



SamurAI 3x3 2016

Rules and Program Interface

IPJSJ Programming Contest Committee

version September 14, 2016

Draft, subject to change

Abstract

SamurAI 3x3 (Samurai three on three) is a game played between two armies each consisting of three samurai with different weapons, competing for wider territory. Each contestants builds an AI program that decides the actions of the samurai of its army. This document describes the rules of SamurAI 3x3 2016 used in the SamurAI Coding 2016-2017 contest and the input/output specification of AI programs.

1 Game Outline

The game *battlefield* is mesh-partitioned into 15×15 *sections*. The objective of the game is to for each army to expand its territory: the number of sections it occupies on the battlefield.

1.1 Game Organization

A game is played between two *armies*. Each army has three *samurai* wielding different kinds of *weapons*, namely, a spear, swords, and a battleaxe. A samurai can occupy sections around it, and the shape of the occupied sections depends on the weapon.

Each game consists of a series of *turns*, in which two armies play in turn. At each turn, one of the armies instructs one of its three samurai with a sequence of actions, within a given budget.

A series of six consecutive turns is called a *period*. One game consists of 16 periods, or 96 turns. In each period, each army will have three turns, and each of an army's samurai will be instructed with a sequence of actions once.

Two games played between the same pair of armies is called a *match*. Each of the two armies plays first in one of the two games of a match, and plays second in the other game of a match.

1.2 Actions of Samurai

Each of the samurai are initially at its own home position. An AI program is given information on the situation of (a part of) the battlefield at its turn, and decides which samurai to act and its actions.

Possible actions include: attacking and occupying surrounding areas, moving around in the battlefield, hiding itself from enemies eyes, and showing itself again. Each action is associated with a cost and actions in one turn can be an arbitrary combination of actions within the given budget.

The battlefield state information a samurai is given is limited to the fields of vision of samurai of the friendly army. Specifically, the information given for each section in the vision field is whether the section has ever been occupied by samurai, and, if so, which samurai occupied it last. For friend samurai, their current positions and whether they are showing themselves or not can be known, while enemy samurai can be detected only when they are showing themselves and are within the vision field of a friend samurai. AI programs have to decide the orders for its three samurai with such limited information and within the given time limit.

1.3 Deciding the Match Result

At the end of the two games of a match, the numbers of sections occupied by each of the armies are summed up. The army with the greater sum is the winner of the match. If two armies have the equal sum of the numbers of occupied sections, the match is a draw.

2 Representation of the Battlefield and Samurai

2.1 Battlefield Sections and Coordinates

The battlefield of the game is partitioned into a mesh of 15×15 sections. Each section has two-dimensional coordinates (x, y) . The coordinate system is left-handed (See Figure 1). The northwest corner has the coordinates $(0, 0)$ and the southeast corner has $(14, 14)$.

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

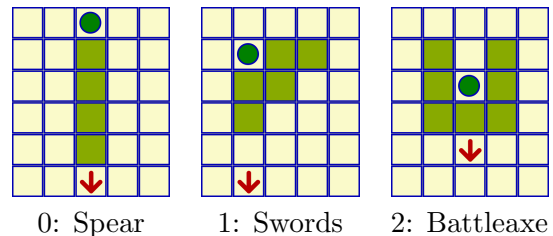
2.2 Weapons and Occupation

The battlefield sections that a samurai can occupy depends on its weapon. Figure 2 lists the *weapon IDs* and the areas of their reach. Blue 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.

Regardless of the weapon used, the section where the samurai stands cannot be occupied. Home positions of other samurai cannot be occupied either.

2.3 Injury and Treatment

If a samurai conducts an occupy action on sections in which one or more enemy samurai currently reside, these enemy samurai will



0: Spear 1: Swords 2: Battleaxe

Figure 2: Weapons and Occupied Areas

become injured and require medical treatment. This will happen regardless of their showing/hiding status. They will show themselves if hidden, and be brought back to their home positions immediately. For the following 18 turns of treatment, they cannot take any actions.

Note that 18 turns of injury treatment does not always mean losing the samurai's turns of three periods. Consider a case in which a samurai took its actions in the first turn of a period (say, period p), and then was injured in the next turn, that is, the second turn of the period p . As the injured samurai already took its actions in this period p , and thus it cannot take actions in its remaining four turns anyway. In the succeeding two periods, periods $p + 1$ and $p + 2$, the samurai cannot take any actions, however, in the third turn of the period $p + 3$, it recovers from the injury and can take actions.

