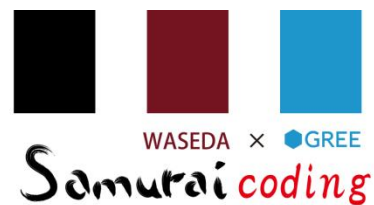


いざ、尋常に勝負。



Presented by
早稲田大学 鷺崎研究室
グリー株式会社

最終更新日：2011/12/29

目次

1. ストーリー	1
2. ゲーム概要	1
2.1 スコアの獲得方法	1
2.2 1 試合の時間	2
3. マップ	2
3.1 マップ構成	2
3.2 使用するマップ	2
4. 時間	3
5. キャラクター	3
5.1 侍	3
5.1.1 通常状態	3
5.1.2 不可視状態	3
5.1.3 将軍状態	3
5.2 犬	4
5.3 衝突	4
5.4 通常状態に戻る仕組み	4
5.5 キャラクターが壁のある方向に行こうとした場合	5
6. スコア	5
6.1 ボーナス	5
6.2 スコア表示	6
7. キャラクターの操作方法	6
7.1 キャラクターへの命令	6
7.2 標準入出力のデータ仕様	7
8. サンプル AI	9
9. 注意・その他	9
9.1 制限	9
9.2 実行環境	9
9.3 提出方法	9
9.4 WEB ランキング	9
9.5 予選・決勝方式	9
9.6 賞品	10
9.7 実行方法	10
9.8 参加確認について	10

1. ストーリー

かつて日本の侍は、日々剣術を研鑽し、そしてお互いの技を競い合った。時は流れ 21 世紀、人々は日々パソコンに打ち込む近未来の生活を手に入れた。しかし、パソコンはただの仕事道具なのだろうか？いや、違う！侍は商売道具である刀を使い、技を競い合った。それなら、我々も商売道具であるパソコンを使い、技を競い合おうじゃないか。お互いのプライドをかけて、パソコンの技を競い合う姿。まさに、SAMURAI。

2. ゲーム概要

マップに配置されたボーナスを獲得したり、他のプレイヤーからスコアを奪ったりして一定時間内に獲得したスコアを競うゲームです。プレイヤーは異なる 2 タイプの AI(侍、犬)を作成します。プレイヤーは 2 つのキャラクターを操作する事で、より高いスコアを獲得することを目指すゲームです。図 2.1 にゲーム画面を示します。

