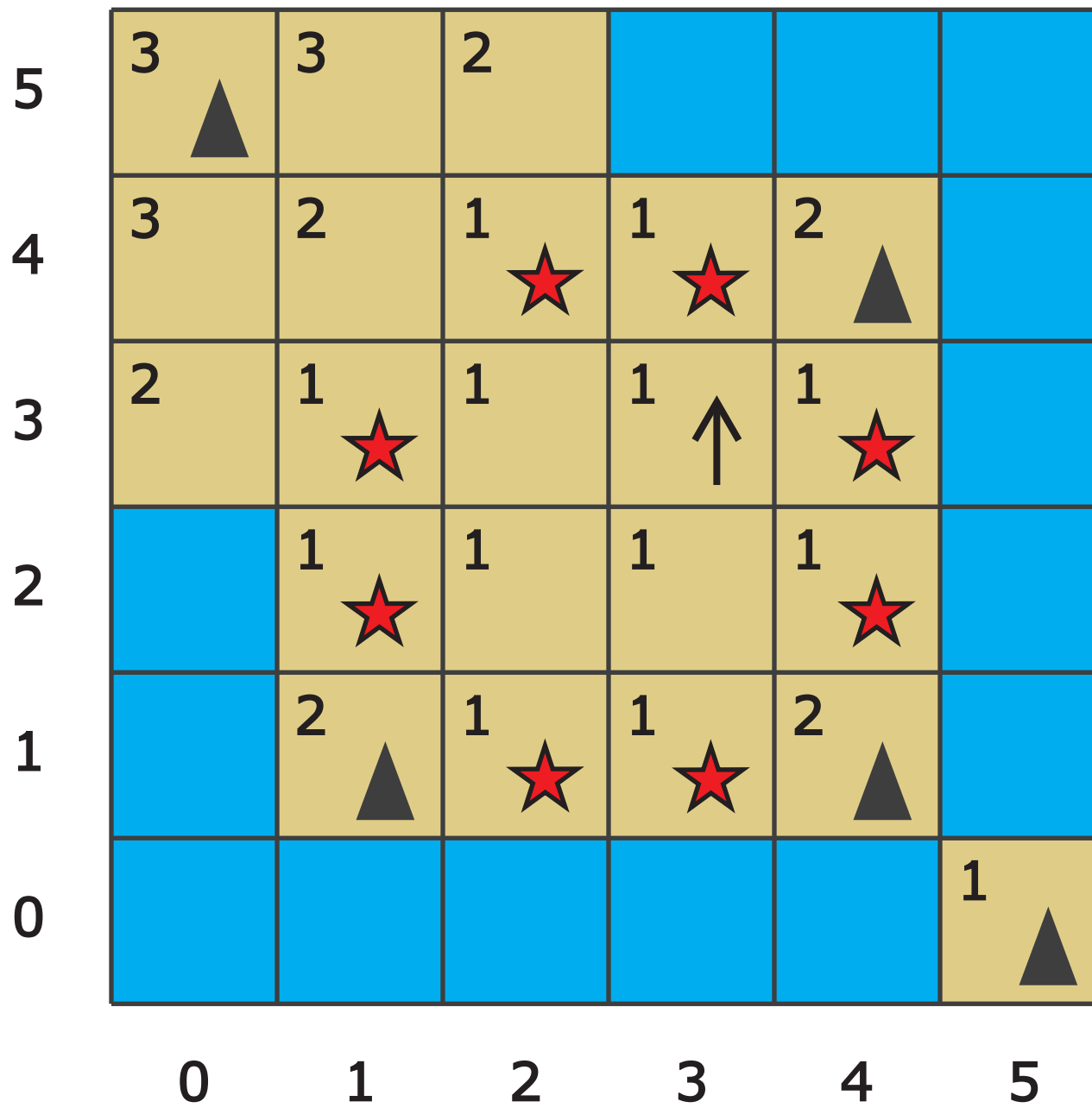


Four by Four

Dynamic Layout

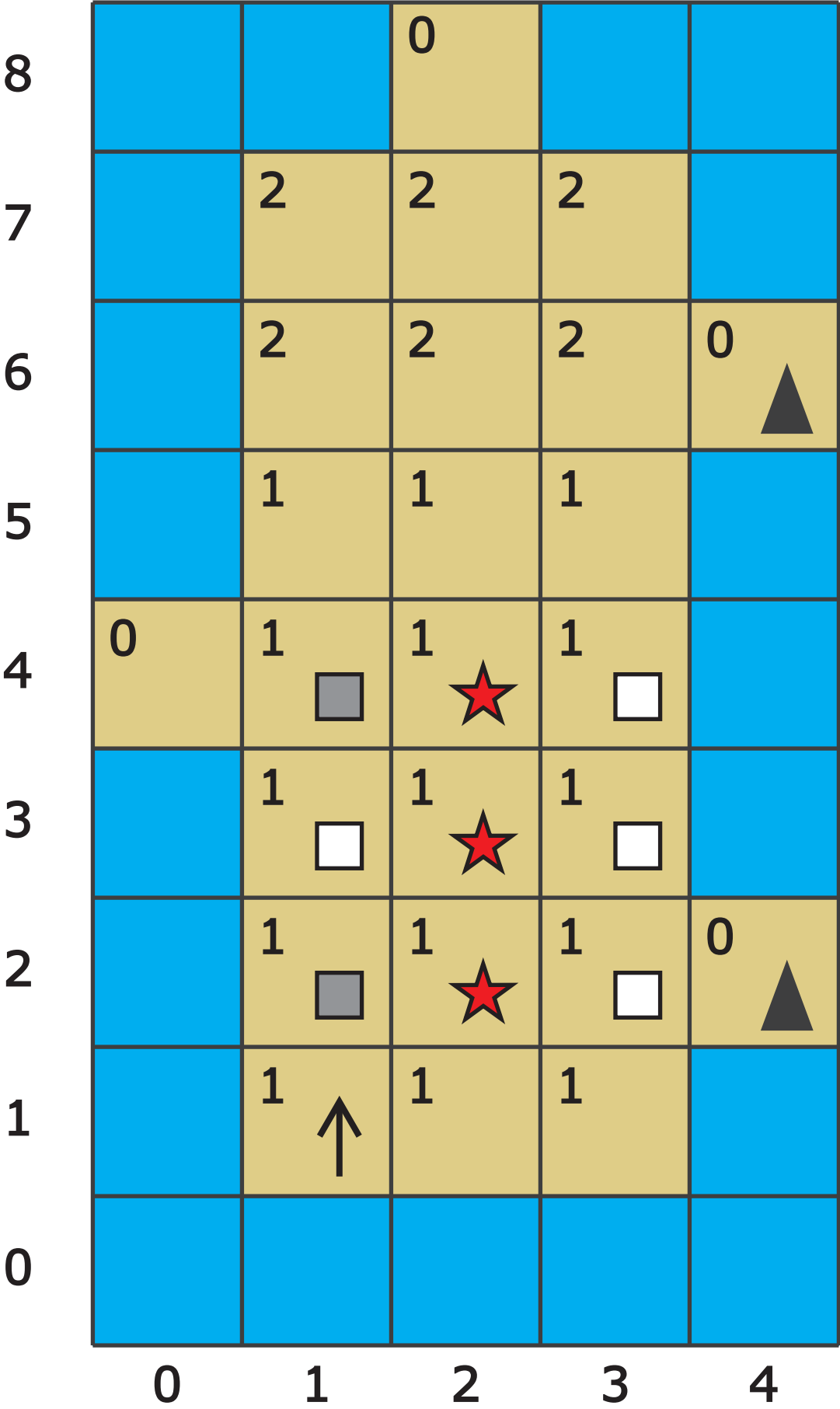


