

# SamurAI 3x3 API



SamurAI 3x3 (Samurai three on three) is a game played by an army of three samurai with different weapons, competing with another such army for wider territory. Contestants build an AI program that decides the actions of one of the samurai that plays the game in cooperation with two other friend samurai. The performances of AI programs are evaluated through many games played with varying weapons and varying army combinations.

This document specifies the input/output specification of AI programs for SamurAI 3x3 games.

## 1 Game Outline

---

The game battlefield is either square or rectangular and mesh-partitioned into sections. The objective of the game is to expand their territories; the number of sections they occupy on the battlefield. The performance of AI programs are evaluated by the total points obtained through many games played with varying weapons and varying army combinations.

### 1.1 Actions of Samurai

Each of the samurai are initially at their own home positions, and take actions in turn. AI programs are given information on the situation of (a part of) the battlefield, and decide actions in their turns. Actions include: moving around in the battlefield, attacking and occupying surrounding areas, hiding themselves from enemies' eyes, and showing themselves again. Each action is associated with its cost and actions in one turn can be an arbitrary combination of actions within the given budget.

Samurai has its own weapon (a spear, swords, or a battleax) on which the area occupied at a time depends. Enemy samurai in the occupied area are injured, brought back to their home positions, and cannot take actions for a certain number of subsequent turns.

The information a samurai obtains are:

### 1.2 Scoring

Each participating samurai receives the total of the following points after each game.

1. Winning points

The army with most total occupied section is the winner. All the samurai in the winning army are awarded

winning points. The winning points is fixed, regardless of the margin of the victory. If two armies have the same number of total section, both teams receives half of the winning points.

## 2. Occupation points

One occupation point is awarded to each individual samurai for every single section it occupies.

## 1.3 How Games Proceed

A game consists of a sequence of turns, in each of which one samurai takes actions. Each action has an associated cost and the samurai can take a set of actions within a given budget in each turn. The decision of which actions to take are decided by the participating AI programs based on game state information.

The samurai of each army, A and B, is identified by its weapon id, 0 through 2. Therefore, all the participating samurai are identified by A0, A1, A2, B0, B1, and B2. In each game, the participating samurai take their turns in the following order, which are repeated until the end of a game.

A0 B0 B1 A1 A2 B2 B0 A0 A1 B1 B2 A2

The number of total turns in a single game is a multiple of 12, with the minimum of 12 turns and the maximum of 1008 turns.

## 1.4 Communication Summary

The following figure depicts a summary of the communication between the game system and an AI program that controls a samurai.