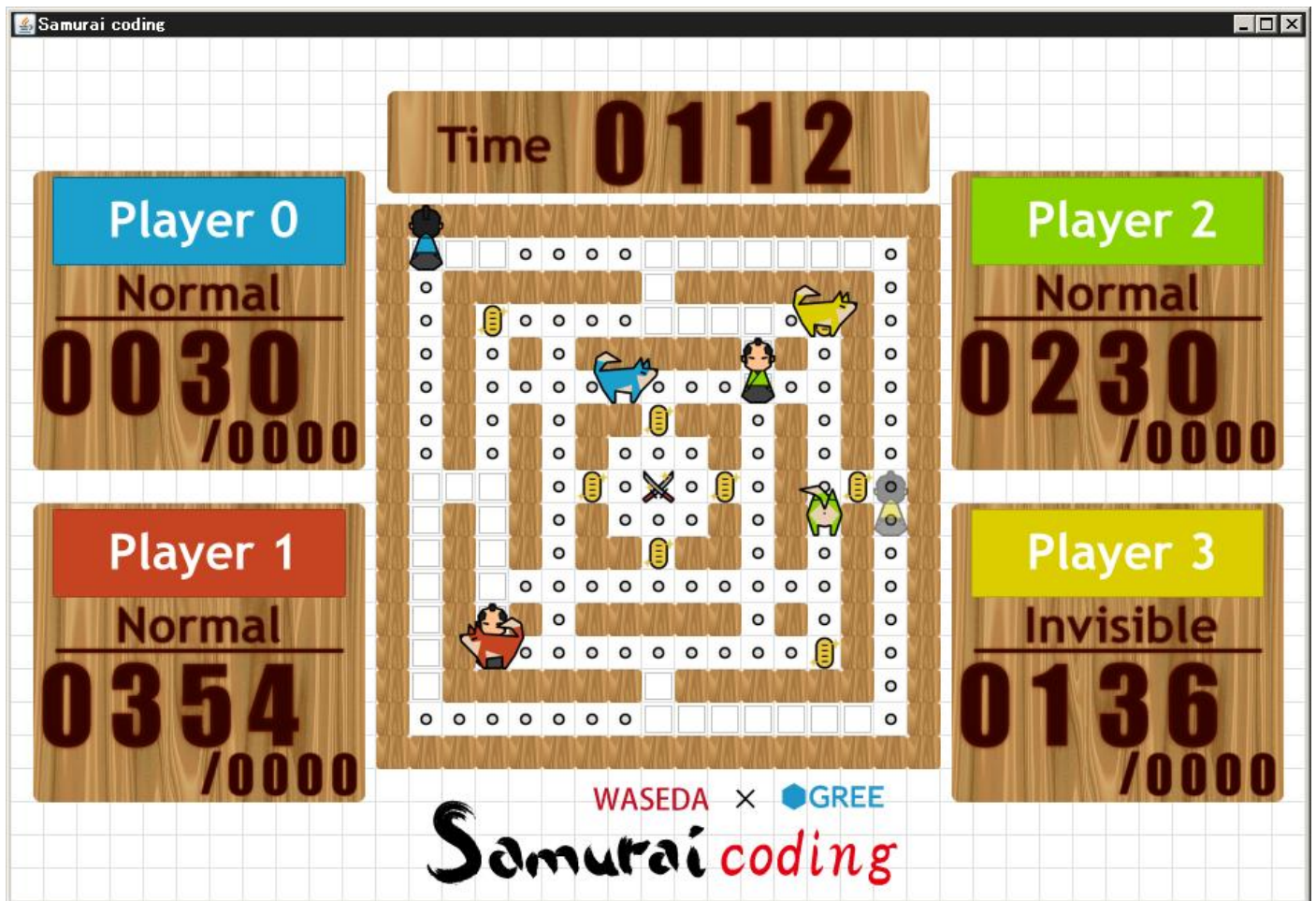


図 2.1. ゲーム画面

2.1 スコアの獲得方法

以下に示す場合が発生した時、スコアが加算されます。

1. 侍がマップに配置されたボーナスを獲得した場合。
2. 犬が敵の侍からスコアを奪い、味方の侍に届けた場合。
3. 特定のボーナスを獲得して特殊な状態(5.1.3 将軍状態を参照)になった侍で、敵犬のスコアを奪った場合。

2.2 1 試合の時間

各キャラクターが1回動く時間を1フレーム、8フレーム(8キャラクター全てが動く為にかかる時間)を1ターンと定義します。

- ・ 予選：200 ターン(1600 フレーム)
- ・ 決勝：随時公開

決勝では終了時に同点の場合は延長戦を行い、点差がつくまで1ターンずつ(8フレームずつ)続ける。また、延長戦には100 ターン(800 フレーム)の時間上限を設け、それまでに決着がつかない時は再試合とします。

3. マップ

3.1 マップ構成

マップは17×17個のマス目で構成されています。マップの周囲は壁で囲まれており、その中はタイルと壁の2種類で構成されています。各キャラクターは、壁には移動できない他、同種類のキャラクター(侍と侍、犬と犬)が既にいるマスには移動できません。なお、各キャラクターのスタート時の位置は、公平を期すため対称性を保ちます。

タイルの配置は、左上の隅の壁を原点(0,0)とし、右下の隅の壁が(16,16)となります。

キャラクターの現在地を(x, y)で表すと

- ・ 左に進む：(x-1, y)
- ・ 右に進む：(x+1, y)
- ・ 上に進む：(x, y-1)
- ・ 下に進む：(x, y+1)