Turned Around

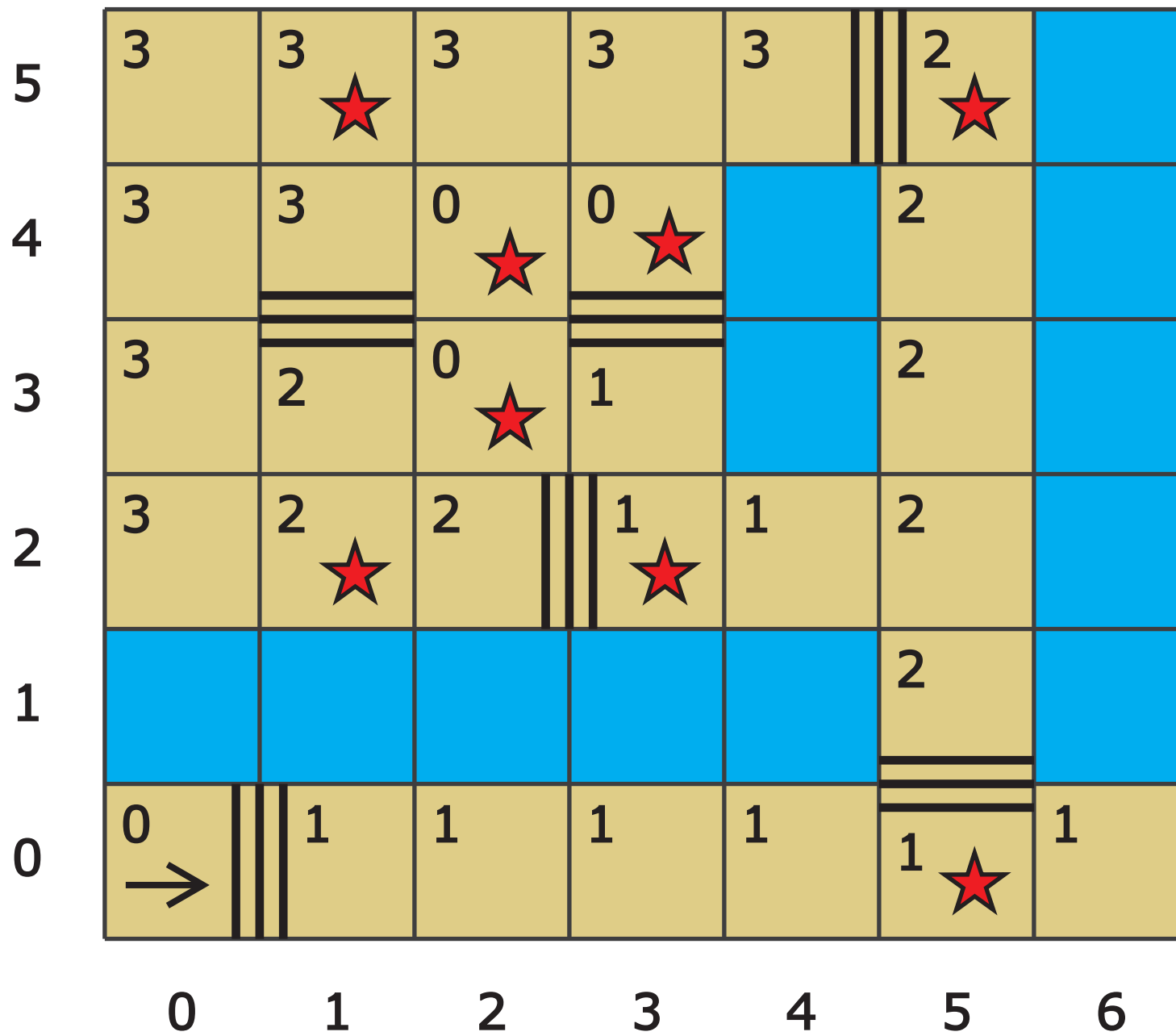


Land of Bounty

Dynamic
Layout