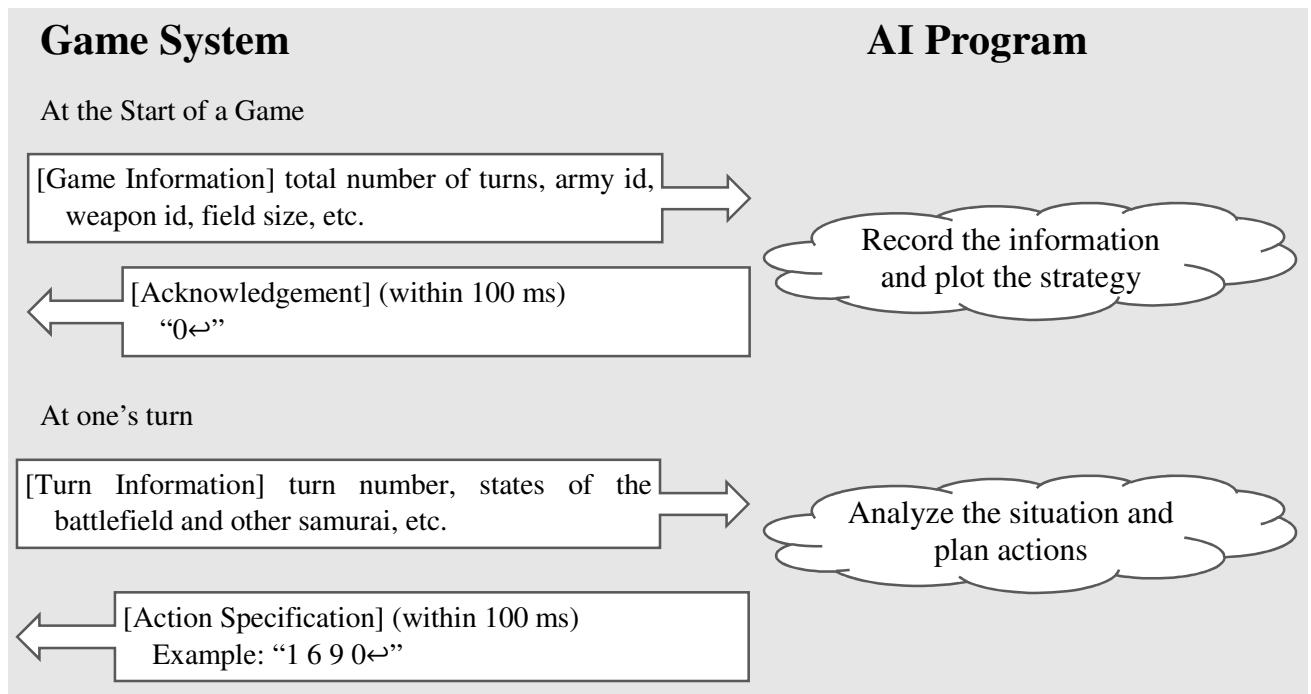


Figure 1: Communication Summary

## 2 Representation of the Battlefield and Samurai

### 2.1 Battlefield Sections and Coordinates

The battlefield of the game is mesh-partitioned to sections. Each section has two-dimensional coordinates  $(x, y)$ . The coordinate system is left-handed. With  $m$  sections east-west and  $n$  sections north-south, the northwest corner has the coordinates  $(0, 0)$  and the southeast corner has  $(m - 1, n - 1)$  as depicted in Figure 2

(0,0)	(1,0)	(2,0)	(3,0)	(4,0)
(0,1)	(1,1)	(2,1)	(3,1)	(4,1)
(0,2)	(1,2)	(2,2)	(3,2)	(4,2)
(0,3)	(1,3)	(2,3)	(3,3)	(4,3)

### 2.2 Weapons and Occupation

Samurai can occupy battlefield sections around depending on its weapon.

Figure 3 shows the IDs of weapons and the areas of their reach. Green circles in the figure show the positions of samurai and sections painted green can be occupied using the weapons. The figure shows areas when occupation actions are southward. Occupation can also be directed westward, eastward, and northward.

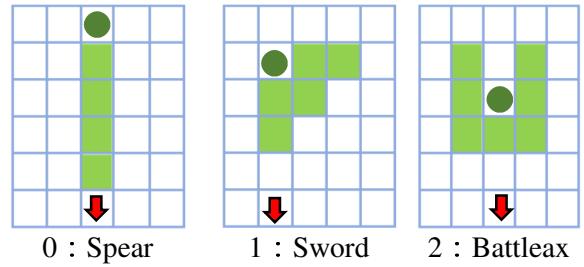


Figure 3: Weapons and Occupied Areas

### 2.3 Samurai ID

Two armies of three samurai play against each other, making a total of six participating samurai in each game. Each participating samurai are identified by a **samurai ID** in the AI program from its own perspective. Each of the three samurai of one army uses a different weapon, and thus each member can be identified by its weapon ID. Suppose a samurai in the friendly army uses a weapon with ID  $n$ , then it has the samurai ID of  $n$ , while the enemy samurai who uses same weapon has the ID  $n + 3$ . For example, if the samurai of the AI program uses a spear then it identifies itself with the samurai ID of 0, and a friendly samurai with a sword has the samurai ID of 1, while the samurai on the opposing army using the battle-axe has the samurai ID of 5.