のように位置座標が変化します。

3.2 使用するマップ

マップの種類は、予選は1種類、決勝大会は複数のマップで行われます(予選・決勝の詳細は 9.5 予選・決勝方式 を参照)。マップの形状は、公平を期すため対称性を保ちます。

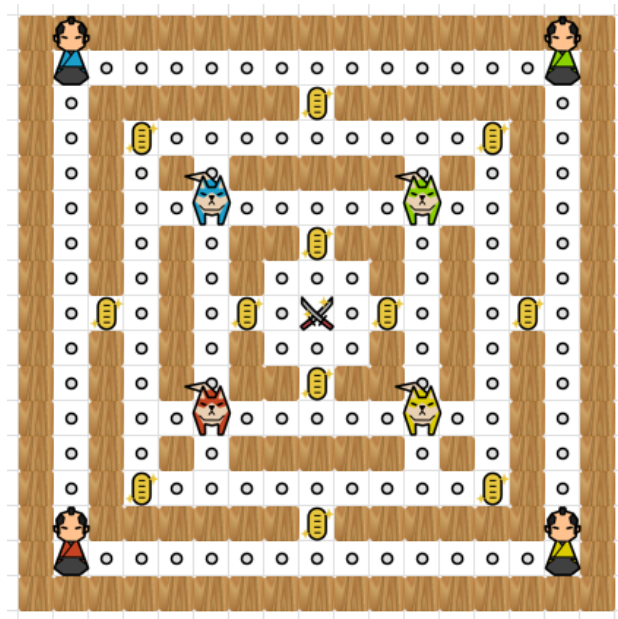


図 3.1. 初期位置の例

4. 時間

1 試合において、予選は 200 ターン(決勝大会は随時公開)で行われます。1 ターンにおいては、各キャラクターに対する 1 つの命令がそれぞれ処理されます。処理の順番は、Player 0(侍)⇒Player 0(犬)⇒Player 1(侍)⇒Player 1(犬)⇒Player 2(侍)⇒Player 2(犬)⇒Player 3(侍)⇒Player 3(犬)⇒Player 0(侍)⇒…となります。この順序はゲーム中において変化しません(いわゆるターン制です)。1 フレームにおいて、プレイヤーは各キャラクターに対して動作を 1 つ命令しなくてはなりません。(詳細は 7.1 キャラクターの操作方法 を参照)

経過時間(経過フレーム数)は図 4.1 のような形で画面の上部に表示されます。



図 4.1. 経過時間の表示

5. キャラクター

各プレイヤーには、それぞれ”侍”と”犬”の異なる役割を持つ 2 種類のキャラクターが与えられます。各プレイヤーのシンボルカラーはそれぞれ「Blue」「Red」「Green」「Yellow」です。なお、各プレイヤー間におけるキャラクターの能力に違いはありません。

5.1 侍

タイトル上のボーナスを獲得していくキャラクターです。侍が保持しているスコアがそのままプレイヤーのスコアとなります。また、侍には 3 種類の状態(通常状態、無敵状態、不可視状態)が存在します。

5.1.1 通常状態

5.1.2 および 5.1.3 で説明する不可視状態または将軍状態のどちらでもないとき、侍は通常状態となります。また、スタート時における状態も通常状態となります。

5.1.2 不可視状態

通常状態の侍が敵犬と衝突した場合、侍は不可視状態となります。不可視状態は、不可視状態になって 10 回目の自分が行動するフレームで、命令を実行した後に解除されて通常状態に戻ります。なお、不可視状態の間は敵から位置情報を取得されないほか、敵犬・味方犬と接触してもスコアの移動は発生せず、敵侍と衝突してもすれ違い処理がされます。さらにこの状態ではボーナスを取得することはできません。

通常状態の侍が敵犬にスコアを奪われた場合、初期位置にワープしますが、ワープした地点のすぐ近くに敵犬が存在する場合があります。連続してスコアを奪われないようにするためには、不可視状態であるうちに安全な場所まで非難する必要があります。