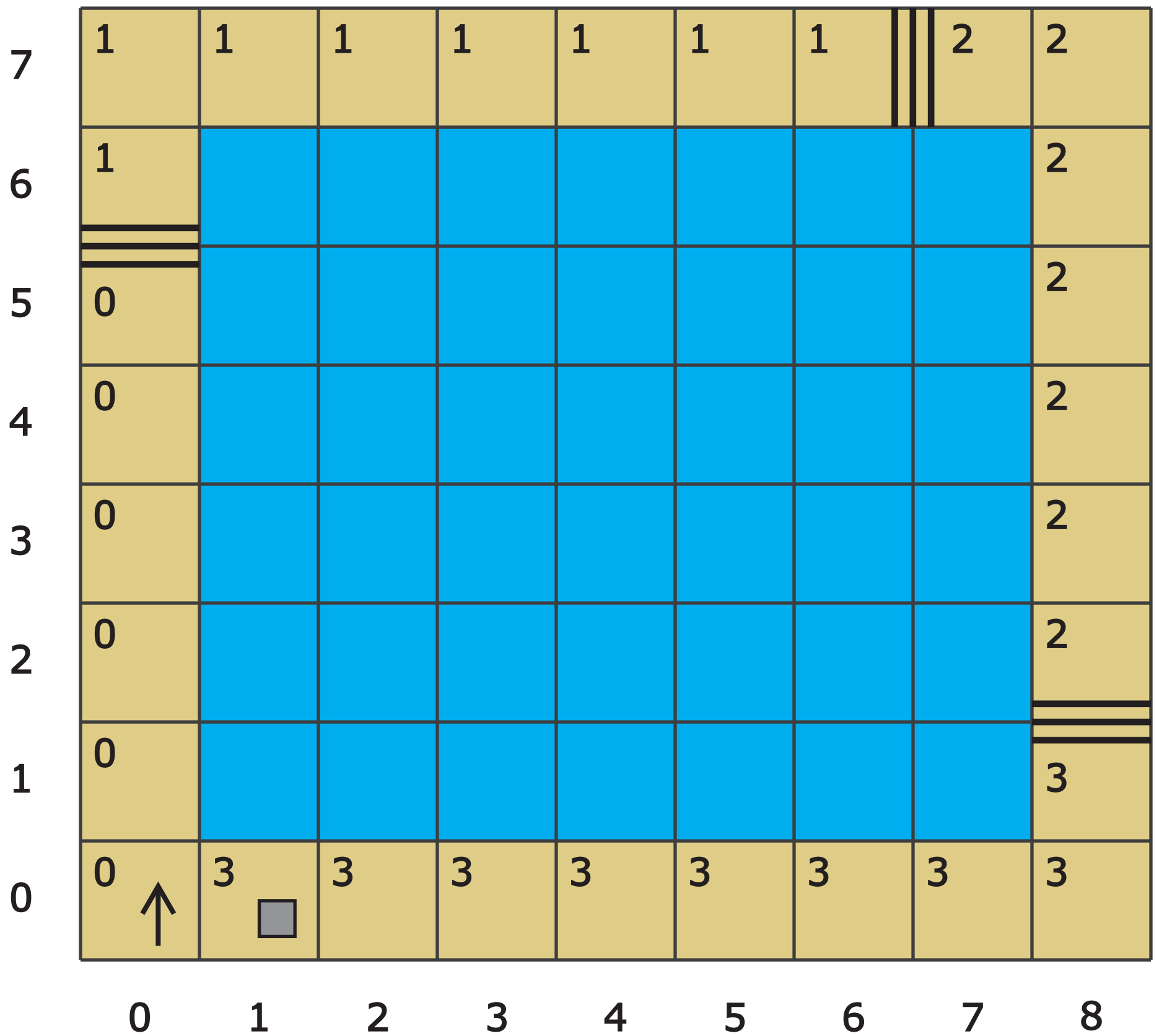


Nesting Loops



Random Rectangles

Dynamic Layout