## 3 AI Programs

### 3.1 Outline

AI programs are initiated by the game system. On its initiation, the AI program receives the overall information and specification of the game (**game information**) from the standard input. The AI program then responds with an acknowledgement to the game system via the standard output, indicating that it is ready. After this initiation phase, the AI program will receive information on the state of the game (**turn information**) from the standard input when it is its turn to take action, and specifications of the actions that the AI program intends to take (**action instructions**) are written to the standard output.

### 3.2 Formats of Transferred Data

All the contents are represented in ASCII characters. Numerical values are decimal integers (a minus sign ('-') in front of the number indicates a negative value). Decimal integers are delimited with blanks and

newlines. A sharp sign ('#') indicates comments, and whatever follows until the end of the line are considered to be a part of the comment.

### 3.3 Time Limit

AI programs should respond within the specified time limit.<sup>1</sup> Failing to meet the time limit disqualifies the samurai. The disqualified samurai will be returned and remain in the home position until the end of the game, and cannot perform any further action.

### 3.4 After Completing a Response

The AI program will be suspended by the game system after completing its response until its next turn to take action.

## 4 Game Information

---

**Game information** consists of the following in this order.

1. Total number of turns
2. Army ID
3. Samurai ID
4. Battlefield size
5. Recovery period
6. Home positions
7. Army composition

Details are described in the following subsections.

### 4.1 Total Number of Turns

The **total number of turns** is the number of turns played in a single game. On each turn, one of the participating samurai take a number of actions within its given budget. The total number of turns is always a multiple of twelve and thus all the six samurai has the same number of turns to take actions in each game.

### 4.2 Army ID

If the samurai of the AI program belongs to the army A, then has the army ID of 0, and if it belongs to B, then it has the army ID of 1. Combining the army ID and the samurai ID specified in the following subsection, the AI can deduce the samurai's turn to make its action.

### 4.3 Samurai ID

The **samurai ID** is the ID of the samurai whose turn it is to perform its actions. The possible value of the specified samurai ID is one of 0, 1, or 2, which is the ID of the weapon the samurai is using.

---

<sup>1</sup> The time limit is planned to be 100 milliseconds, but is subject to change depending on the platforms. This is not CPU time but wall-clock time elapsed after the game management system sent information to an AI until the game system finishes receiving the response.

## 4.4 Battlefield Size

The **battlefield size** is specified two integers  $w$  and  $h$ . Here,  $w$  is the number of sections in the east-west way and  $h$  is north-south. Both  $w$  and  $h$  are between 10 and 20, inclusive.

## 4.5 Recovery Period

The recovery period is the number of turns injured samurai have to take rests before they can take actions again. The period is the same for all the samurai in a game and is between 12 and 48, inclusive.

## 4.6 Home Positions

**Home positions** give coordinates of home positions of all six samurai in the order of samurai IDs, each in two integers  $x_0$  and  $y_0$  meaning the position  $(x_0, y_0)$ . Home positions are initial positions of samurai. They are at sections on one (or two) of the outer edge(s) of the battlefield.

The sections which are the home positions of samurai are already occupied by the corresponding samurai, and can never be occupied by any other samurai.

## 4.7 Army Composition

**Army composition** is information on performance shown so far by all six samurai participating in the game. For each samurai, two integers, the **rank** and **score sum** up to the game just before. These are given for all six samurai in the order of their ID, and thus there will be twelve integers.

The **rank** is the rank of the samurai among all the samurai participating in the *tournament*, not only those participating in this *game*. The **score sum** is the total of the score the samurai obtained so far. The highest rank is zero. When there are more than one samurai with the same score sum, they have the same rank. On the very first game of a tournament, all the samurai have their rank and score sum of both zero.

## 4.8 Game Parameters

Game parameters such as battlefield sizes and home positions of samurai are decided for each tournament, but remains the same for all the games in one tournaments. They are noticed to the participants before submissions of AI programs.<sup>2</sup>

---

<sup>2</sup> Battlefield sizes and home positions are given in the game information so that the same software can be used on different tournaments with different parameters.

