

2012年度卒業演習発表

AIプログラム大会向けのコード作成

谷聖一 研究室 並木 彰浩
荒金 誠二

目次

1. はじめに

2. SamurAI Coding: 概要

3. SamurAI Coding: 提出した犬Agentの動作

4. SamurAI Coding: 提出した侍Agentの動作

5. SamurAI Coding: 結果と考察

6. Mario AI Championship

目次

1. はじめに

2. SamurAI Coding: 概要

3. SamurAI Coding: 提出した犬Agentの動作

4. SamurAI Coding: 提出した侍Agentの動作

5. SamurAI Coding: 結果と考察

6. Mario AI Championship

1 はじめに

1.1 背景

人工知能(AI)について

【定義】

人間と同等の知能を実現する事を目標とし、
人間と同様な作業をコンピューターで模倣させるシステム

【知能】

判断, 理解, 推理, 計算, 学習...etc

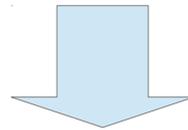
1 はじめに

1.1 背景

人工知能(AI)について

【AIと呼べるのだろうか？】

産業用ロボット，携帯のアラーム機能，デジタルカメラ...etc



【AIの厳密な定義】

個人だけの定義が存在

1 はじめに

1.1 背景

人工知能(AI)について

【発表者の考える定義】

人間と同等の知能を実現する事を目標とし、
人間と同様の判断を部分的にコンピューター
で模倣させるシステム

1 はじめに

1.1 背景

歴史

【始まり】

参考文献
【タイトル】 人工知能概論
【著者】 荒屋 真二

1956年にダートマス会議が開催(マッカーシー, ミンスキー, サイモン, ニューウェルなどが参加)

【現代】

数学, 人類学, 哲学, 遺伝子学など様々な分野でAIは応用され, 現代社会の技術開発に寄与

【未来】

人間と同等の知能を持つAIを開発する技術が確立?

1 はじめに

1.2 動機と目的

動機

ゲームは重要な娯楽の要素であり、
多数のユーザーが利用しているから

目的

SamurAI Codingの大会参加と
MarioAI用プログラム試作

目次

1. はじめに

2. SamurAI Coding: 概要

3. SamurAI Coding: 提出した犬Agentの動作

4. SamurAI Coding: 提出した侍Agentの動作

5. SamurAI Coding: 結果と考察

6. Mario AI Championship

2 SamurAI Coding: 概要

2.1 SamurAI Codingについて

主催

情報処理学会

スポンサー

グリー株式会社

大会の概要

世界中の若手エンジニア, 学生をターゲットにした大規模な国際的コンテスト

2 SamurAI Coding: 概要

2.1 SamurAI Codingについて

2011年度

The screenshot displays the SamurAI Coding game interface. At the top center, a wooden timer shows "Time 1592". Below the timer is a 10x10 game board with various samurai-themed icons and pieces. On the left side, there are three player scoreboards:

- Player 0** (blue header): Shogun, score 1730 / 0382
- Player 1** (red header): Normal, score 1925 / 0000
- Player 2** (green header): Shogun, score 1601 / 0000
- Player 3** (yellow header): Normal, score 1692 / 0000

At the bottom center, the logo "WASEDA × GREE SamurAI coding" is displayed.

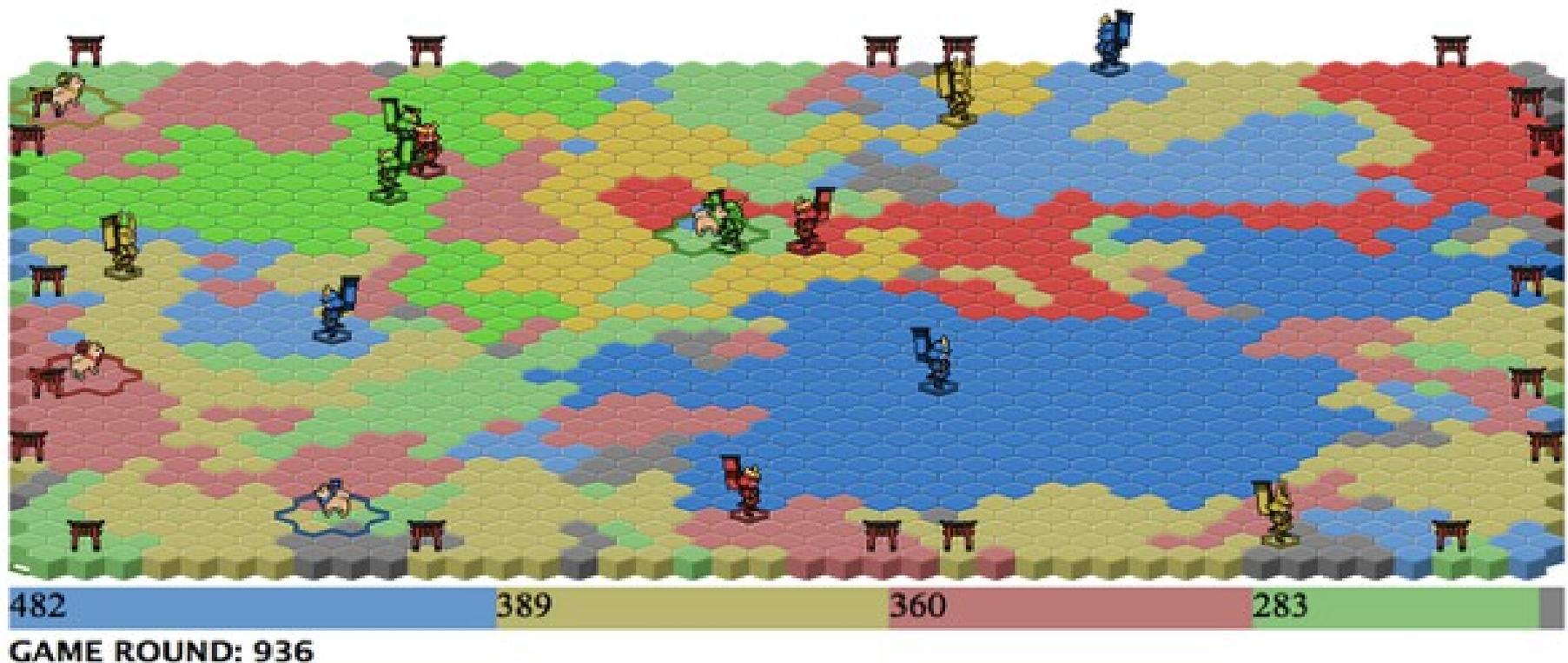
並木 彰浩 荒金 誠二
AIプログラム大会向けのコード作成

2 SamurAI Coding: 概要

2.1 SamurAI Codingについて

2012年度

Welcome to "SamurAI Coding" official site!



2 SamurAI Coding: 概要

2.2 使用言語と実行環境

言語

samurai 言語

実行環境

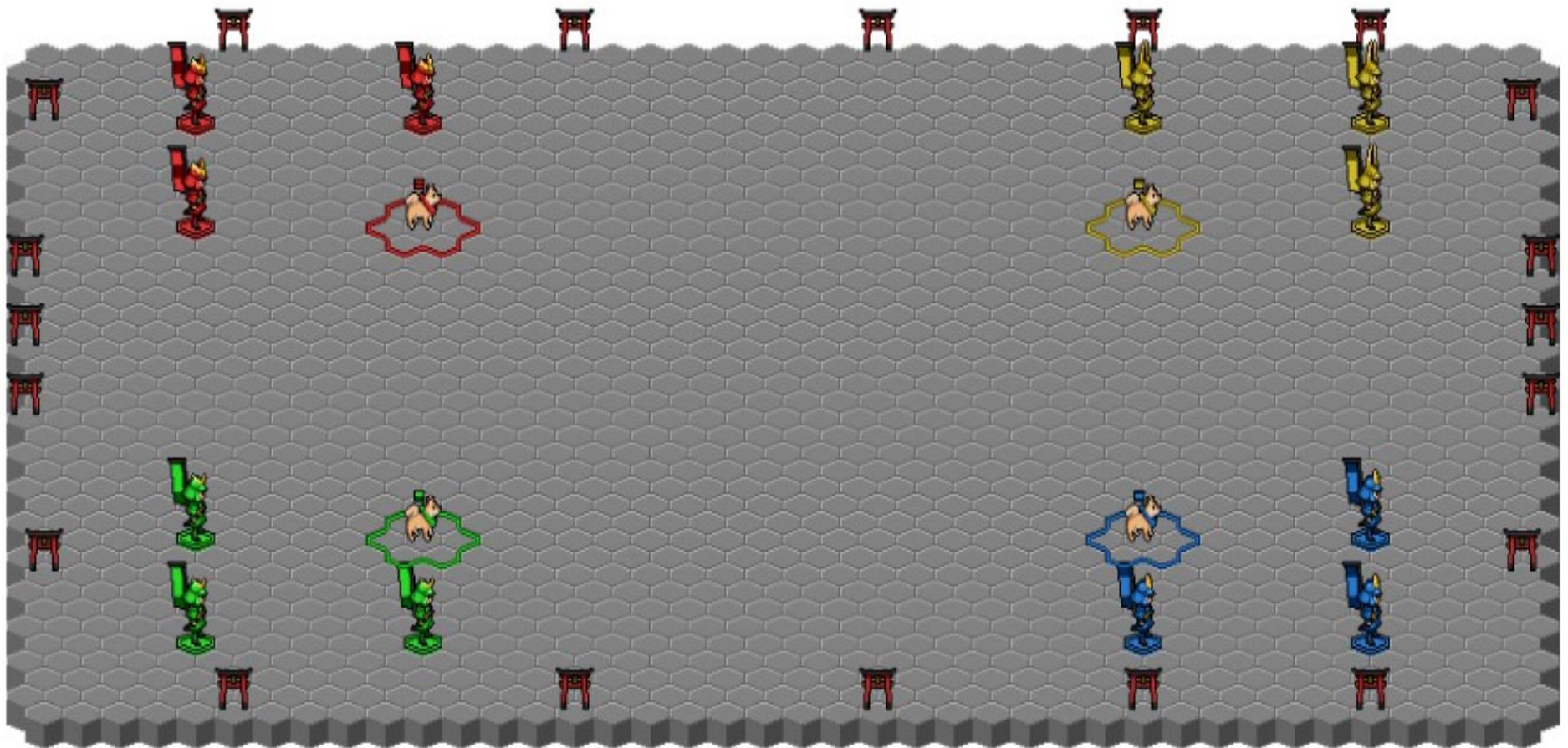
Web上で動作 (Google Chromeで動作確認)

RED TEAM	YELLOW TEAM	GREEN TEAM	BLUE TEAM
<pre>move 0, get_random_number(6) wait 1 pass 2 move 3, get_random_number(6)</pre>	<pre>i = 0 while i < 4 team = get_team_id() status = get_agent_status(team, i) if status == 1 pass i else move i, get_random_number(6)</pre>	<pre>i = 0 while i < 4 rand = get_random_number(5) if rand < 1 pass i endif if rand < 2 wait i else</pre>	<pre>i = 0 while i < 4 team = get_team_id() status = get_agent_status(team, i) if status == 1 pass i endif act = get_random_number(6)</pre>

2 SamurAI Coding: 概要

2.3 開始 (初期設定)

プレイ人数
4チーム

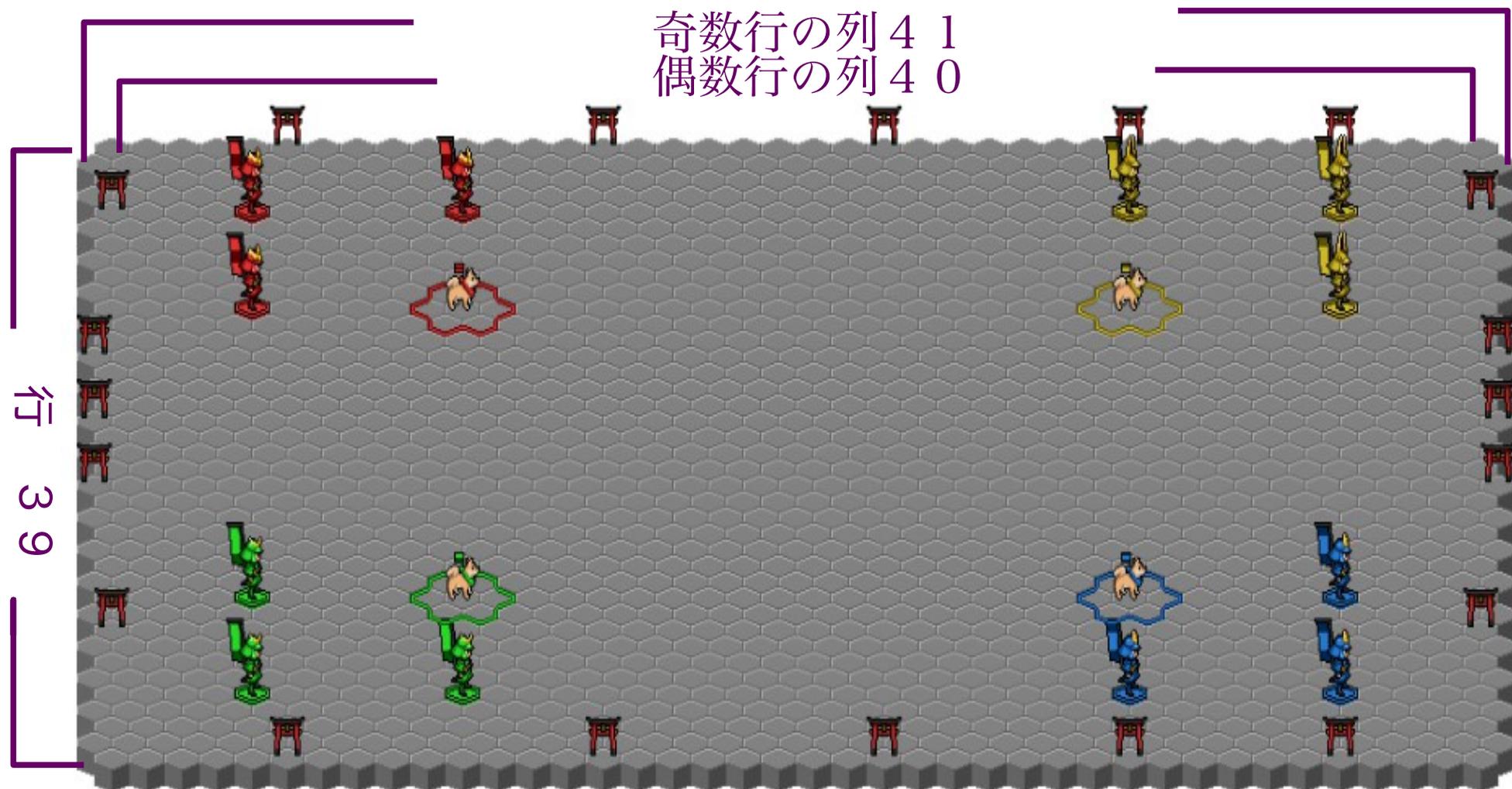


2 SamurAI Coding: 概要

2.3 開始 (初期設定)