Dynamic Layout



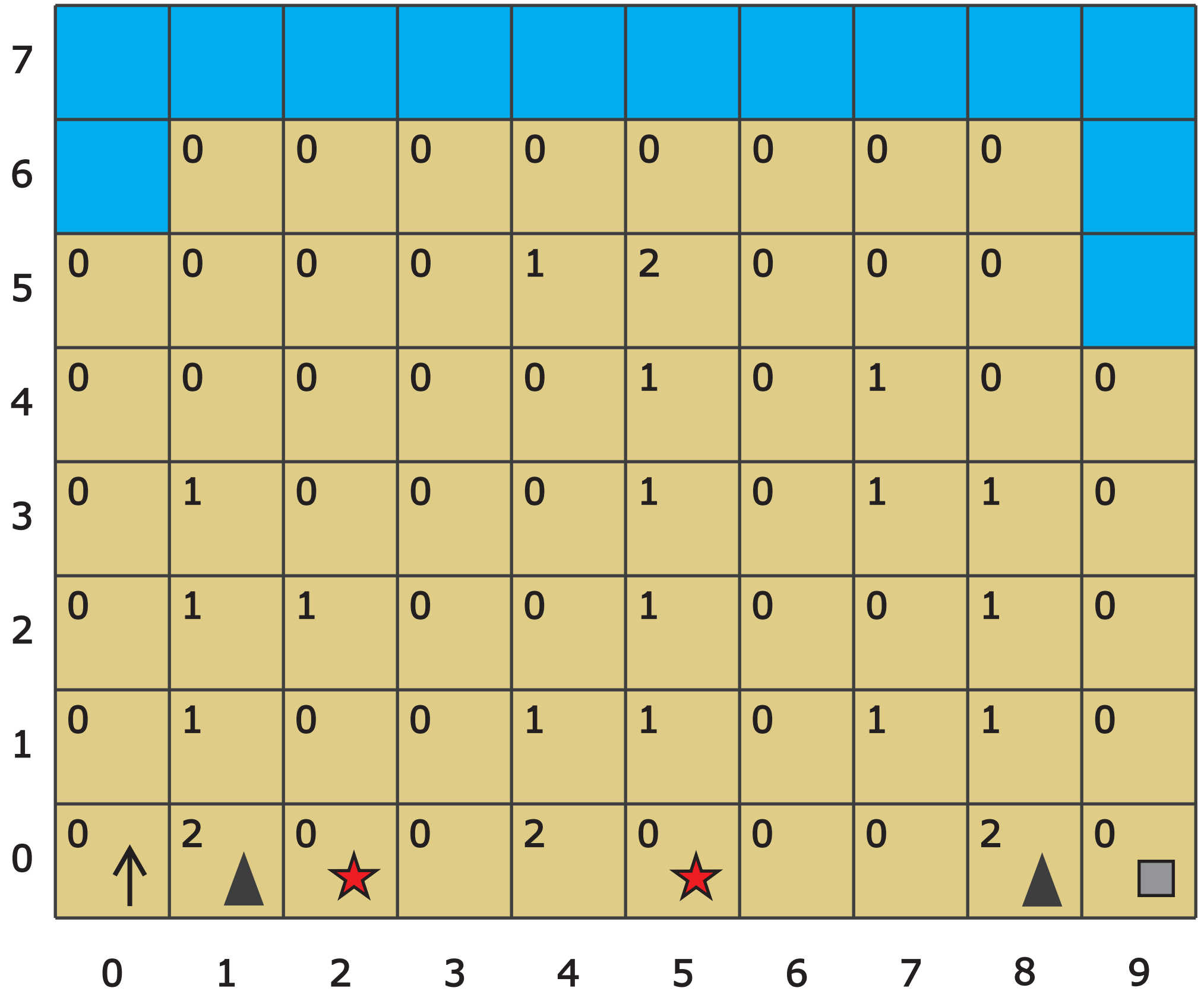
The Right-Hand Rule

Dynamic Layout