5.1.3 将軍状態

侍はパワーボーナス(6.1 ボーナスを参照)を獲得したとき、将軍状態となります。将軍状態は、将軍状態になって 30 回目の自分の行動するフレームで、命令を実行した後に解除されて通常状態に戻ります。将軍状態の間は、接触した敵犬が保持しているスコアを全て奪うことができます。**なお、侍が将軍状態の時にパワーボーナスを獲得した場合、通常状態に戻るのは新たにパワーボーナスを獲得してから 30 回目の自分の行動するフレームとなります。**



図 5.1. 侍のアイコン(左から、通常状態、不可視状態、将軍状態)

5.2 犬

敵侍を妨害するキャラクターです。敵侍(通常状態)に接触することで敵侍のスコアの 20%を奪うことができます。保持しているスコアはプレイヤーのスコアとは別に扱われ、プレイヤーのスコアとするためには、味方侍と接触してスコアを渡す必要があります。味方侍と接触すると犬が持っているすべてのスコアを味方侍に渡します。

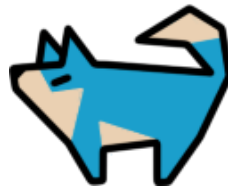


図 5.2. 犬のアイコン

5.3 衝突

違う種類のキャラクター同士が衝突(同じタイル上に存在)した場合、侍の状態によって処理が変わります。

1. 異なるプレイヤーの侍(通常状態)と犬：犬は侍が保持するスコアの 20%を奪い、侍はゲーム開始時の位置に戻り不可視状態となります。
2. 異なるプレイヤーの侍(無敵状態)と犬：侍は犬が保持するスコアを全て奪います。
3. 異なるプレイヤーの侍(不可視状態)と犬：スコアの移動は発生せずただすれ違うのみです。
4. 同じプレイヤーの侍と犬：犬が全てのスコアを侍に渡します(これにより、犬が持っていた分のスコアが正規のスコアとしてカウントされます)。
5. 同じプレイヤーの侍(不可視状態)と犬：3.と同じです。

5.4 通常状態に戻る仕組み

不可視状態および将軍状態の侍は、自分が行動するフレームで指定された回数(不可視状態：10、将軍状態：30)命令を実行した後に、そのフレーム内で通常状態に戻ります。以下に具体的な例を示します。

以降、プレイヤーNの侍を **Samurai(N)**、犬を **Dog(N)** (N=0~3)と表記します。

例 1：45 フレーム目(Dog(2)が行動するフレーム)で Samurai(0)が Dog(2)に衝突された場合

45 フレーム目で Dog(2)が Samurai(0)に衝突し、Samurai(0)は初期位置にワープし、不可視状態に変化します。その後、128 フレーム目(Samurai(0)が行動するフレーム)で Samurai(0)が命令を実行した後に通常状態に戻ります。

例 2：44 フレーム目(Samurai(2)が行動するフレーム)で Samurai(2)が Dog(1)に衝突した場合

44 フレーム目で Samurai(2)が Dog(1)に衝突し、Samurai(2)は初期位置にワープし、不可視状態に変化します。その後、120 フレーム目(Samurai(2)が行動するフレーム)で Samurai(2)が命令を実行した後に通常状態に戻ります。

例 3 : 46 フレーム目で(Samurai(3)が行動するフレーム)で Samurai(3)がパワーボーナスを獲得した場合

46 フレーム目で Samurai(3)はパワーボーナスを獲得して将軍状態に変化します。その後、286 フレーム目(Samurai(3)が行動するフレーム)で Samurai(3)が命令を実行した後に通常状態に戻ります。

例 4 : 46 フレーム目で(Samurai(3)が行動するフレーム)で Samurai(3)がパワーボーナスを獲得した後、さらに 94 フレーム目(Samurai(3)は将軍状態)でもパワーボーナスを獲得した場合

46 フレーム目で Samurai(3)はパワーボーナスを獲得して将軍状態に変化します。その後、96 フレームでも新たにパワーボーナスを獲得したので、336 フレーム目(Samurai(3)が行動するフレーム)で Samurai(3)が命令を実行した後に通常状態に戻ります。