盤面

6 角形のセルが敷き詰められている

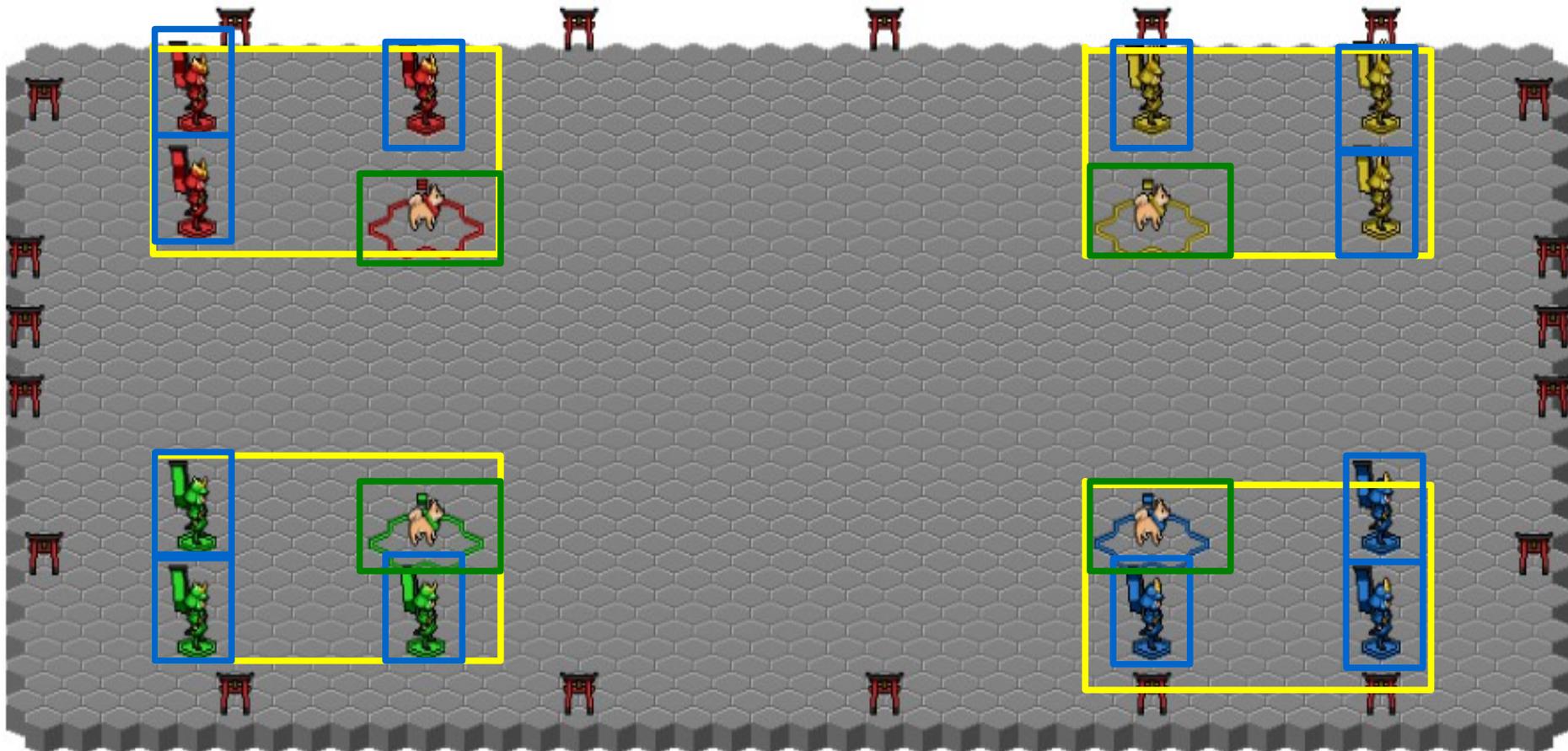


2 SamurAI Coding: 概要

2.3 開始 (初期設定)

Agent

侍 3 犬 1 が各チームに配置

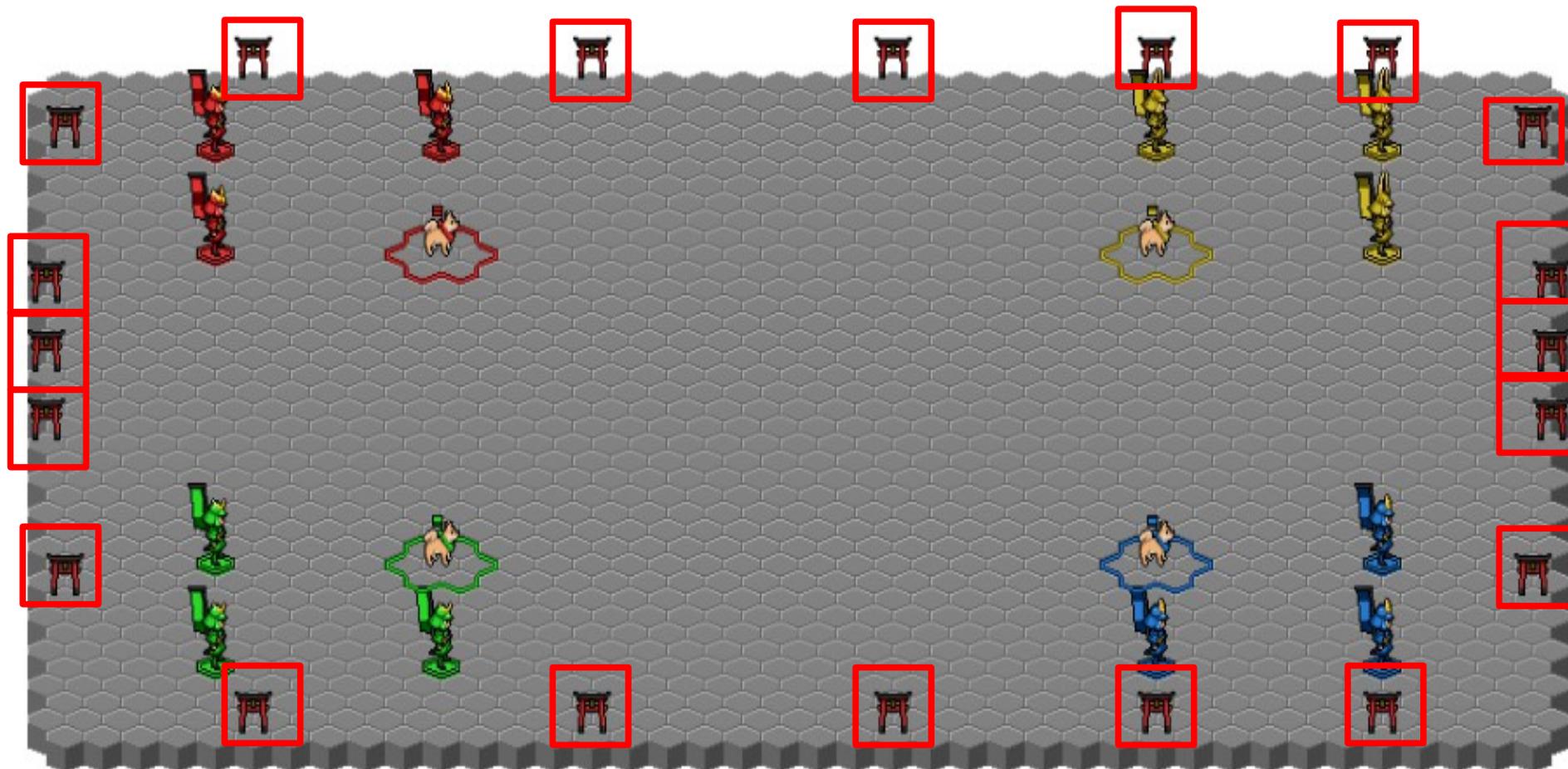


2 SamurAI Coding: 概要

2.3 開始 (初期設定)

Gate (門)

上下対称左右対称に 5 個ずつ

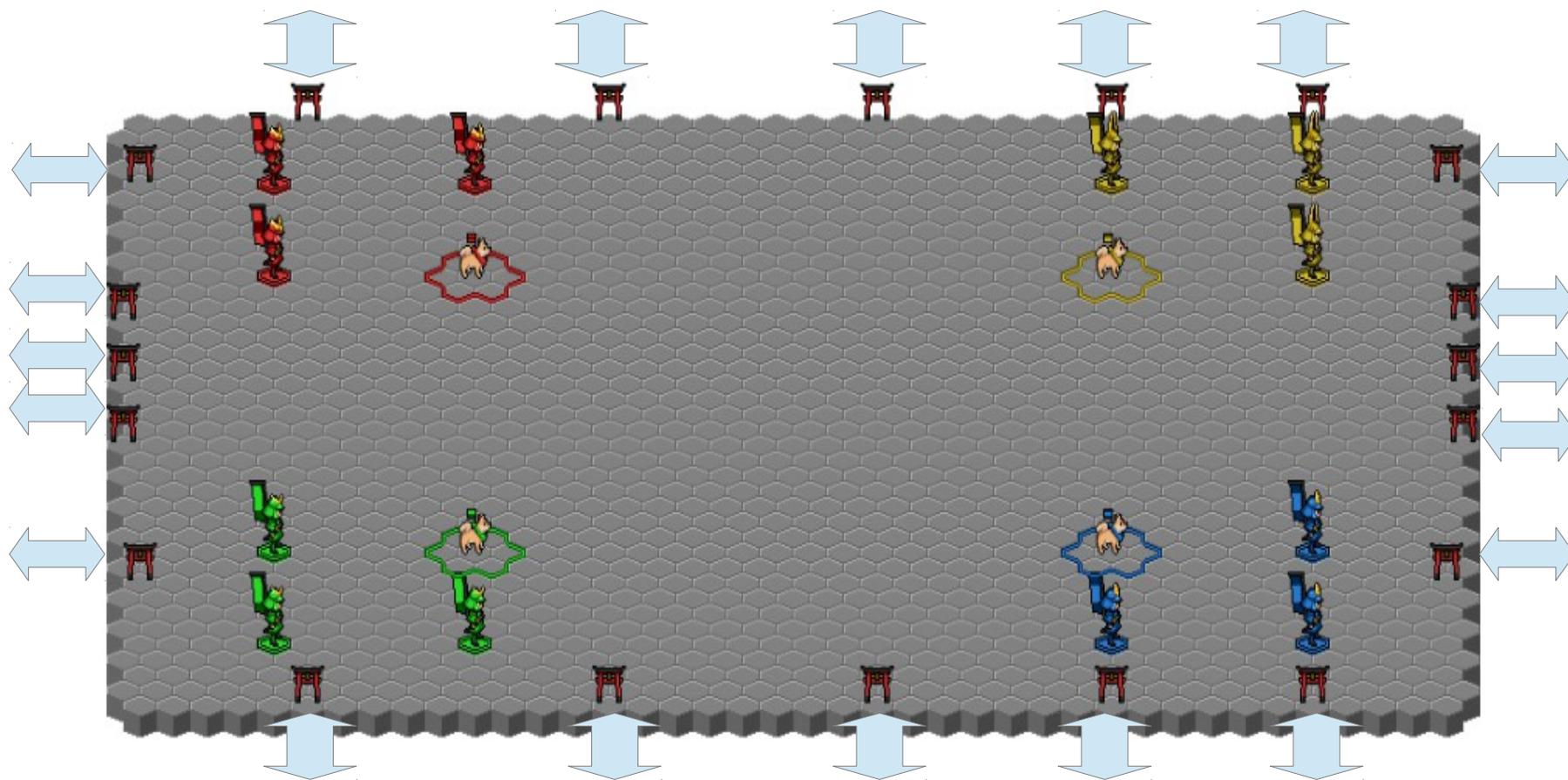


2 SamurAI Coding: 概要

2.3 開始 (初期設定)

Gate (門) の役割

対称のワープにAgentが移動



2 SamurAI Coding: 概要

2.4 進行

ラウンド

試合毎に回数変化(1000~2000)

GAME ROUND: 128/1975

GAME ROUND: 132/1537

GAME ROUND: 90/1072

佐木 彰浩 荒金 誠二
アラム大会向けのコード作成

2 SamurAI Coding: 概要

2.4 進行

ラウンド

試合毎に回数変化(1000~2000)

プログラム

ラウンド毎にソースが読まれる

GAME ROUND: 128/1975
GAME ROUND: 132/1537
GAME ROUND: 90/1072

RED TEAM

```
move 0, get_random_number(6)
wait 1
pass 2
move 3, get_random_number(6)
```

2 SamurAI Coding: 概要

2.4 進行

ラウンド

試合毎に回数変化(1000~2000)