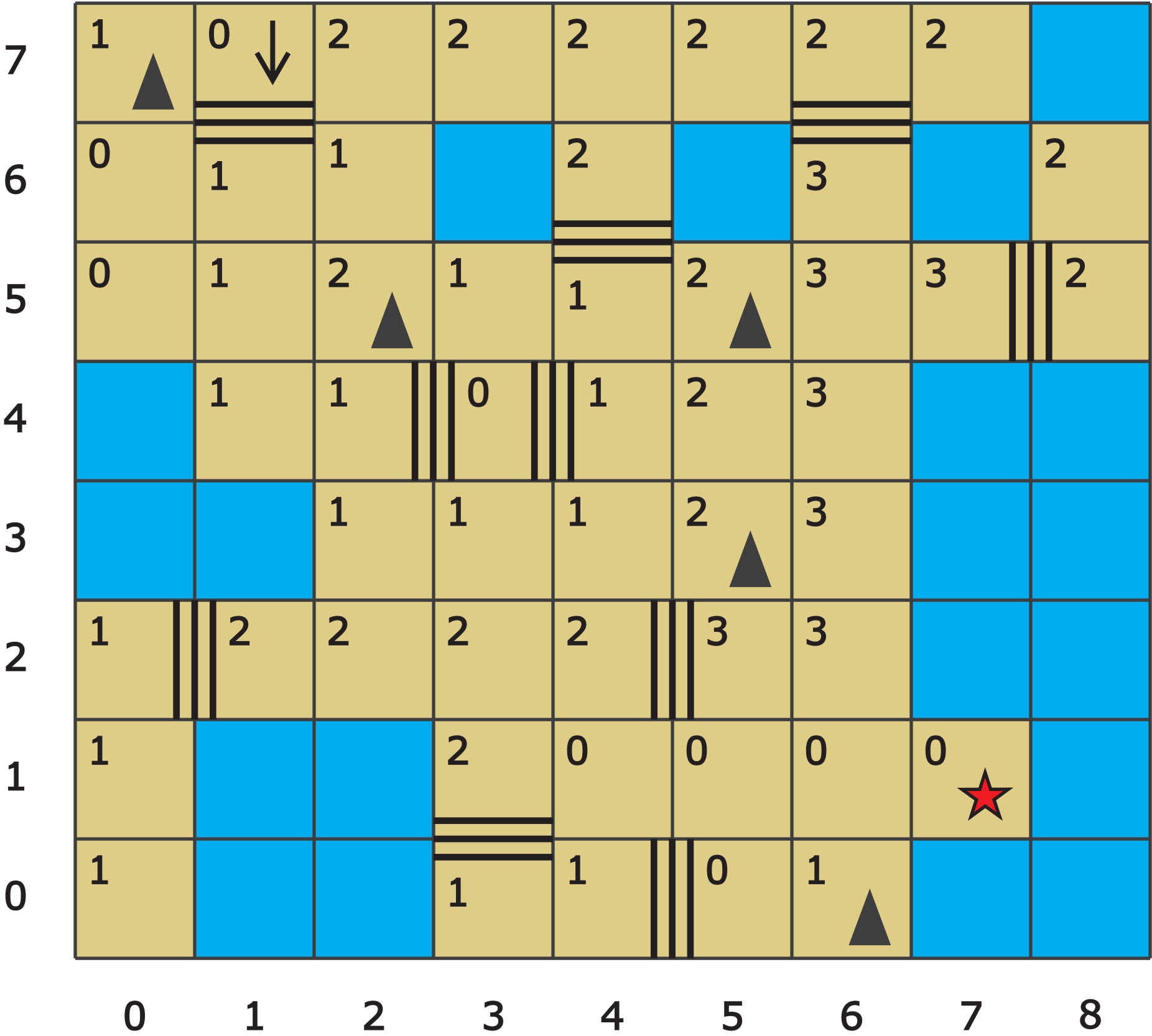
6	0	0	0	0	0	0	0	0	0	
5	0	0	0	0	0	0	0	0	0	
4	0	0	0	0	0	0	0	0	0	
3	0	0	0	0	0	0	0	0	0	
2	0	0	0	0	0	0	0	0	0	
1	0	0	0	0	0	0	0	0	0	
0	0 ↑	1	0 ★	1	0 ★	1	1	0 ★	1 ■	
	0	1	2	3	4	5	6	7	8	9