2.4 Home Positions

Two armies of three samurai play against each other, making a total of six participating samurai in each game. Each participating samurai is identified by its weapon ID. Samurai are initially in their *home positions* as listed in Table 1.

The home positions are occupied by corresponding samurai before the game starts. They will never be occupied by other samurai throughout the game.

Table 1: Home Positions

Weapon	Coordinates	
	First	Second
0: Spear	(0,0)	(14,14)
1: Swords	(0,7)	(14,7)
2: Battleaxe	(7,0)	(7,14)

3 AI Programs

3.1 Outline

AI programs are initiated by the game system. On its initiation, an AI program receives the overall information and specification of the game (game information) from the standard input. It then should respond with an acknowledgment 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 on its turn, and the order, consisting of a samurai ID and the sequence of actions to take, are to be sent through 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.

3.3 Time Limits

AI programs should respond within the imposed *time limits*. The time limit is 10 seconds for the response to the game information at initiation; it is 200 milliseconds for the response at each of the play turns.

The time limits are not imposed on CPU time, but wall clock time elapsed between when the game system completes sending information to the AI program and when it completes receiving the response. Failing to meet the time limit disqualifies the army: All the samurai of the army will show themselves and be sent back to their home positions. They cannot take any actions until the end of the game.

3.4 After Completing a Response

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

4 Game Information

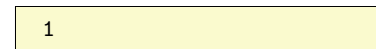
Game information tells whether the army is the first player or the second player.

4.1 Play Order

When the army plays first, it is indicated by a 0; otherwise, that is, it plays second, 1.

4.2 Game Information Example

Figure 3 shows an example of game information. It says that the army is the second player in the game.



4.3 Acknowledgment 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 extra spaces, but the total number of characters should not exceed 100.

Figure 3: Game Information

5 Turn Information

Turn information consists of the following in this order.

1. Turn number
2. Samurai states
3. Battlefield states

Details are discussed below.

5.1 Turn number

The *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 96, that is, the total number of turns, or, six times the total number of periods.

5.2 Samurai States

The *samurai states* are information of six samurai immediately before the turn. Information for one samurai consists of five integers: two for its current position, one for its order status, that is, whether or not an order for the samurai has already been placed in the same period, and one for its showing status, that is, whether or not it shows itself, and, finally, one for the number of treatment turns. The samurai state information is given first for three samurai of the friendly

army in the order of their weapon IDs, and then for three in the enemy army in the same order. There are, thus, 30 integers in total.

The *current position* is indicated with 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 *order status* is 1 if the samurai already has given an order in the same period, and 0 otherwise. This also applies to samurai in injury treatment that cannot take any actions.

For friendly samurai, the *showing status* is 1 when it is hiding itself and 0 otherwise. For enemies, you cannot tell whether they are hiding or simply out of the vision. The showing status of 1 is given in either case.

When a samurai is in the section occupied by an enemy samurai, it will show itself and be immediately brought back to its home position. During 18 consequent turns of injury treatment, it cannot take any action. The number of *treatment turns* is the number of turns remaining for the samurai to recover from injury. It is 0 if the samurai is not currently under treatment. This information is provided even if the samurai is out of the vision fields of any of the friendly samurai.

5.3 Battlefield State

Battlefield state gives occupation states of battlefield sections.

The state of a section is given as a single integer. The states are given in the ascending order of y and x axes, as $15 \times 15 = 225$ integers. Not all the sections are provided with their state information. Only those in the vicinity, or *vision field*, of a 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 is, with a friendly samurai is at (x, y) , information is provided for sections at $(x + d_x, y + d_y)$ for $|d_x| + |d_y| \leq 5$ (Figure 4). Even if a samurai is in recovery or is disqualified, it can provide battlefield state information around its position (which is its home position).

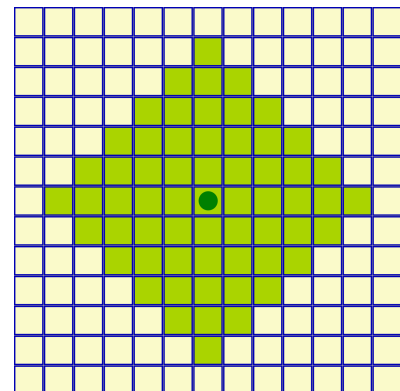


Figure 4: Field of Vision

Information on each section is encoded as follows.

0 to 2: This section was last occupied by a friendly samurai with weapon ID 0 to 2.

3 to 5: This section was last occupied by an enemy samurai with weapon ID 0 to 2.