プログラム

ラウンド毎にソースが読まれる

Agentの行動

ラウンド毎に全てのAgentは6方向に1マス

移動, または**待機**が可能

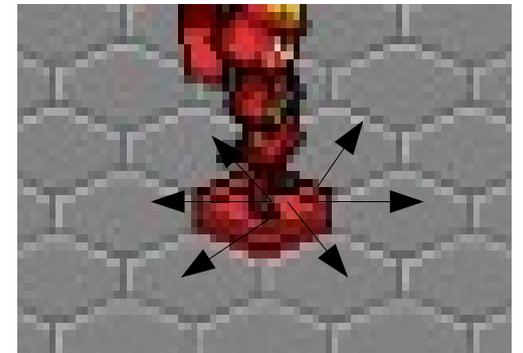
GAME ROUND: 128/1975

GAME ROUND: 132/1537

GAME ROUND: 90/1072

RED TEAM

```
move 0, get_random_number(6)
wait 1
pass 2
move 3, get_random_number(6)
```

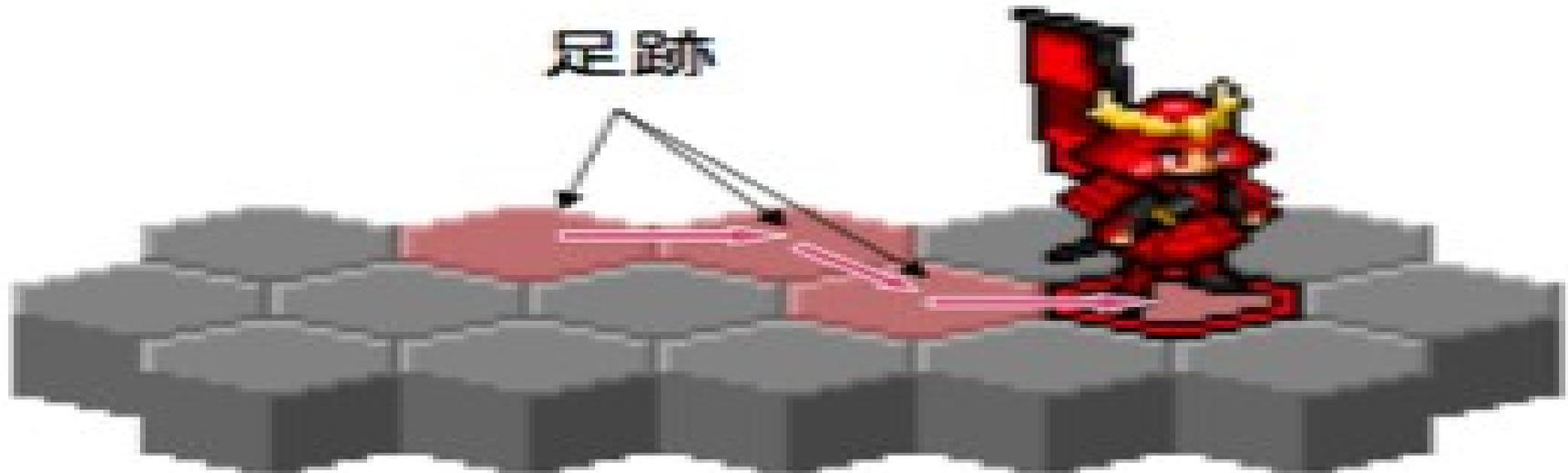


2 SamurAI Coding: 概要

2.4 進行

侍の移動(役割)

足跡を残す



2 SamurAI Coding: 概要

2.4 進行

犬の移動(役割)

【犬の脅威(青い枠内)】

侍は侵入不可 (足跡は残らない)



2 SamurAI Coding: 概要

2.4 進行

Agent（侍と犬）の待機

フリーズ状態(行動不可)を解除

発生条件

他のAgentとの衝突

壁に衝突

犬の脅威側から侍を捕縛



2 SamurAI Coding: 概要

2.5 終了(勝利条件)

大陸横断

ステージの上辺下辺，または右辺左辺に足跡を繋げた場合即座に勝利



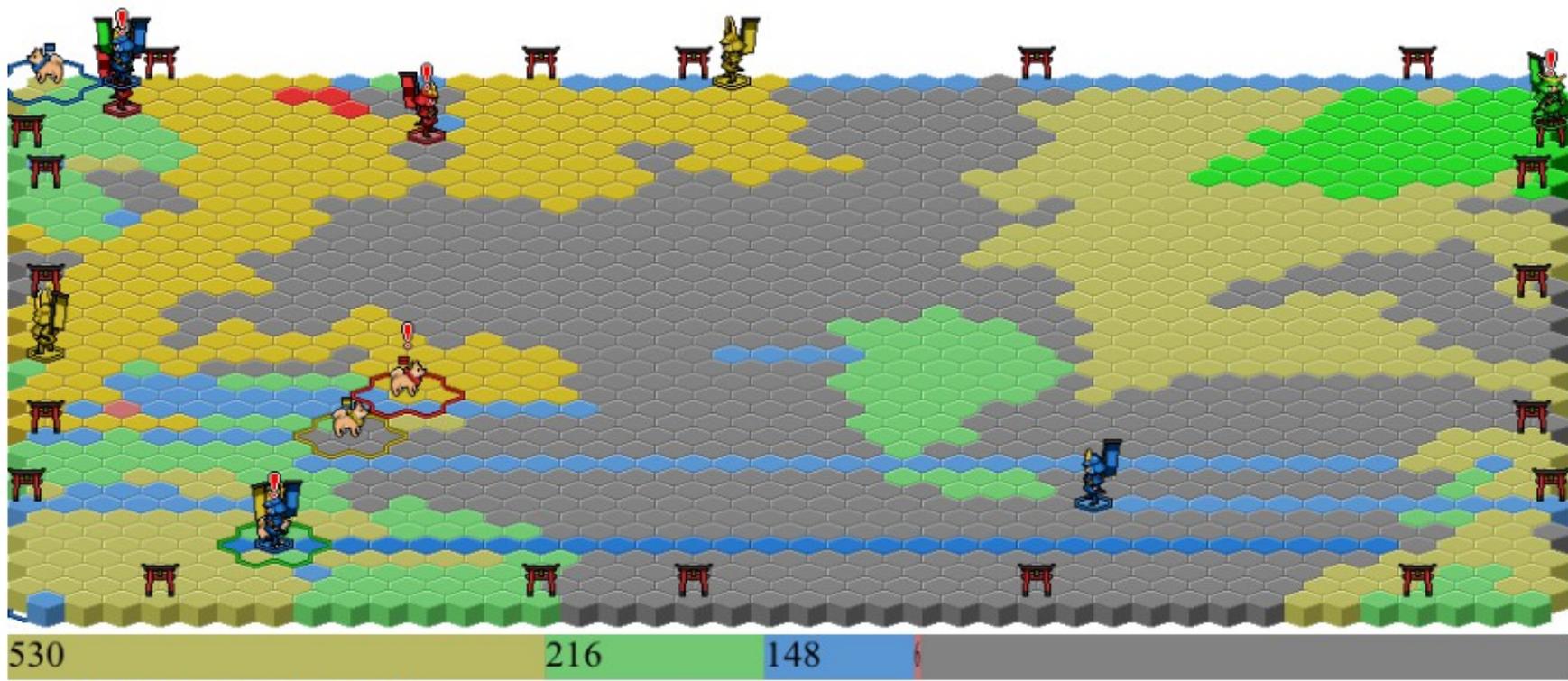
GAME ROUND: 104/1590

2 SamurAI Coding: 概要

2.5 終了(勝利条件)

大陸占領

ラウンド終了時，最も占領したチームが勝利



530

216

148

GAME ROUND: 1172/1172

2 SamurAI Coding: 概要

2.5 終了(勝利条件)

考えられる戦略

序盤で一気に
大陸横断を狙う

相手の邪魔をしつつ
自分の領地を広げて
大陸占領を狙う

2 SamurAI Coding: 概要

2.5 終了(勝利条件)

採用した戦略

序盤で一気に
大陸横断を狙う

相手の領土を少しずつ
自分の領土を広げて
大陸横断を狙う

2 SamurAI Coding: 概要

2.5 終了(勝利条件)

コスト制約

(全体で使えるコスト = 2000)

条件式 (if 文)

繰り返し文 (while 文)

変数, 配列, 定数

演算子 = == != <= >= > < + - / * % \$

関数は中身の二倍

補足 : Agent 4 人で 2000 コスト (共有)