Adjusting Your Algorithm

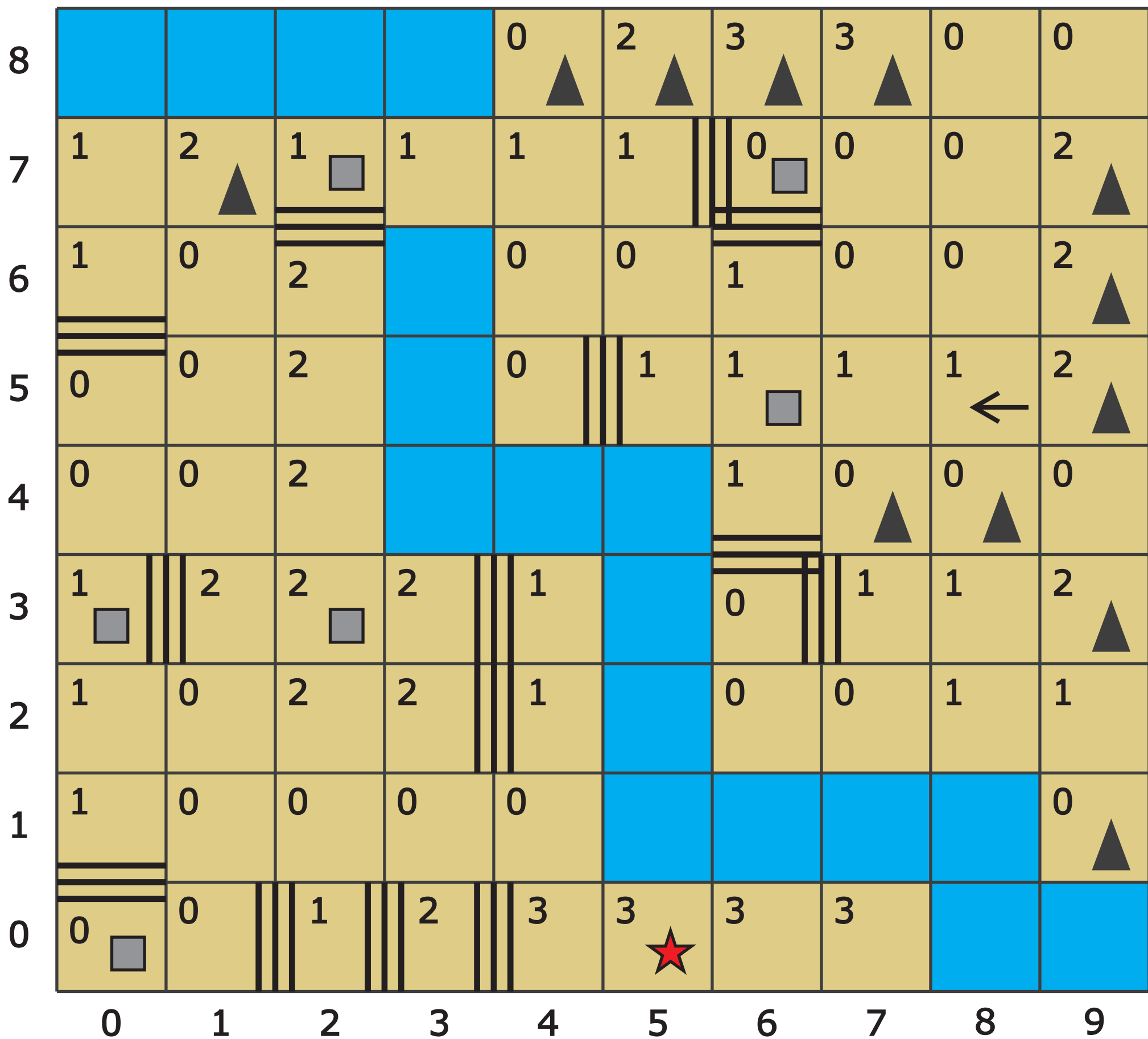
Dynamic Layout



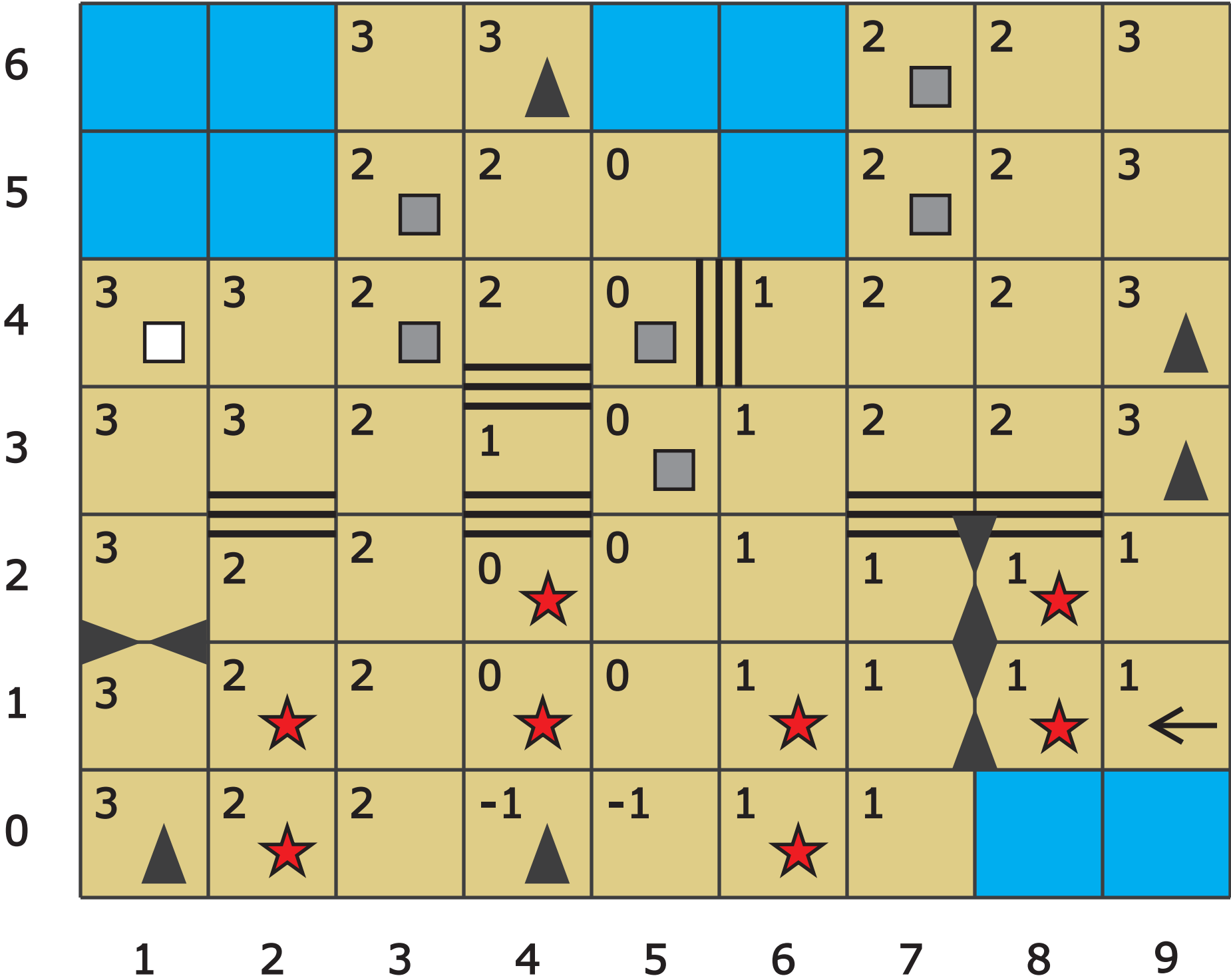
Conquering a Maze



Which Way to Turn?



Roll Right, Roll Left



Produced by:

LightHouse for the Blind and Visually Impaired

lighthouse-sf.org