8: Not occupied with any samurai yet.

9: No information is provided as there are no friendly samurai nearby.

5.4 Turn Information Example

An example of turn information is shown in Figure 5.

6 Ordering

An order to a samurai consists of the weapon ID of the samurai followed by a sequence of integers specifying its actions and terminated with an integer 0.

6.1 Actions of Samurai

Samurai can take any of the following 9 *actions* of 3 categories.

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 2. 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 8; 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 friendly territory section. Whether showing or hiding itself, home positions of other samurai cannot be entered. The move direction is specified as 5 for southward, 6 for eastward, 7 for northward, and 8 for westward.

Show/Hide (9; cost 1): Switches its showing state. This can be done more than once in the action sequence of a single order. Hiding is only possible when the samurai is in a friendly territory section. Showing is not possible if there is another non-hiding samurai, either friendly or enemy, in the same section.

The constraints on move and show/hide actions ensures that no two samurai can share the same section when both are showing themselves. On the other hand, more than one hidden samurai can share a section occupied by a friend.

6.2 Order

An *order* tells which samurai to take actions and a sequence of actions.

The samurai to take actions is specified by the ID of its weapon. The action sequence is specified by a list of integers which denote the actions described in the previous section, within the cost total of 7, followed by a 0 to indicate the end of the list. An order with only a weapon ID and a 0, for example, means that the specified samurai is ordered to do nothing.

When the specified weapon ID is that of a samurai already ordered in the same period, the order is invalid and no actions are taken. Invalid actions (moving out of the battlefield, moving into an enemy territory while hiding itself, trying to hide itself in an enemy territory, etc.) and all the following actions in the action sequence are ignored, although actions before the invalid one are effective. When the total cost of the actions exceeds the given budget, actions up to their total cost of 7 are valid, and the rest are ignored. An order given for a single turn should not exceed 100 characters, and the army will be disqualified if this is violated.

Samurai in under injury treatment can be ordered, although actions specified will be ignored.

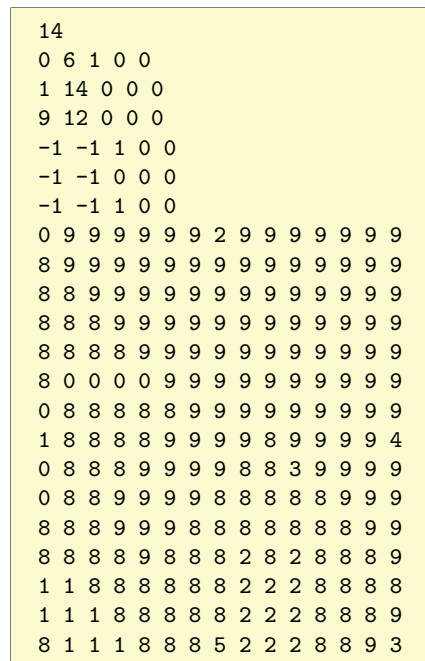


Figure 5: Turn Information

6.3 Order Example

An example of an order is given in Figure 6. This order tells three actions to the samurai with swords (1), southward occupation (1), move eastwards (6), and then switch show/hide (9), in this order, followed by a 0 to indicate the end. The sum of the action costs is seven ($= 4 + 2 + 1$) in this case.

1 1 6 9 0

Figure 6: Order

7 Miscellaneous

- In the acknowledgment response to the game information or the orders, any content following the terminating 0 will show up as the output of the AI program in its next turn.
- AI programs will be terminated by the game system. Therefore, the AI programs are not required to process their own termination.
- The game has many aspects common with the SamurAI 3x3 game used in the SamurAI Coding 2015-2016 contest. Major differences are the following.
 - A single program controls all three samurai in an army.
 - The samurai to take actions is specified in an order.
 - Parameters such as the battlefield size, the total number of turns, and the number of turns for injury treatment are fixed.
 - Actions to switch between show/hide status are merged.

A number of changes other than those listed above are also made on the API.

8 Updates

The following are major updates from prior draft editions.

- The notion of a match, which is a pair of games with switched play orders, is introduced. The winner is decided based on each match.
- Comments in input and output no longer exist.
- Time limits are made longer.

Index

actions, 6
armies, 1

battlefield, 1

current position, 5

home positions, 3

injury, 3

match, 1

order, 6
order status, 5

period, 1

samurai, 1
samurai states, 4
sections, 1
showing status, 5

time limits, 3
treatment turns, 5
turn number, 4
turns, 1

vision field, 5

weapon ID, 2
weapons, 1