目次

1. はじめに

2. SamurAI Coding: 概要

3. SamurAI Coding: 提出した犬Agentの動作

4. SamurAI Coding: 提出した侍Agentの動作

5. SamurAI Coding: 結果と考察

6. Mario AI Championship

3 SamurAI Coding: 提出した犬Agentの動作

3.1 犬の戦略

採用した戦略

大陸横断

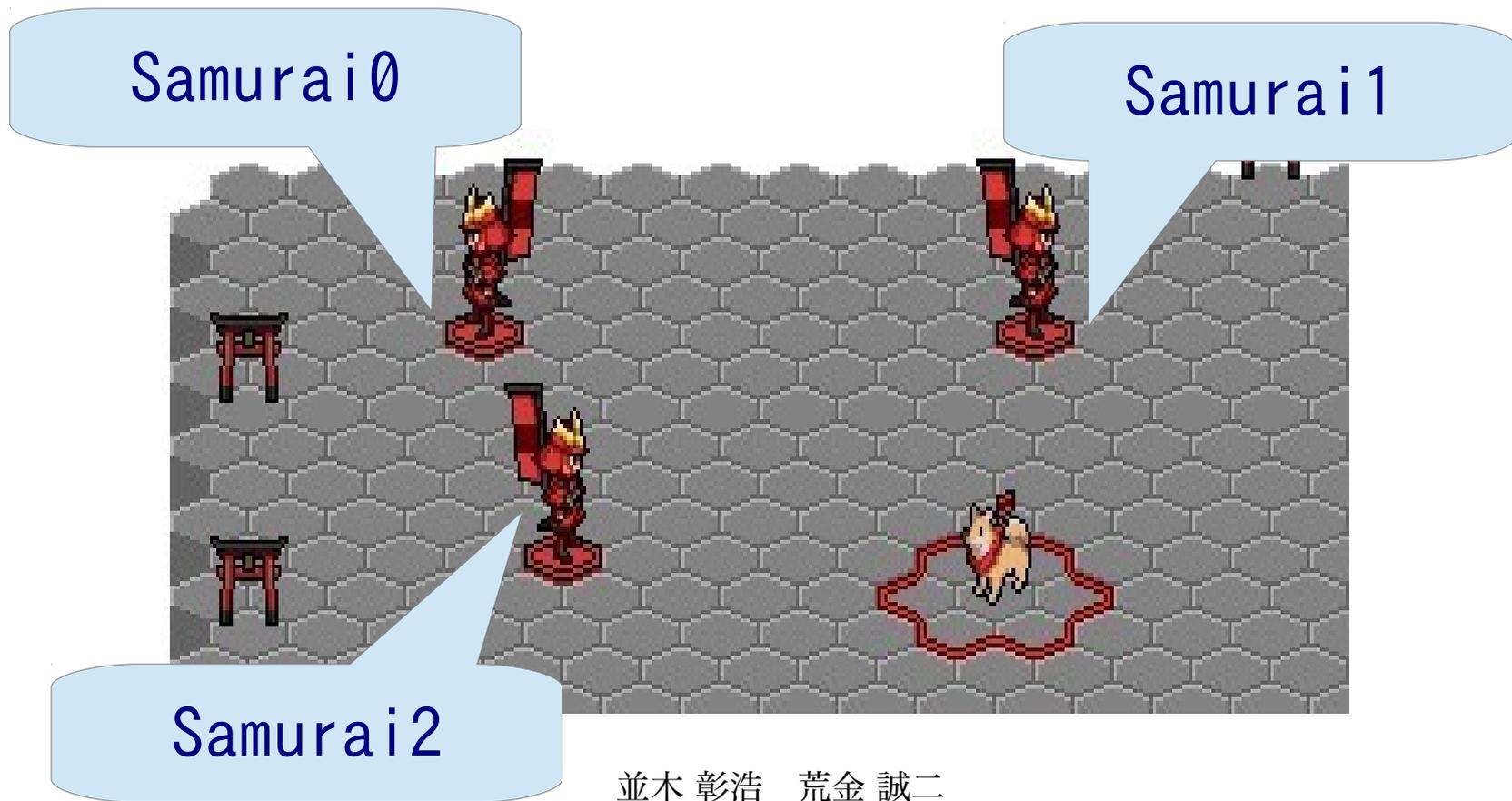


GAME ROUND: 104/1590

3 SamurAI Coding: 提出した犬Agentの動作

3.1 犬の戦略

大陸横断を狙う侍達

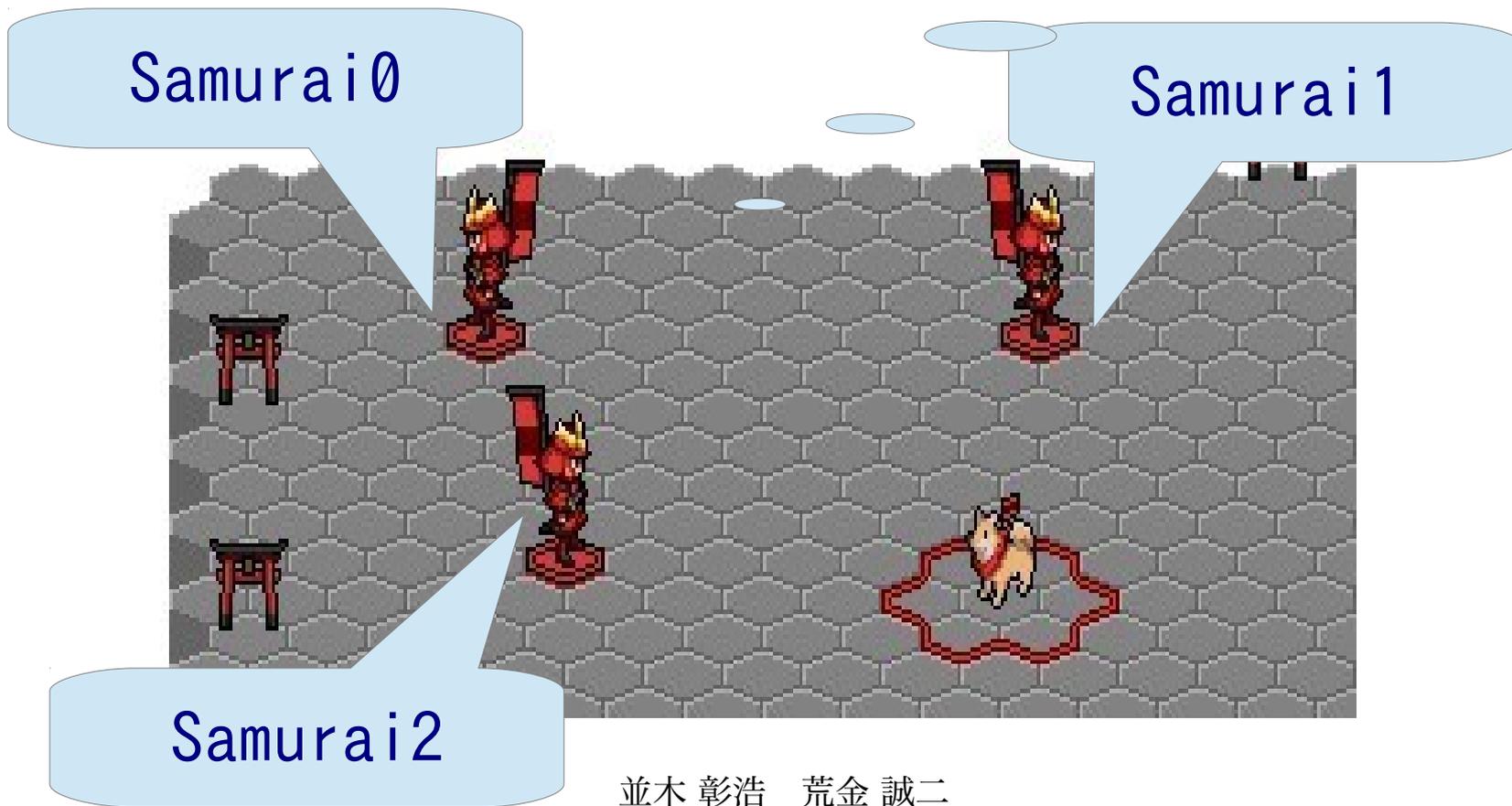


3 SamurAI Coding: 提出した犬Agentの動作

3.1 犬の戦略

大陸横断を狙う侍達

妨害されたくない!

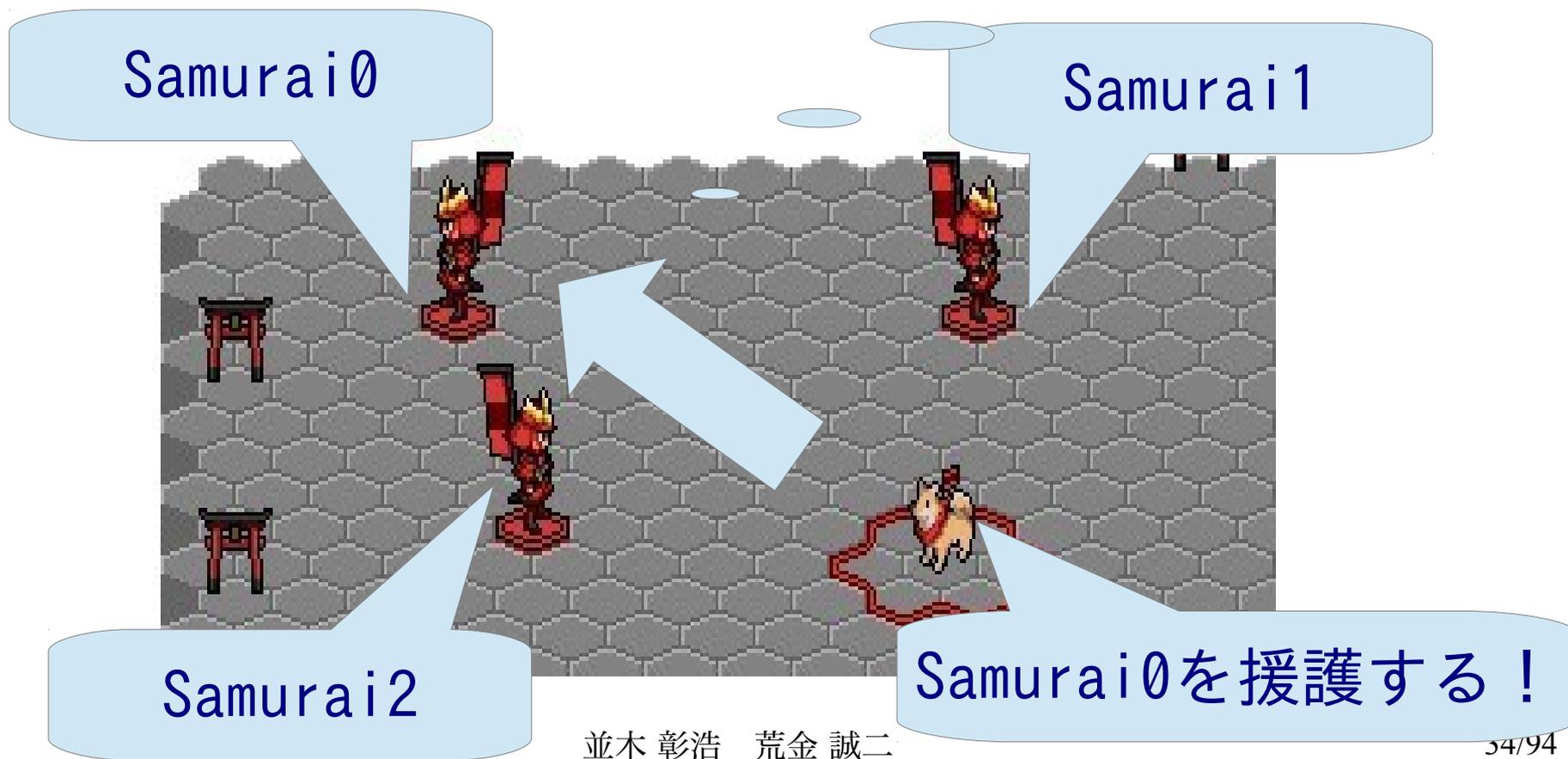


3 SamurAI Coding: 提出した犬Agentの動作

3.1 犬の戦略

Samurai0を援護する犬

妨害されたくない!



3 SamurAI Coding: 提出した犬Agentの動作

3.1 犬の戦略

Smurai0の一番近い敵のAgentに向かって突進



Target

Smurai0

3 SamurAI Coding: 提出した犬Agentの動作

3.1 犬の戦略

Target

全ての敵のAgentの中から、
Samurai0に一番近いAgent

3 SamurAI Coding: 提出した犬Agentの動作

3.1 犬の戦略

一番近いとは？

後ほどご説明！

3 SamurAI Coding: 提出した犬Agentの動作

3.2 探索

2つの探索の距離の定義

距離 1 : 最短歩数で定義

距離 2 : x座標の差とy座標の差の和で定義

3 SamurAI Coding: 提出した犬Agentの動作

3.2 探索

2つの探索の距離の定義

距離 1 : 最短歩数で定義

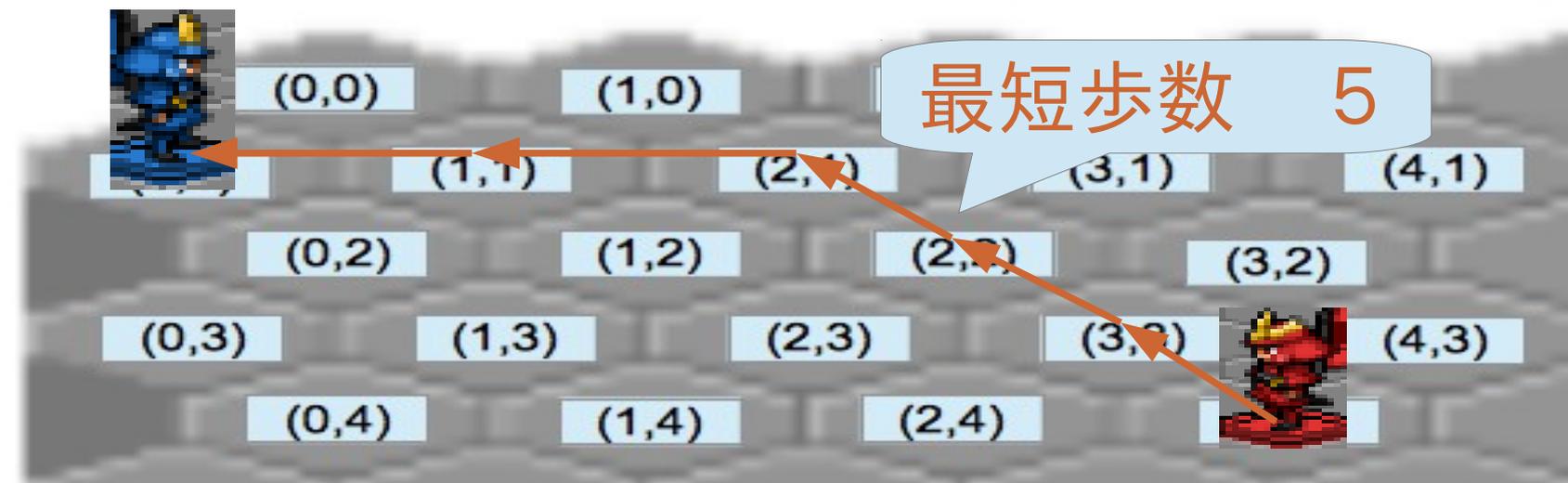
距離 2 : x座標の差とy座標の差の和で定義

3 SamurAI Coding: 提出した犬Agentの動作

3.2 探索

距離 1 : 最短歩数で定義

最短歩数

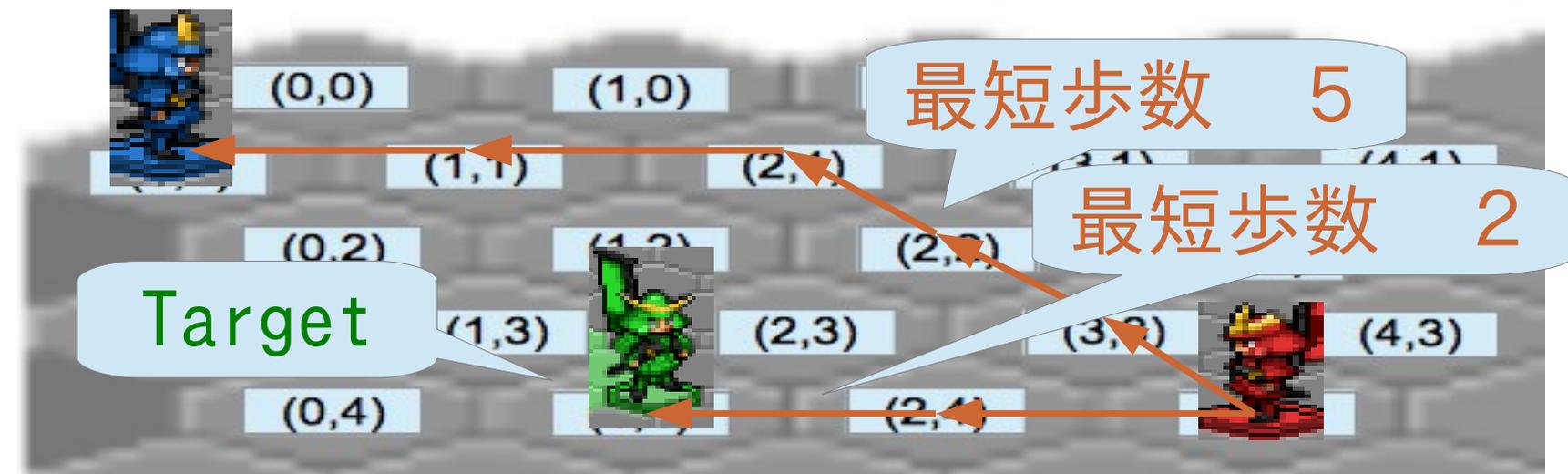


3 SamurAI Coding: 提出した犬Agentの動作

3.2 探索

距離1で一番近いとは

Samurai0と全ての敵のAgentを距離1で比較し、最小値を示すAgentをTarget



3 SamurAI Coding: 提出した犬Agentの動作

3.2 探索

距離 1 の算出

Target



Samurai 0



3 SamurAI Coding: 提出した犬Agentの動作

3.2 探索

距離 1 の算出

Target



Samurai 0



3 SamurAI Coding: 提出した犬Agentの動作

3.2 探索

距離 1 の算出

Target



列が一緒になるまで斜め探索

Samurai 0

3 SamurAI Coding: 提出した犬Agentの動作

3.2 探索

距離 1 の算出

Target



行が一緒になるまでジグザグに探索

列が一緒になるまで斜め探索

Samurai 0

3 SamurAI Coding: 提出した犬Agentの動作

3.2 探索

距離 1 の算出

Target



Samurai 0



3 SamurAI Coding: 提出した犬Agentの動作

3.2 探索

距離 1 の算出

Target



列が一緒になるまで斜め探索

Samurai 0



3 SamurAI Coding: 提出した犬Agentの動作

3.2 探索

距離 1 の算出

Target



Samurai 0



3 SamurAI Coding: 提出した犬Agentの動作

3.2 探索

距離 1 の算出

Target



行が一緒になるまで斜め探索

Samurai 0



3 SamurAI Coding: 提出した犬Agentの動作

3.2 探索

距離 1 の算出

Target



列が一緒になるまで真横に探索

行が一緒になるまで斜め探索

Samurai 0



3 SamurAI Coding: 提出した犬Agentの動作

3.2 探索

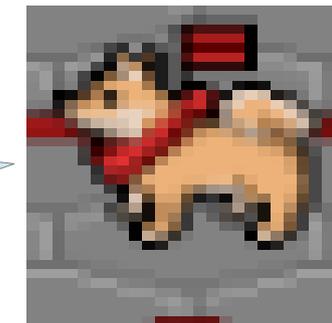
距離 1 に基づきTargetを探索



Samurai 0



Targetはだれかな？



3 SamurAI Coding: 提出した犬Agentの動作

3.2 探索

距離 1 に基づきTargetを探索



Target : なし



Samurai 0

3 SamurAI Coding: 提出した犬Agentの動作

3.2 探索

距離 1 に基づきTargetを探索



3 SamurAI Coding: 提出した犬Agentの動作

3.2 探索

距離 1 に基づきTargetを探索



3 SamurAI Coding: 提出した犬Agentの動作

3.2 探索

距離 1 に基づきTargetを探索



Target : Green(4)