## 4.9 Game Information Example

Figure 4 shows an example of game information.

- Lines starting with “#” are comments.
- The second line says that there will be total of 192 turns in the game, the samurai to control belongs to the army A and has weapon 2 (a battleax), the battlefield has the size of  $15 \times 15$ , and the cure period is 24.
- Six lines starting from the fourth tells the home positions. As the samurai to control has its ID 2, its home position is at (9,14).
- Six line starting from the eleventh gives the army composition. The samurai to control had rank 4 (as the highest rank is 0, it means the fifth position) and score sum of 31.

```
# Game Info
192 1 2 15 15 24
# Home positions
0 5
0 14
9 14
14 9
14 0
5 0
# Ranks and scores of samurai
3 40
7 24
4 31
1 80
0 93
6 28
```

Figure 4: Game Information Example

## 5 Acknowledgement Response to the Game Information

Receiving the game information, the AI program should return an integer zero and a newline to acknowledge it. This may include comments and extra spaces, but the total of characters should not exceed 100. The acknowledgement response should be made within the time limit, or else the samurai will be disqualified.

## 6 Turn Information

**Turn information** consists of the following in this order.

1. Turn number
2. Recovery period
3. Samurai states
4. Battlefield states

Details are discussed below.

### 6.1 Turn number

**Turn number** is the sequential number of the turn for which the actions of the samurai should be planned. Turn numbers start with zero and is less than the total number of turns.

### 6.2 Recovery period

When a samurai is injured by an enemy attack, it is immediately brought back to its home position and, for a certain number of turns, it cannot make any actions for recovery. Although no actions can be taken, the samurai can observe the battlefield states during this period. **Recovery period** tells how many more turns this recovery period will continue. If the samurai is not injured, it is zero and immediate actions are possible.

### 6.3 Samurai States

**Samurai states** are information of all six samurai immediate before the turn. As information for one samurai is three integers, showing its current position and status, there are twelve integers in total.

The **current position** has two integers  $x$  and  $y$ , which gives the coordinates  $(x, y)$  of the samurai before the turn. Positions of friendly samurai are always found, but those of enemy samurai cannot be found when they are out of the vision of all friendly samurai or they are hiding. For such samurai, their current positions are indicated as two  $-1$ .

The **status** is whether or not the samurai is hiding itself. For friendly samurai, it is 1 when it is hiding and 0 otherwise. For enemies, you cannot tell whether they are hiding or simply out of the vision. The hiding status of 1 is given in either case. The status of a disqualified samurai is  $-1$  for both friendly and enemy samurai.

### 6.4 Battlefield State

**Battlefield state** gives occupation states of battlefield sections.

The state of a section is given as a single integer. They are given in the ascending order of  $y$  and  $x$  axes, as  $w \times h$  integers. Not all the sections are provided with their state information. Only those in the vicinity of friendly samurai are visible. The sections for which information are provided are those within the Manhattan distance of 5 from the positions of friendly samurai, that are, with a friendly samurai is at  $(x, y)$ , sections at  $(x + d_x, y + d_y)$  for  $|d_x| + |d_y| \leq 5$  (Figure 5). Even if a samurai is in recovery or is disqualified, it can provide battlefield state information around its position (which is its home position).

Information on each section is encoded as follows.

- 0 to 5: This section was last occupied by a samurai with that samurai
- 8: Not occupied with any samurai yet.
- 9: No information is provided as there are no friendly samurai nearby.

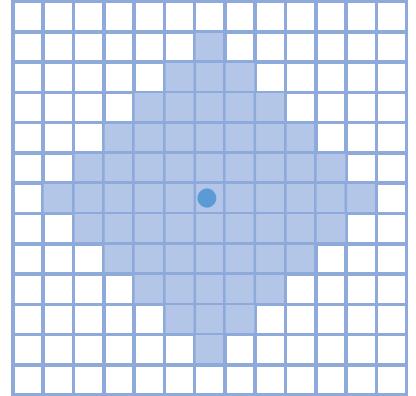


Figure 5: Field of Vision

## 6.5 Turn Information Example

An example of turn information is shown in Figure 6.