5.5 キャラクターが壁のある方向に行こうとした場合

キャラクターが壁のある方向に行こうとしたとき、キャラクターは移動せずにその場にとどまります。

6. スコア

プレイヤーの正規のスコアとして認められるスコアは、侍が持っているスコアのみとなります。犬が持っているスコアは正規のスコアとはカウントされません。犬が持っているスコアは味方侍に渡して初めて正規のスコアとしてカウントされます。なお、本コンテストでは、正規のスコア、つまり侍が持っているスコアのみで競います。

プレイヤーのスコア = 侍が取得したボーナススコア + 犬から受け取ったスコア

6.1 ボーナス

ボーナスは以下の 3 種類が存在します。

- ・通常ボーナス(10 点)
- ・ビッグボーナス(100 点)
- ・パワーボーナス(0 点、獲得した侍は 30 ターン(240 フレーム)の間、将軍状態となります)

また、ボーナスがマップ上から全て消滅した場合、そのフレーム終了時にゲーム開始時と同じ位置に同じボーナスが復活します。なお、ボーナスを獲得できるのは侍のみで、犬は獲得することができません。



図 6.1. ボーナスのアイコン(左から、通常ボーナス、ビッグボーナス、パワーボーナス)

6.2 スコア表示

各キャラクター(侍、犬)のスコアは図 6.2 のように表示されます。



図 6.2. スコアの表示

7. キャラクターの操作方法

プレイヤーは標準入出力を介してゲームシステムとのやりとりを行います。具体的には、標準入力から現在のゲームの状況を受け取り、標準出力を用いてキャラクターに命令を出します。ゲーム状況の取得には C, C++, C#, Java, Ruby, Python の対応保証言語でサンプル AI および標準入出力のパースクラスのセットが随時公開される予定です。さらに、対応保証言語以外のプログラミング言語でも標準入出力をサポートしている言語であれば、AI の作成が可能です。ただし、自作言語等予選実行環境において動作しない言語に関しては保証し兼ねますので予めご了承ください(対応保証言語以外での実装における注意点は 9.3 提出方法を参照)。

7.1 キャラクターへの命令

命令の種類は、侍、犬共に以下の 5 つです。

- NONE : 動かない
- RIGHT : 右
- UP : 上
- LEFT : 左
- DOWN : 下

それぞれ文字列を標準出力から出力する事で指示を出す事ができます。(詳細はサンプル AI 等を参照してください)動作を決定する処理及び動作の命令は、後ほど公開される実行環境において 1000[msec]以内に行われる必要があります。**命令が 1000[msec]以内に行われない場合、その超過時間を加算していき、合計で 10 秒となりましたら失格となります(超過時間が累積 10 秒を超過して失格となったプレイヤーの侍、犬は共に命令が NONE(ゲームソフトウェアのアーカイブに同梱された README.txt およびサンプル AI を参照)として処理され続けます)**ので注意してください。なお、予選用の敵 AI については動作の時間制限はありません。

7.2 標準入出力のデータ仕様

ゲームシステムからの提供されるデータの仕様を以下に示します。

経過時間(int)

プレイヤーインデックス(int:)

マップの幅(int) マップの高さ(int)

タイル情報(char[height][width])

スコア(int)	位置の X 座標(int)	位置の Y 座標(int)	向き(int)	状態(int)	状態の残り時間(int)	: Samurai(0)
スコア(int)	位置の X 座標(int)	位置の Y 座標(int)	向き(int)			: Dog(0)
スコア(int)	位置の X 座標(int)	位置の Y 座標(int)	向き(int)	状態(int)	状態の残り時間(int)	: Smaurai(1)
スコア(int)	位置の X 座標(int)	位置の Y 座標(int)	向き(int)			: Dog(1)
スコア(int)	位置の X 座標(int)	位置の Y 座標(int)	向き(int)	状態(int)	状態の残り時間(int)	: Samurai(2)
スコア(int)	位置の X 座標(int)	位置の Y 座標(int)	向き(int)			: Dog(2)
スコア(int)	位置の X 座標(int)	位置の Y 座標(int)	向き(int)	状態(int)	状態の残り時間(int)	: Samurai(3)
スコア(int)	位置の X 座標(int)	位置の Y 座標(int)	向き(int)			: Dog(3)