最短距離 7



Samurai0



3 SamurAI Coding: 提出した犬Agentの動作

3.2 探索

距離 1 に基づきTargetを探索



Target : Green(4)



Samurai 0

3 SamurAI Coding: 提出した犬Agentの動作

3.2 探索

距離 1 を用いて実装した場合の利点・欠点

メリット

確実に一番近いAgent
を探索

デメリット

コストが膨大にかかる

3 SamurAI Coding: 提出した犬Agentの動作

3.2 探索

2つの探索の距離の定義

距離 1 : 最短歩数で定義

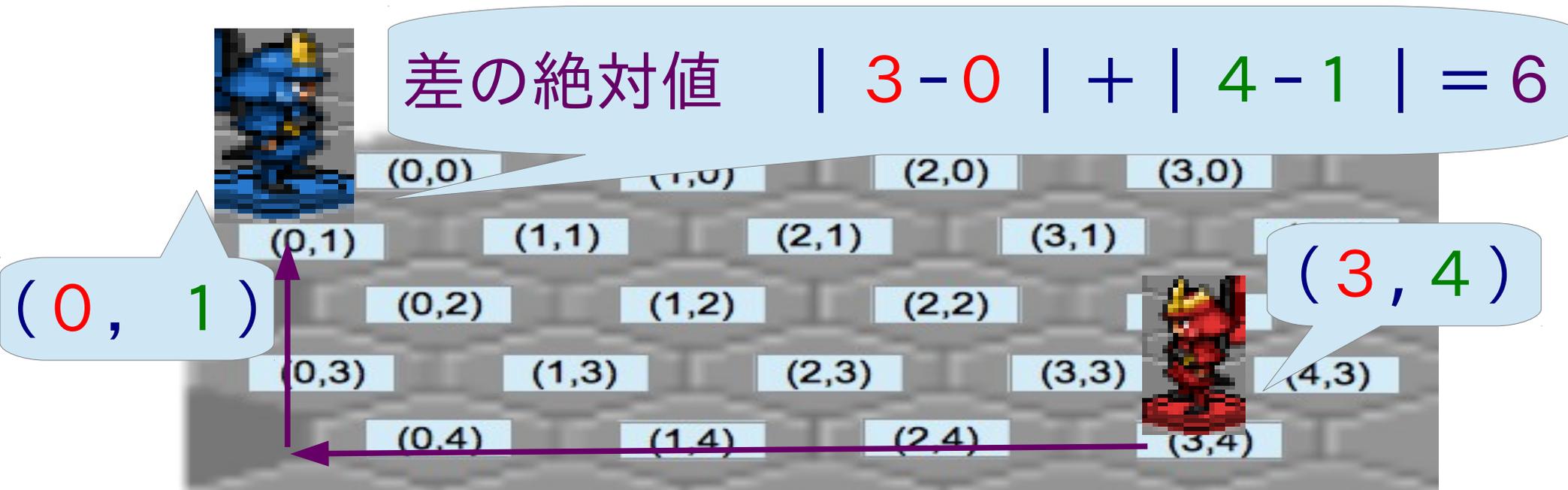
距離 2 : x座標の差とy座標の差の和で定義

3 SamurAI Coding: 提出した犬Agentの動作

3.2 探索

距離 2: x座標の差とy座標の差の和で定義

TargetとSamurai0のx, y座標をそれぞれ
差の絶対値

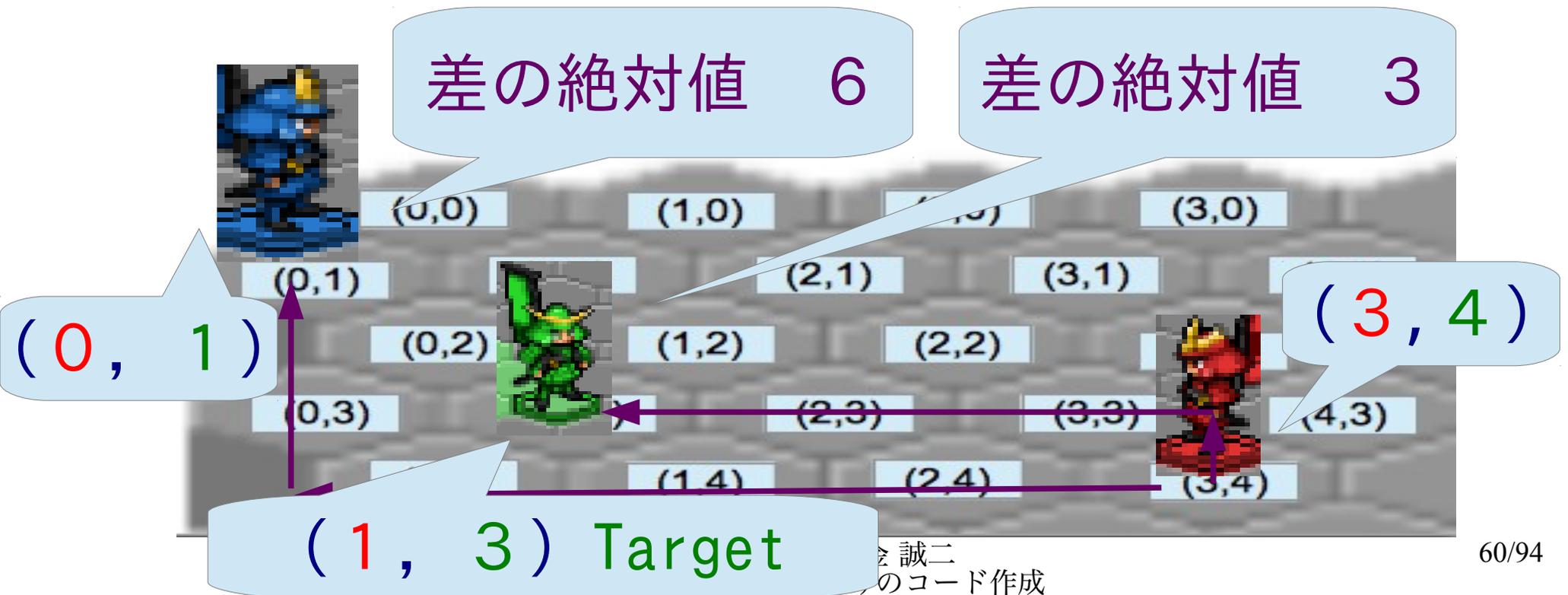


3 SamurAI Coding: 提出した犬Agentの動作

3.2 探索

距離2で一番近いとは

Samurai0と全ての敵のAgentを距離2で比較し、
最小値を示すAgentをTarget



3 SamurAI Coding: 提出した犬Agentの動作

3.2 探索

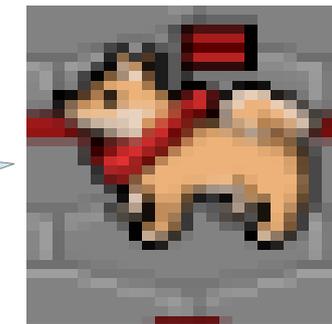
距離 2 でTargetを探索



Samurai 0



Targetはだれかな？



3 SamurAI Coding: 提出した犬Agentの動作

3.2 探索

距離 2 でTargetを探索

Target : なし



3 SamurAI Coding: 提出した犬Agentの動作

3.2 探索

距離2でTargetを探索



3 SamurAI Coding: 提出した犬Agentの動作

3.2 探索

距離2でTargetを探索



3 SamurAI Coding: 提出した犬Agentの動作

3.2 探索

距離 2 でTargetを探索



3 SamurAI Coding: 提出した犬Agentの動作

3.2 探索

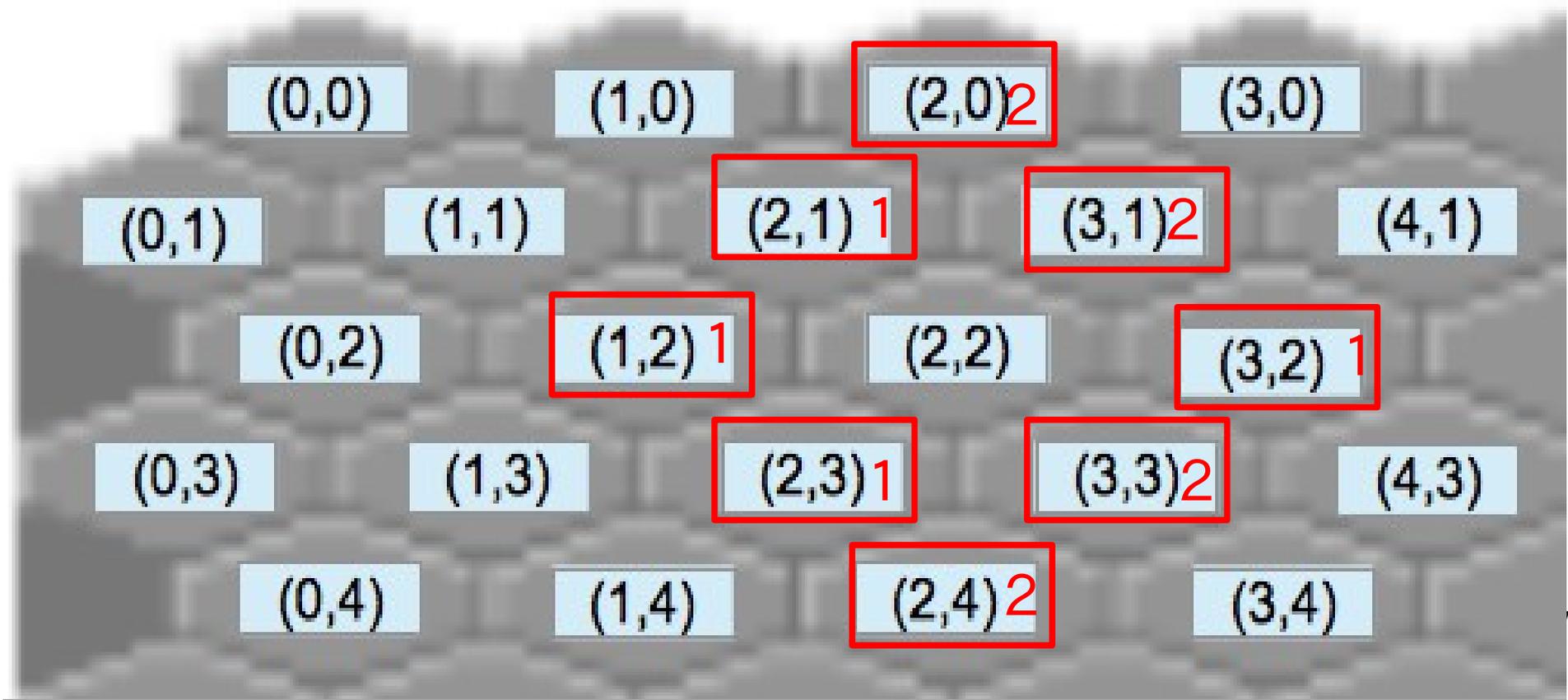
距離2でTargetを探索



3 SamurAI Coding: 提出した犬Agentの動作

3.2 探索

距離 2 と距離 1 の差

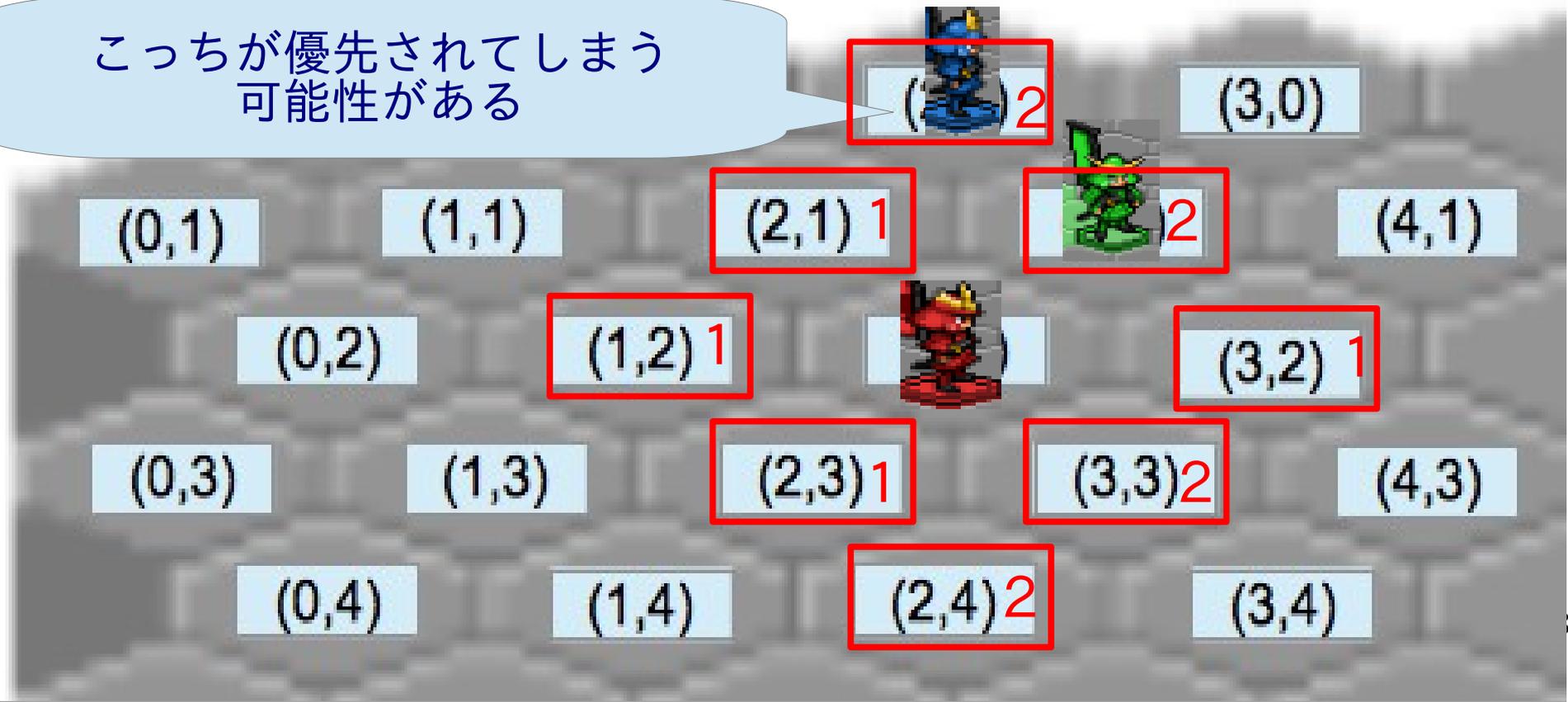


3 SamurAI Coding: 提出した犬Agentの動作

3.2 探索

距離 2 と距離 1 の差

こっちが優先されてしまう
可能性がある



3 SamurAI Coding: 提出した犬Agentの動作

3.2 探索

距離 2 を用いて実装した場合の利点・欠点



3 SamurAI Coding: 提出した犬Agentの動作

3.3 検証

距離 1 で探索と距離 2 で探索とのコスト差

【1Agent分】

距離 2 で探索 < 距離 1 で探索: 30コスト

【全体のAgent分】

(16) - 自軍(4) = 12

30 * 12 = 360

3 SamurAI Coding: 提出した犬Agentの動作 3.3 検証

犬の追跡プログラムのコストを足した場合の平均

【犬の追跡プログラムのコスト】

100 コスト

【距離1で探索の場合】

100 + 760 = 860 コスト

【距離2で探索の場合】

100 + 400 = 500 コスト

目次

1. はじめに

2. SamurAI Coding: 概要