## 7 Action Specification

A samurai can be instructed to take any of the following ten actions of four categories within their cost sum of seven.

- Occupy (1 through 4; cost 4): Occupies neighboring sections. The sections occupied are decided by the weapon used and the direction to apply it. The sections that can be occupied are shown in Figure 3. Directions to apply the weapon is specified as 1 for southward, 2 for eastward, 3 for northward, and 4 for westward. Occupation actions cannot be made while hiding.
- Move (5 through 7; cost 2): Moves to one of the adjacent sections. If the samurai is not hiding itself, it cannot move to a sections in which a non-hiding samurai is in. A samurai hiding itself can only move to a section already occupied by one of the friend samurai. The move direction is specified as 5 for southward, 6 for eastward, 7 for northward, and 8 for westward.
- Hide (9; cost 1): Hides itself. Only possible when the samurai is in a section occupied by a friendly samurai (including itself).
- Show (10; cost 1): Stop hiding itself. A samurai cannot stop hiding if there is another non-hiding samurai in the same section.

**Action specification** consists of a list of integers indicating the above actions, with their total cost not exceeding the give amount of budget 7, and terminates with a zero. For example, an action specification consisting of a single 0 indicates that no actions are to be made.

When an invalid action, such as moving out of the battlefield or into an enemy territory while hiding itself, such an action and any following actions instructed for the same turn are invalidated, although action instructions prior to it are effective. When the total cost of the actions exceed the given budget, action instructions up to their total cost of seven are valid, and the rest are invalid. Action specification for one turn should not exceed one hundred characters, and the samurai will be disqualified if this is violated.

The AI programs are still required to give action specification even if it is in the recovery period, and the samurai will be disqualified if the AI program fails to give any action specification.

An example of action specification is given in Figure 7. This specifies three actions, southward occupation, followed by a move eastwards, and then followed by hiding. The sum of the costs of actions is seven in this case.

```
# Turn information
# <turn>
12
# <cure period>
0
# Samurai states
0 6 1
1 14 1
9 12 1
-1 -1 1
-1 -1 1
-1 -1 1
# Battle field states
9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9
8 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9
8 8 9 9 9 9 9 9 9 9 9 9 9 9 9 9
8 8 8 9 9 9 9 9 9 9 9 9 9 9 9 9
8 8 8 8 9 9 9 9 9 9 9 9 9 9 9 9
8 0 0 0 0 9 9 9 9 9 9 9 9 9 9 9 9
0 8 8 8 8 9 9 9 9 9 9 9 9 9 9 9 9
0 8 8 8 8 9 9 9 9 8 9 9 9 9 9 9 9
0 8 8 8 9 9 9 9 8 8 8 8 8 8 9 9 9
0 8 8 9 9 9 9 8 8 8 8 8 8 8 8 9 9
8 8 8 9 9 9 8 8 8 8 8 8 8 8 8 9 9
8 8 8 8 9 8 8 8 8 2 8 2 8 8 8 8 9
1 1 8 8 8 8 8 8 2 2 2 8 8 8 8 8
1 1 1 8 8 8 8 8 2 2 2 8 8 8 8 9
8 1 1 1 8 8 8 8 2 2 2 8 8 8 9 9
```

Figure 6: Turn Information Example

```
# Sample actions
1 6 9 0
```

Figure 7: Action Spec. Example

## 8 Miscellaneous

---

- In the acknowledgement response to the game information or the action specifications, any content following the terminating 0, including comments, will not be processed. The unprocessed contents will show up as the output of the AI program in its next turn, so it might cause the program to act unpredictably.
- AI programs will be terminated by the game system. Therefore, the AI programs are not required to process its own termination.