- ・経過時間

経過したフレーム数

- ・プレイヤーインデックス

Player 0 : 0 , Player 1 : 1 , Player 2 : 2 , Player 3 : 3,

- ・タイル情報

壁 : * , スモールボーナス : s , ビッグボーナス : b , パワーボーナス : p , なし : (空白)

- ・スコア

各キャラクターが保持しているスコア

- ・位置の座標

キャラクターの位置座標(X, Y)

- ・向き

不明 : -1 , 右 : 0 , 上 : 1 , 左 : 2 , 下 : 3

- ・状態

通常状態 : 0 , 不可視状態 : 1 , 将軍状態 : 2

- ・状態の残り時間

侍が将軍状態または、不可視状態で命令を実行できる残りの回数で、通常状態であれば 0 となります。不可視状態は(10~1)、将軍状態は(30~1)となります。その状態になった瞬間が 10(不可視状態)または 30(将軍状態)で、自分が行動するフレームで命令を処理する毎に減っていきます。

・ サンプルデータ

以下にサンプルデータとその時のゲーム画面例を示します。

0112

0

17 17

```
*****
*   SSSS   S*
*S***** S*****
*s*bssss sb*s*
*s*s*s***** *s*s*
*s*ssssssss ss*s*
*s*s*s**b**s*s*s*
*s*s*s*ssss*s*s*s*
*   *sbspsbs*sbs*
* * *s*ssss*s* *s*
* * *s**b**s* *s*
* * sssssssss*s*
* * *s*****s*s*s*
* * sssssssssb*s*
* ***** S*****
*s*s*s*s*s   S*
*****
```