3. SamurAI Coding: 提出した犬Agentの動作

4. SamurAI Coding: 提出した侍Agentの動作

5. SamurAI Coding: 結果と考察

6. Mario AI Championship

4 SamurAI Coding: 結果と考察

4.1 侍の戦略

考えられる戦略

序盤で一気に
大陸横断を狙うか

相手の邪魔をしつつ
自分の領地を広げて
大陸占領を狙うか

4 SamurAI Coding: 結果と考察

4.1 侍の戦略

採用した戦略

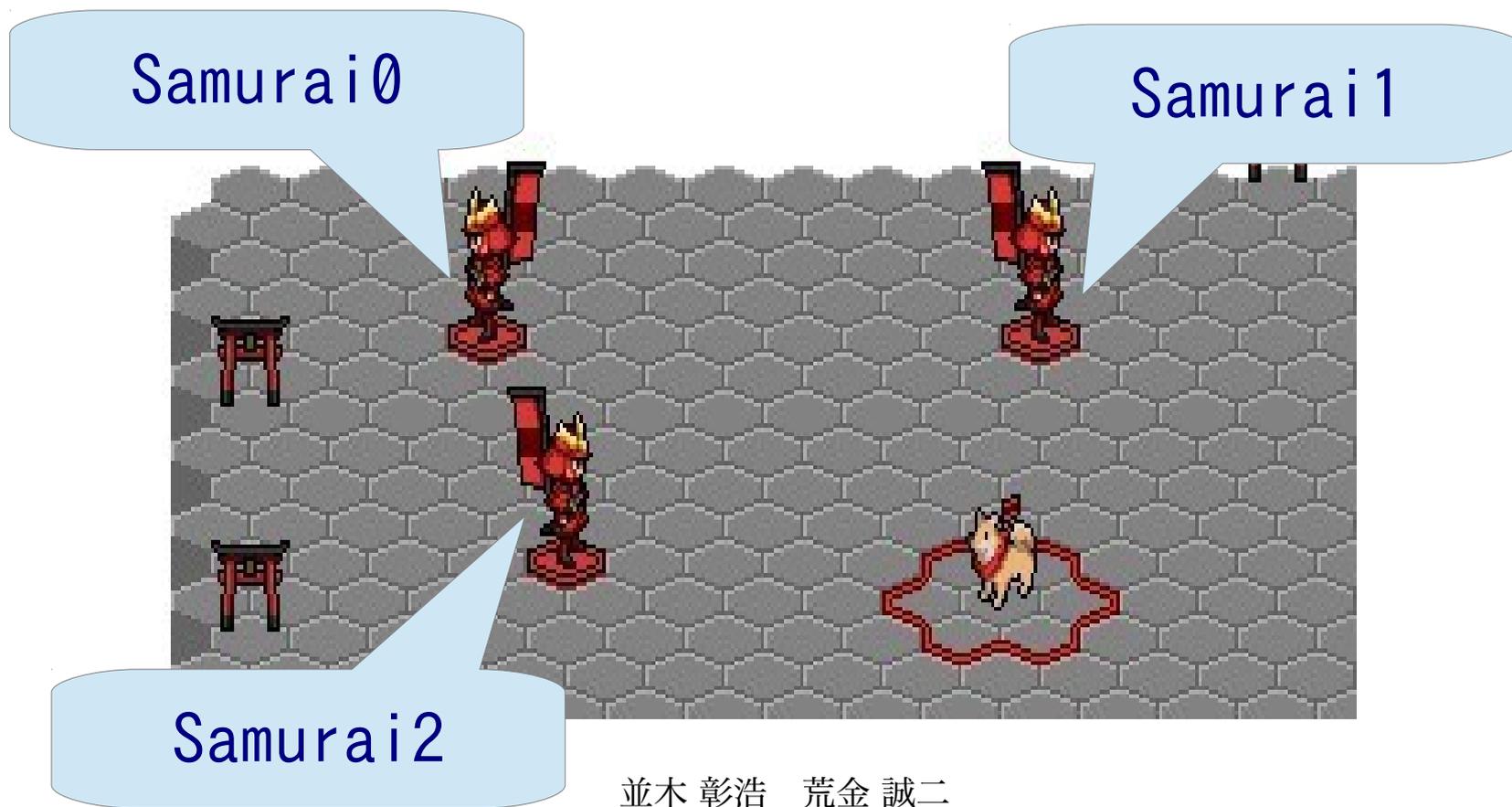
序盤で一気に
大陸横断を狙うか

相手の領地を削りつつ
自分の領地を広げて
大陸占領を狙うか

4 SamurAI Coding: 結果と考察

4.2 Samurai0とSamurai2の動作

大陸横断を狙う侍達



並木 彰浩 荒金 誠二
AIプログラム大会向けのコード作成

4 SamurAI Coding: 結果と考察

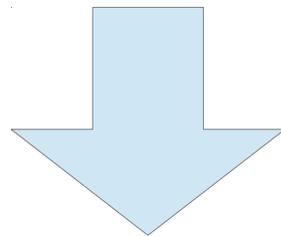
4.2 Samurai0とSamurai2の動作

Samurai0

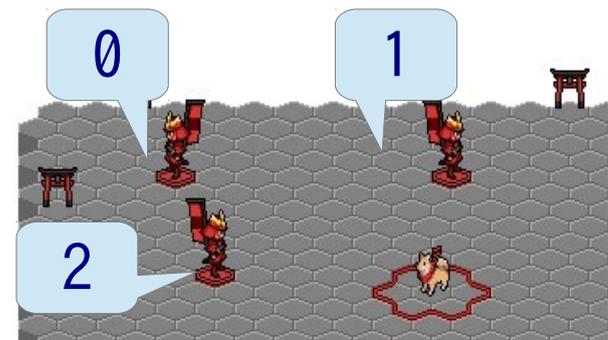
近くのゲートと同じy座標まで真横に移動し、下へ移動

Samurai2

近くのゲートと同じy座標まで真横に移動し、上へ移動



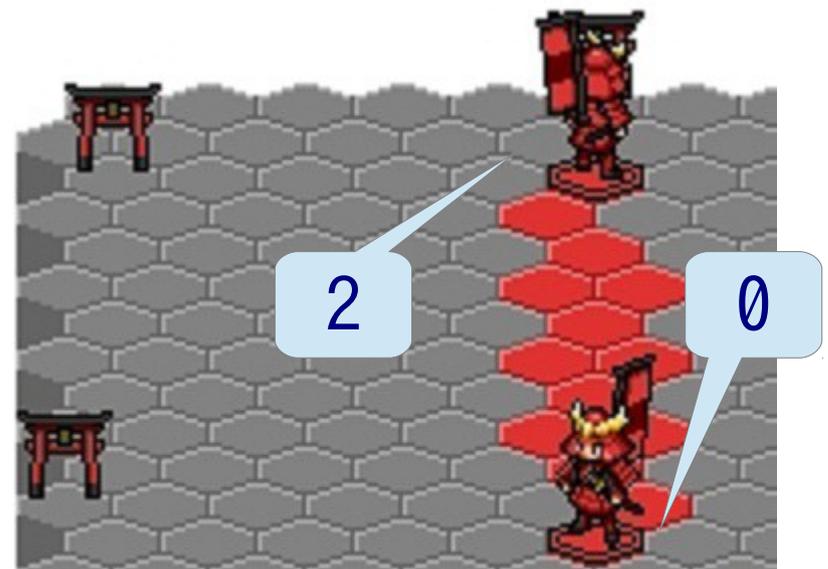
交差してラウンド数を短縮



4 SamurAI Coding: 結果と考察

4.2 Samurai0とSamurai2の動作

交差

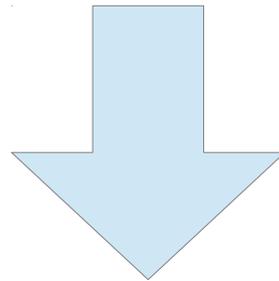


4 SamurAI Coding: 結果と考察

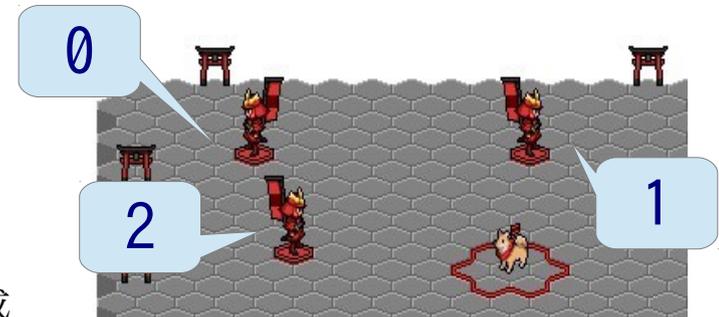
4.3 Samurai1の動作

Samurai1

近くのゲートと同じx座標まで上下に移動し、真横に移動



Samurai0, 2が失敗したときのためのリカバリー



目次

1. はじめに

2. SamurAI Coding: 概要

3. SamurAI Coding: 提出した犬Agentの動作

4. SamurAI Coding: 提出した侍Agentの動作

5. SamurAI Coding: 結果と考察

6. Mario AI Championship

5 SamurAI Coding: 結果と考察

5.1 結果

順位

残念ながら結果は20チーム中18位
決勝に行けたのは上位9チーム

5 SamurAI Coding: 結果と考察

5.1 結果

試合回数と得点の割り振り

試合回数は40回

- ◆ 1位 6点
- ◆ 2位 4点
- ◆ 3位 2点
- ◆ 4位 0点

QED++	117 point(s)
MSKF	125 point(s)
cureMarine	174 point(s)
Algori	112 point(s)
HEDSPI	116 point(s)
DDM	117 point(s)
_____	161 point(s)
Hungry	139 point(s)
UFM	127 point(s)
As_you_like	121 point(s)
BabyStars	126 point(s)
Sendy	121 point(s)
Tsujikiris	122 point(s)
Waribashi	123 point(s)
Wataru	67 point(s)
NNK@FukudaLab	119 point(s)
tosi	113 point(s)
Itoi	58 point(s)
IHACY	117 point(s)
arakawaken	125 point(s)

5 SamurAI Coding: 結果と考察

5.1 結果

侍の反省点

Samurai0の周りの特定のマスに誰かがいると、横へ1マス移動するプログラムを実装にもかかわらず味方同士で衝突

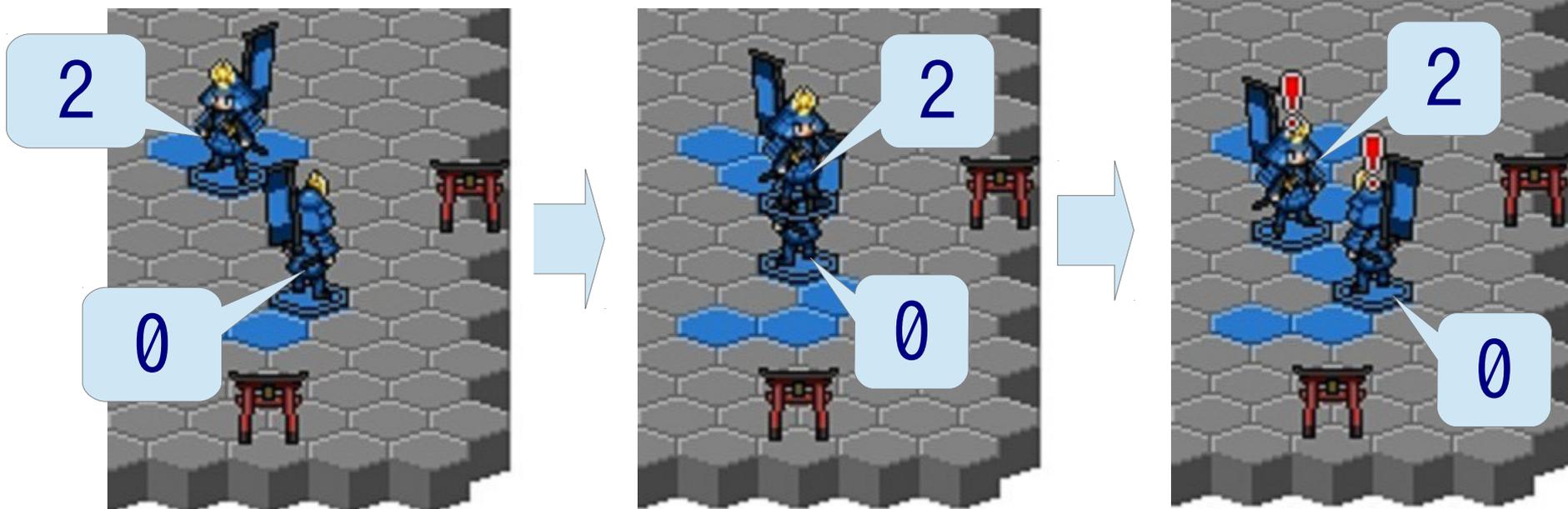
(SamurAI言語には文字数と関数コストの上限を設けられており、コストの上限に達したので下図のマス目にしか適用されなかった)



5 SamurAI Coding: 結果と考察

5.1 結果

侍の反省点



5 SamurAI Coding: 結果と考察

5.1 結果

侍の反省点

- 大会後，提出プログラムを再レビュー → バグを発見
- フィールド上部の赤，黄色チームにのみ反映
緑，青チームが衝突

5 SamurAI Coding: 結果と考察

5.2 考察

犬の反省点

- 追跡は予想通りに動作したと思われるが、座標で探索でもコストがかかりすぎてしまった（合計500～600コスト）

目次

1. はじめに

2. SamurAI Coding: 概要

3. SamurAI Coding: 提出した犬Agentの動作

4. SamurAI Coding: 提出した侍Agentの動作

5. SamurAI Coding: 結果と考察

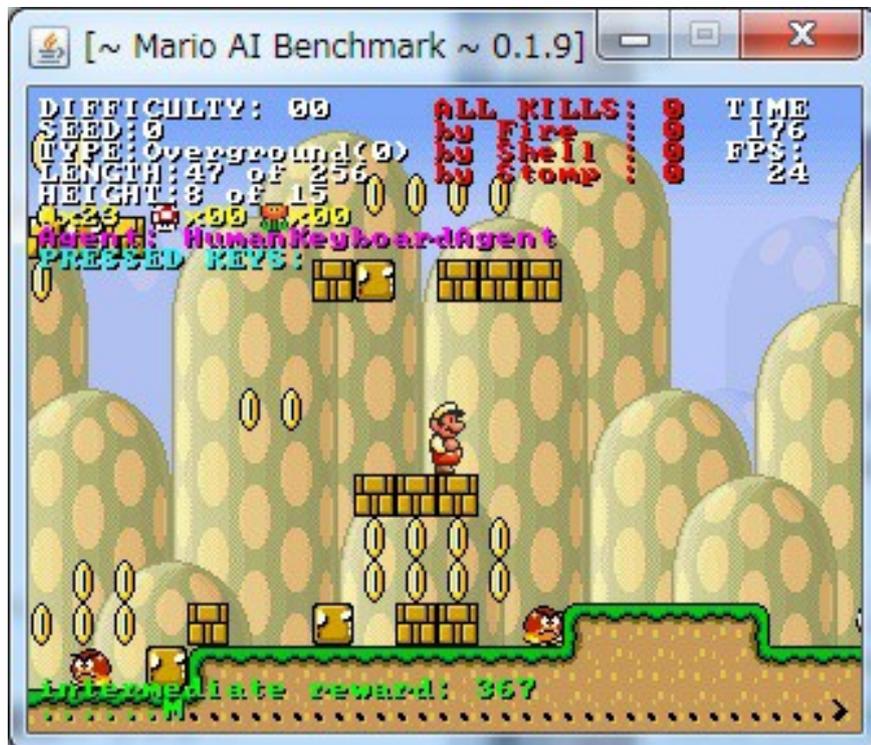
6. Mario AI Championship

6 Mario AI Championship

6.1 ゲームの概要

主催 コペンハーゲンIT大学の准教授Julian Togelius

プレイ可能パッケージ Mario AI Benchmark source package



並木 彰浩 荒金 誠二
AIプログラム大会向けのコード作成

6 Mario AI Championship

6.1 ゲームの概要

競技

- GamePlayTrack . . . 推論型
- LearningTrack . . . 学習型
- LevelGenerationTrack . . . ステージ生成
- TuringTestTrack . . . 人間味ある AI

6 Mario AI Championship

6.1 ゲームの概要

競技

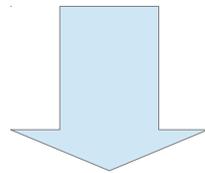
- **GamePlayTrack . . . 推論型**
- LearningTrack . . . 学習型
- LevelGenerationTrack . . . ステージ生成
- TuringTestTrack . . . 人間味ある AI

6 Mario AI Championship

6.1 ゲームの概要

ルール

- 敵を倒した数
- コインを取得した数
- ゴールまでの時間

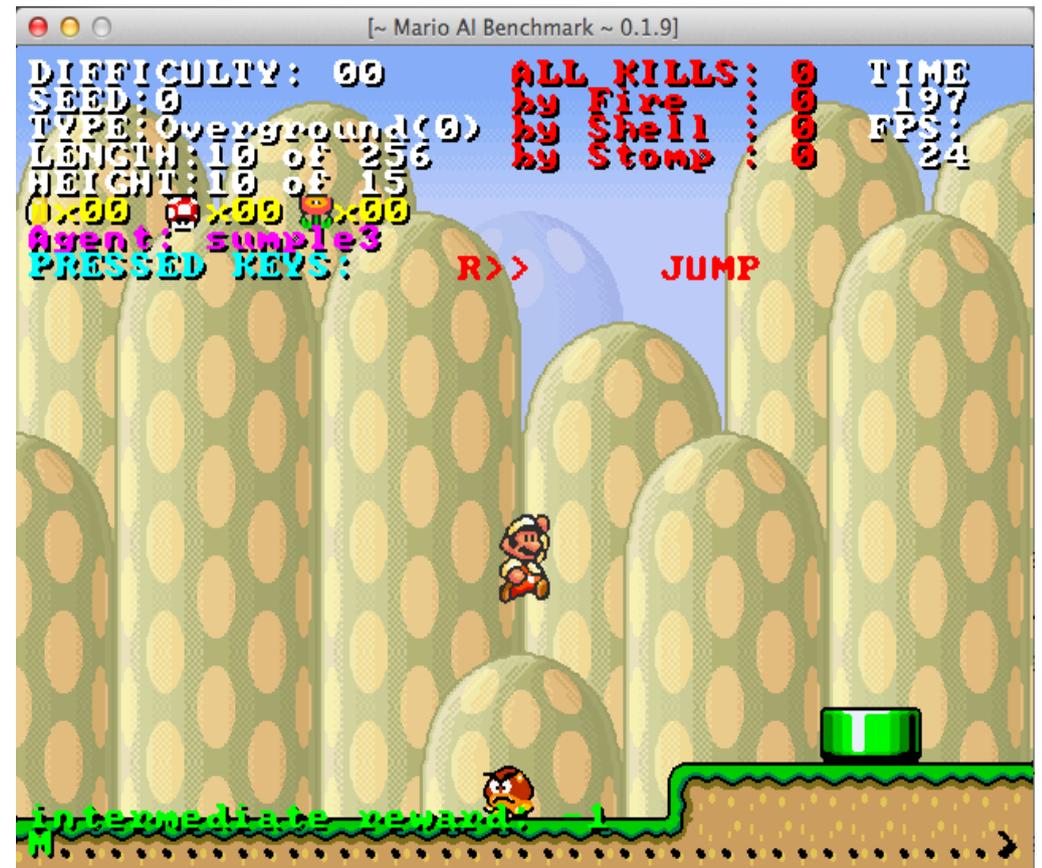
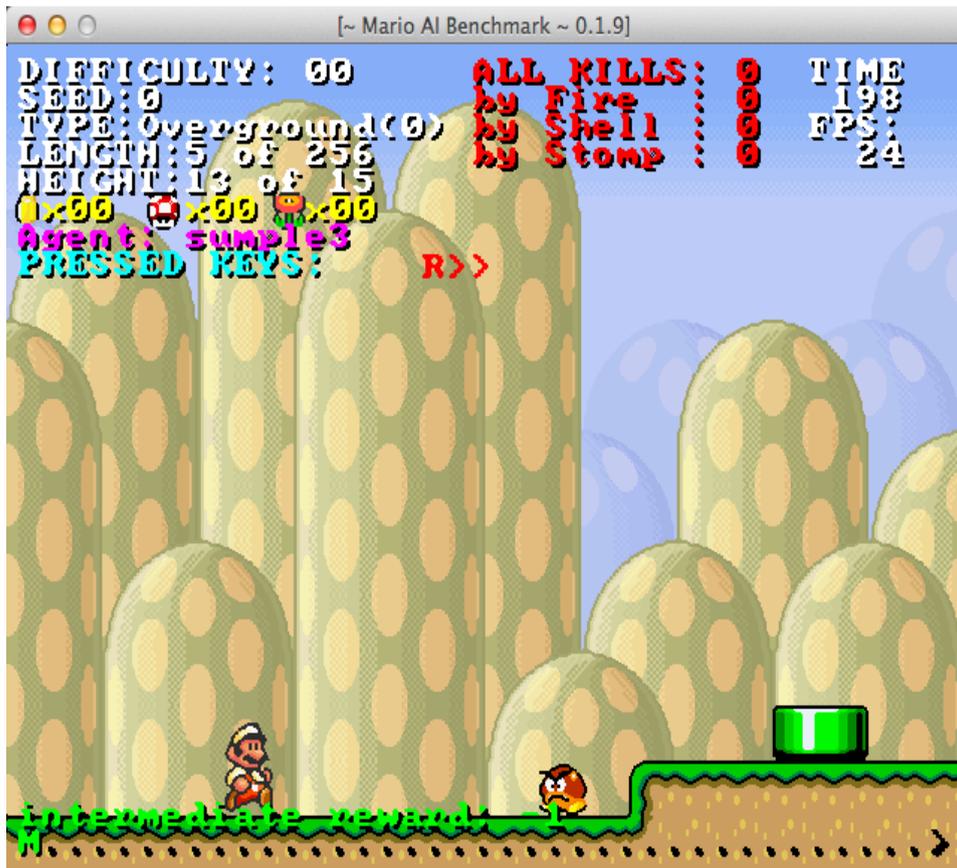