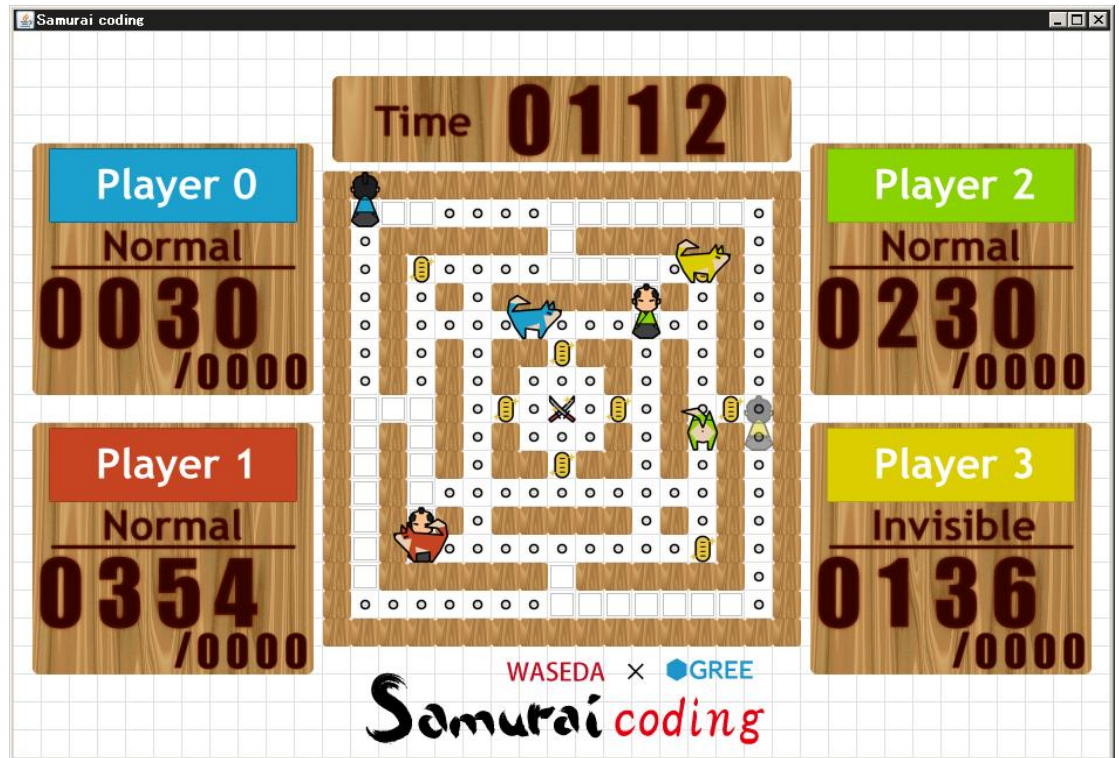


図 7.1. ゲーム画面例

30 1 1 1 0 0

0 7 5 0

354 3 13 3 0 0

0 3 13 2

230 11 5 3 0 0

0 13 9 1

136 -1 -1 -1 1 4

0 13 3 0

・ 上記のサンプルにおける各種データ

経過時間 : 112 (Samurai(0)が行動するフレーム)、プレイヤーインデックス 0、

Samurai(0) : スコア : 30、位置座標 : (1,1)、向き : 上、状態 : 通常状態、状態の残り時間 : 0

Dog(0) : スコア : 0、位置座標 : (7,5)、向き : 右

Samurai(1) : スコア : 354、位置座標 : (3,13)、向き : 下、状態 : 通常状態、状態の残り時間 : 0

Dog(1) : スコア : 0、位置座標 : (3,13)、向き : 左

Samurai(2) : スコア : 230、位置座標 : (11,5)、向き : 下、状態 : 通常状態、状態の残り時間 : 0

Dog(2) : スコア : 0、位置座標 : (13,9)、向き : 上

Samurai(3) : スコア : 136、位置座標 : 不明、向き : 不明、状態 : 不可視状態、状態の残り時間 : 4

Dog(3) : スコア : 0、位置座標 : (13,3)、向き : 右

* 敵侍が不可視状態の場合は位置座標および向きは取得できませんが、味方侍が不可視状態の場合は位置座標、向きともに取得できます。

8. サンプル AI

サンプル AI および標準入出力のパースクラスのセット(以降、サンプル AI セット)がゲームソフトウェアのアーカイブに同封および、WEB で公開されております。AI 作成の参考にしてください。運営側では、C、C++、C#、Java、Python、Ruby に対してサンプル AI セットを用意する予定です。サンプル AI セットが用意されていない言語についても、標準入出力が可能な言語であれば AI を作れます(つまり、**API は不要で、標準入出力をパースすれば AI の作成が可能**となります)が、運営側で検証を行っておりませんので、予めご了承ください。

9. 注意・その他

9.1 制限

各言語の標準 API や外部ライブラリの利用は自由となっております。そのため、ネットワークやファイルアクセス等は利用可能となっております。ただし、提出する際は、運営側がコンパイル&実行できるような形で提出をお願い致します。我々も動作させる努力は致しますが、パッケージ名の衝突等も含め、動作しない場合は失格とさせていただきます。また、バイトコードエンジニアリングライブラリ(BCEL)などのリバースエンジニアリング行為は固く禁止します。

9.2 実行環境

予選の最終結果は、予選 AI 提出締め切り後、運営側で用意した同じ環境のマシンで実行したうえで算出します。問題公開の時点では、OS は Ubuntu 11.10、GCC(C / C++)、Mono(.NET 系 C#など)、Ruby、Python 等を導入予定です。マシンスペックおよびバージョン等におきましては決定次第公開させていただきます。

現時点での実行環境

- ・ GCC 4.6.1 (Ubuntu/Linaro 4.6.1-9ubuntu3)
- ・ Mono C# compiler version 2.10.5.0
- ・ JDK 1.6.0_30
- ・ rvm 1.10.0