スコアを算出

6 Mario AI Championship

6.2 現状

壁と敵をジャンプで避けるプログラムを作成



6 Mario AI Championship

6.2 現状

デモ

6 Mario AI Championship

6.3 今後の課題

- 穴を越える
- 敵を避けるから倒すへ
- 目の前ではなくその先を見る
- 余裕あればコイン取得

参考サイト

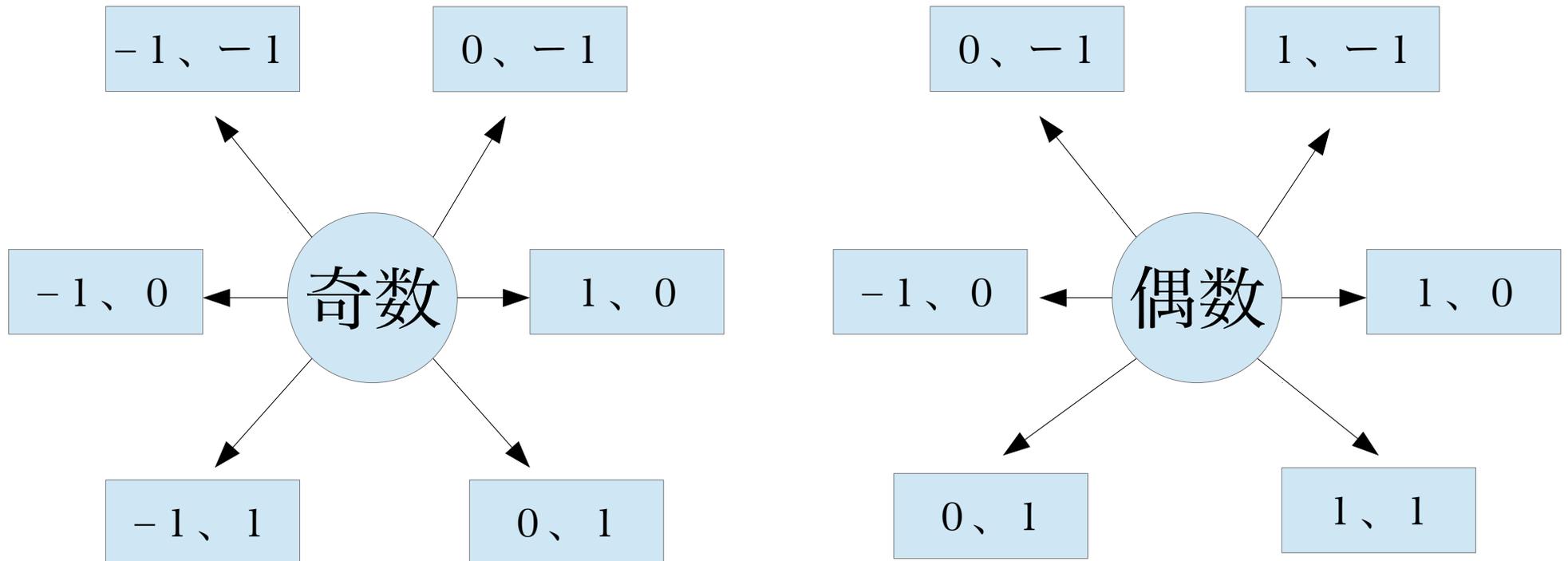
<http://www.ai-gakkai.or.jp/whatsai/>

<https://samuraicoding.org/>

<http://www.marioai.org/>

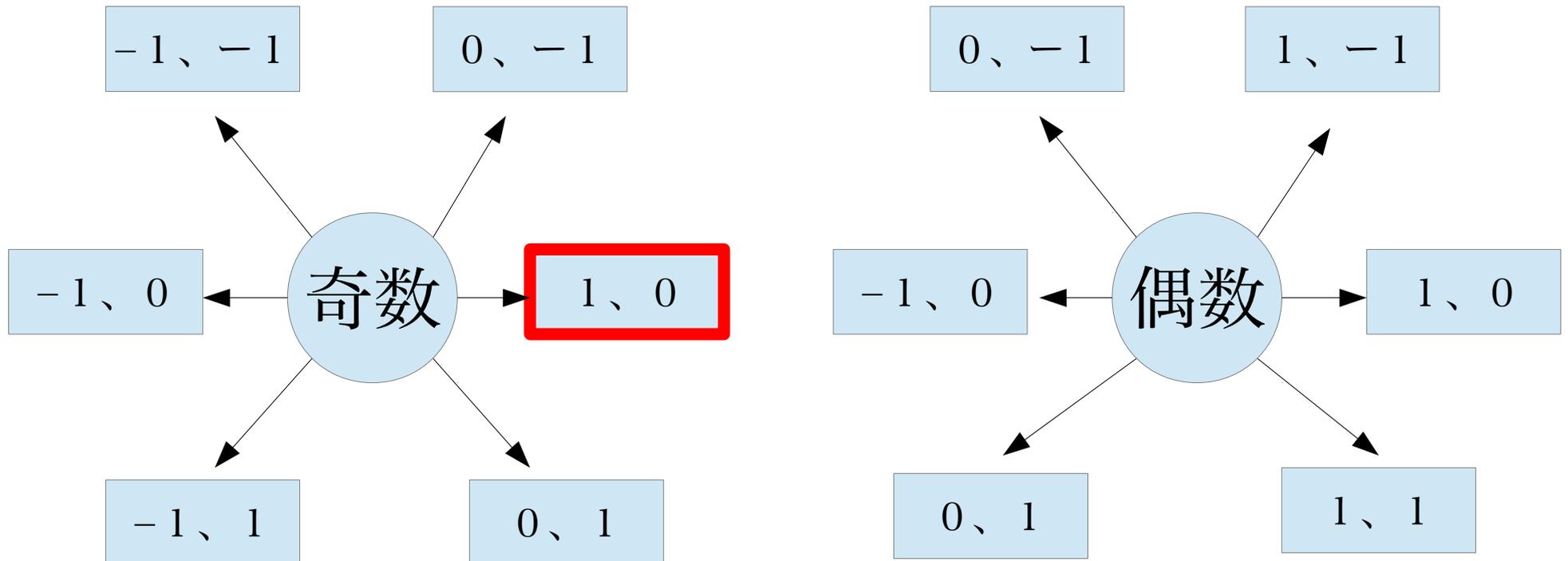
予備ページ

6角形の高さでの奇数と偶数からの一歩：



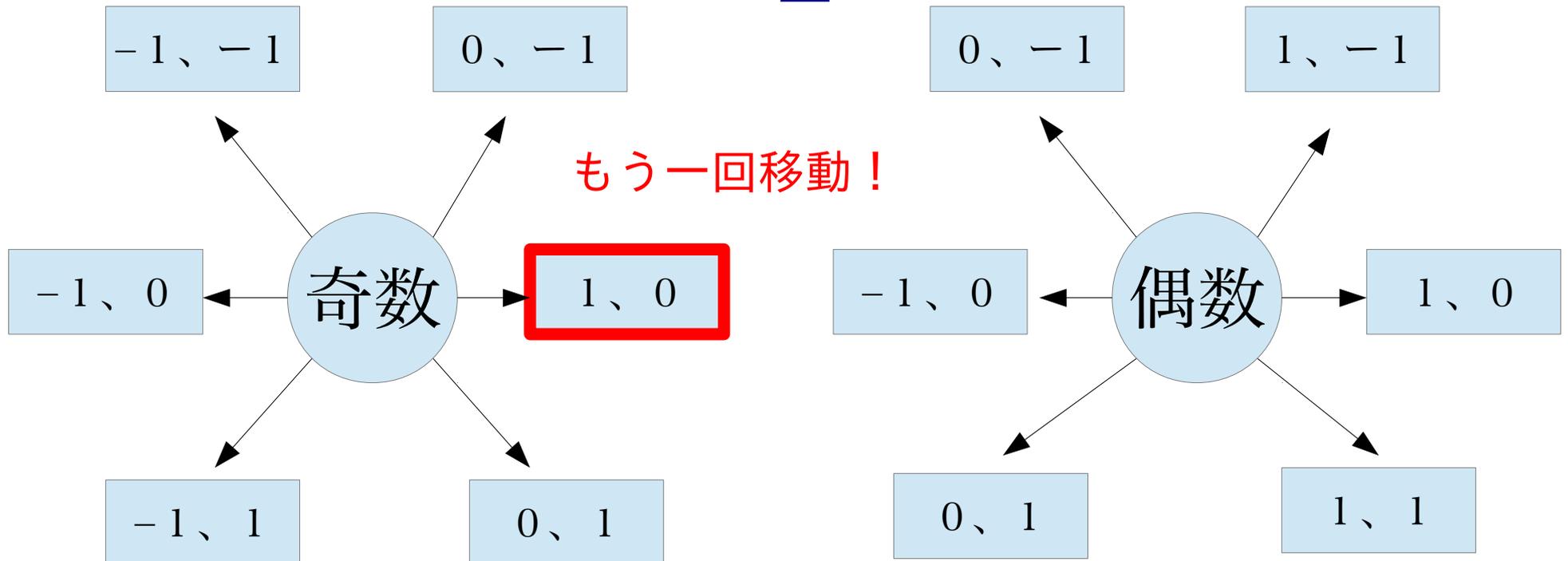
予備ページ

最短距離 2 のセルで最小のマンハッタン距離



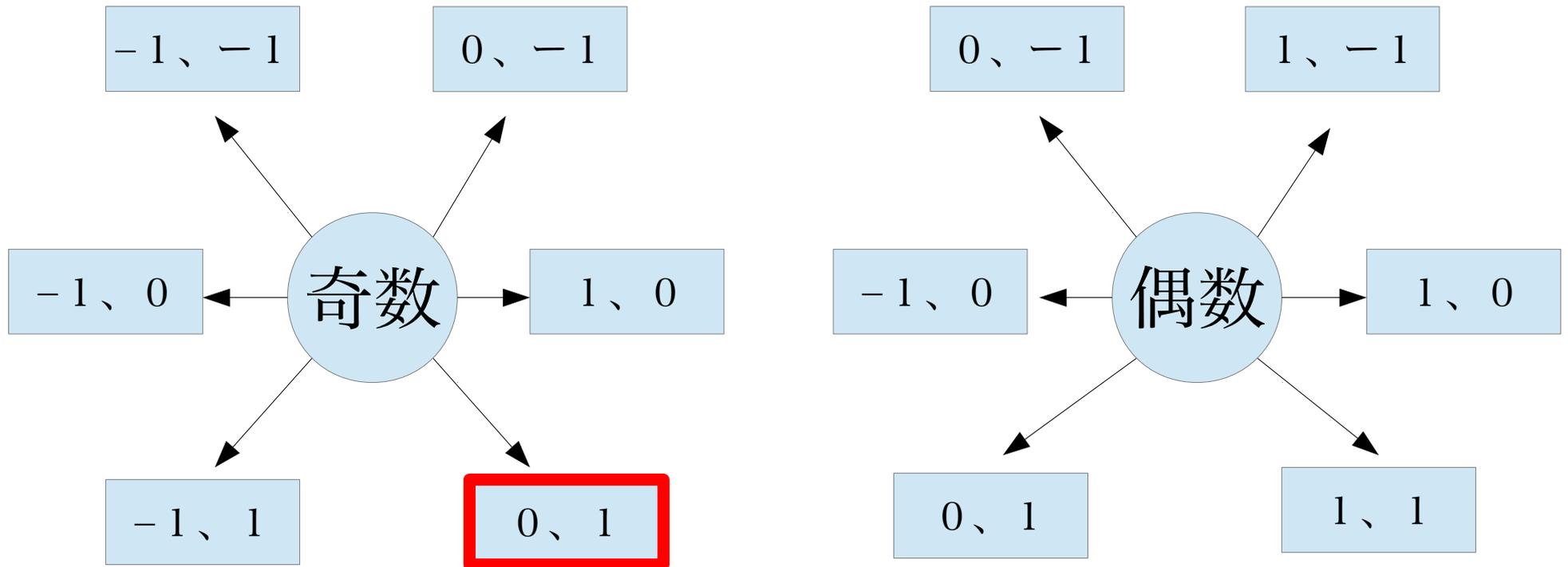
予備ページ

最短距離 2 のセルで最小のマンハッタン距離
2



予備ページ

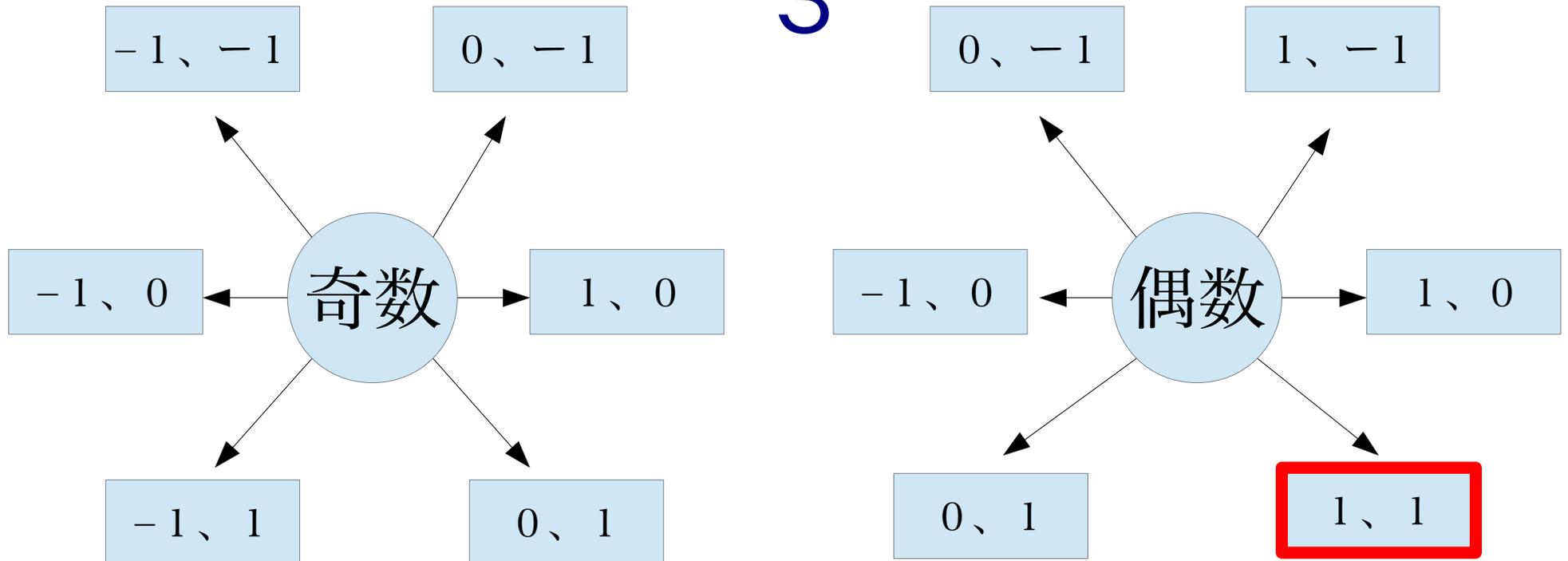
最短距離 2 のセルで最大のマンハッタン距離



予備ページ

最短距離 2 のセルで最大のマンハッタン距離

3



予備ページ

つまり

最短距離が2離れるほど, マンハッタン距離は
($3 - 2 = 1$) 1つつ離れる

最短距離 = $2n$ と仮定

マンハッタン距離の範囲 $2n \leq 2n + n$

マンハッタン距離は一緒になっても, 最短
距離は違ってくる

予備ページ

例

$$\text{最短距離}(n=6) = 12 \quad 2n < 2n + n$$

$$\text{マンハッタン数} = 12$$

$$\text{最短距離}(n=4) = 8 \quad 2n < 2n + n$$

$$\text{マンハッタン数} = 12$$

$$12 - 8 = 4 \text{ 歩の誤差}$$

予備ページ 距離の公理

X : 空でない集合

任意の $x, y \in X$ に対して、ある実数値関数 d が存在して、次を満たすと仮定

- 非負性 $d(x, y) \geq 0$
- $d(x, y) = 0 \Leftrightarrow x = y$
- 対称性 $d(x, y) = d(y, x)$
- 三角不等式 $d(x, z) \leq d(x, y) + d(y, z)$

この関数 d 距離関数, 単に距離という