9.3 提出方法

予選の AI は、WEB に提出します。詳しい提出方法は公式サイトを参照してください。予選用敵 AI の公開(2011 年 12 月 23 日予定)後は、動作確認依頼された方には提出された AI が運営側の環境で動作するかどうか検証致します。提出から最長 3 日以内に運営側の環境で動作するかを返信致します。なお、動作確認依頼が終わるまでは次の依頼には対応できませんのでご了承ください。また、対応保証言語以外で AI を作成する場合は 1 度早めに動作確認依頼をしてください。我々も動作させる努力は致しますが、動かない場合は 9.1 に基づき失格となる可能性が御座いますのでご了承ください。なお、動作確認依頼方法の詳細については WEB を参照してください。

9.4 WEB ランキング

本大会の WEB 上にランキングが公開予定ですが、予選の最終結果には、皆様が最終に提出された AI を運営側で用意したマシンにより実行した結果を用います。そのため、WEB 上のランキングは必ずしも正しいものとは限りませんので遊び感覚でご利用ください。

9.5 予選・決勝方式

予選では、決勝大会に進む 16 チームの選抜を行い、順位によって決勝での対戦相手を決定します。予選は運

営側が用意した 3 つの AI と対戦して獲得したスコアを競う方式です。予選用の敵 AI は、2011 年 12 月 23 日の予定です。また、予選用 AI の提出期限は 2012 年 1 月 8 日となっています。予選結果の公開は 2012 年 1 月 10 日です。予選結果公開後、決勝進出チームは決勝大会に向けて、AI をブラッシュアップする期間が設けられています。決勝用 AI の提出期限は、2012 年 1 月 20 日です。決勝大会は 2012 年 1 月 22 日に開催され、16 チームの AI 同士で直接対戦します。トーナメント制でファーストラウンド、セミファイナル、ファイナルの 3 ラウンドを行います。このうち、ファーストラウンドとセミファイナルは 1 ゲームの結果で順位付けをし、4 チーム中上位 2 チームが次のラウンドに進出します。ファイナルは 3 ゲーム行い、その合計スコアで順位を決定します。なお、プレイヤーは、予選では“Player 0”が割り当てられ、決勝時は Player 0～Player 3 のいずれかが割り当てられます。

9.6 賞品

グリー株式会社様より、優勝チームには(Kinect+Xbox360)×2 をはじめ、準優勝チームにも素敵な賞品が用意されており、決勝大会に御来場された方々にもノベルティーが用意されています。

9.7 実行方法

実行方法については、ゲームソフトウェアのアーカイブに同梱された README.txt を参照してください。

9.8 参加確認について

本プログラミングコンテストでは予想をはるかに超える参加登録があった為、参加チームを先着 100 チームに制限させて頂きました。また、登録頂いたチームには 1 チームでも多く参加していただきたいと考えており、参加登録解除の制度を導入致しました。

上記の流れに従い予選参加辞退のチームが現れた場合に限り、追加募集を行わせていただきます。2011 年 12 月 29 日に先着順にて追加募集を行いますので、参加希望者はご確認の程よろしくお願い致します。

コンテスト日程(日本標準時、予定)

2011 年 12 月 16 日	: 問題公開
2011 年 12 月 23 日	: 予選開始(敵 AI 公開)
2012 年 01 月 08 日	: 予選用 AI 提出期限
2012 年 01 月 10 日	: 予選結果公開
2012 年 01 月 20 日	: 決勝用 AI 提出期限
2012 年 01 月 22 日	: 決勝大会開催

*本コンテストで指定している日程は全て日本標準時(JST)としています。各提出期限におきましても、日本標準時における 23 時 59 分 59 秒とします。

Samurai coding 公式 WEB サイト

http://www.washi.cs.waseda.ac.jp/samurai_coding/

Samurai coding 運営委員(五十音順)

大橋 昭、坂本 一憲、神谷 知行、塩濱 龍志、下條 清史、鈴木 翔大、福留 康之、和田 雅彦