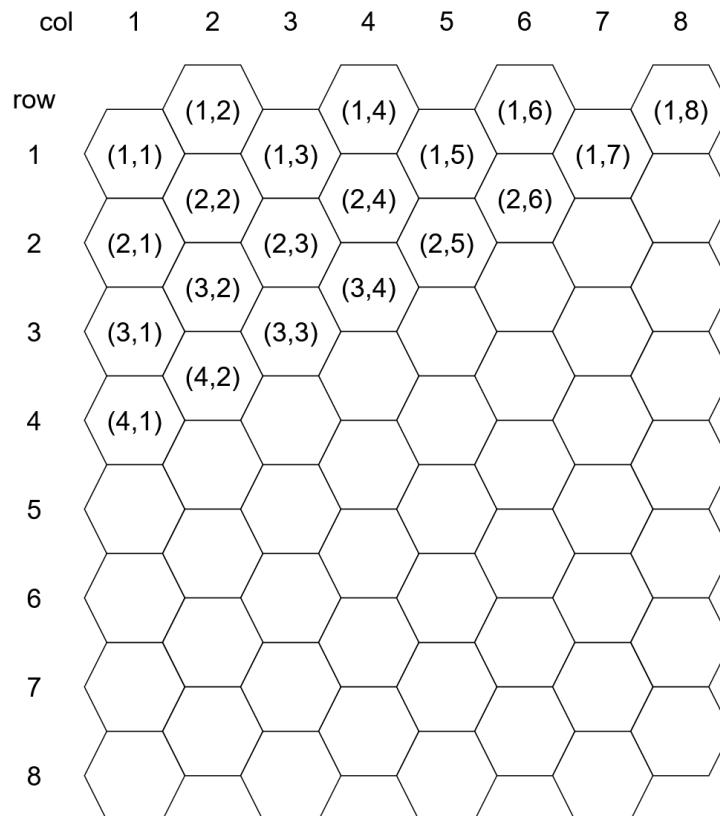


# Kickstart Offense with Minion's Best in an Amicable Territory (KOMBAT)

As a leader of your tribe residing in a limited-area territory, you have trained an army of minions capable of performing a variety of martial arts. Scarcity of resources has caused your tribe to split into two factions, each equipped with its own army that you have trained. The other faction has appointed a leader of its own, ready to challenge the status quo with you. To ensure your faction's survival, your job is to strategize your remaining army to show that you deserve to remain a true leader of the tribe.

## Overview of KOMBAT

KOMBAT is a turn-based game in which two players attempt to eliminate the other player's *minions* on the playing field. The playing field is an  $8 \times 8$  field of hexagons (called *hexes* for short). The coordinate of a hex is specified by a pair of row and column numbers of the hex. Observe that hexes on the same row but adjacent columns will skew vertically, as shown in the following figure.



A *minion strategy* is a script that dictates how a minion will act in each turn. Before the game begins, both players agree on the strategy for each kind of minion. These strategies will be used until the end of the game and cannot be modified partway.

The game ends when all the minions belonging to one of the players have been eliminated; the other player is the winner. The game also ends when each player has played a specified number of turns. In the latter case, the winner is determined by the number of minions left in the territory, the HPs for the remaining minions, and the remaining budget.

## Setup

First, the game should read the [configuration file](#) for necessary parameters.

Each game will work in one of three modes: duel, solitaire, and auto. In the duel mode, two players take turns playing; in the solitaire mode, one of the players becomes a bot, but the game proceeds identically otherwise. In the auto mode, both players become bots.

Before the start of the game, the players (excluding the bot) first agree on the number of kinds of minions that may be placed during the game. The game supports between 1 and 5 kinds of minions. The players should assign a name to each kind of minion. Each kind of minion has a *defense factor*, which will be used to reduce the amount of attack damage, and a strategy that will be executed in each turn. The players should agree on the defense factor and strategy for each kind of minion during setup. Minion strategies and defense factors cannot be altered once the game begins.

In the auto mode, humans still choose minion strategies and defense factors.

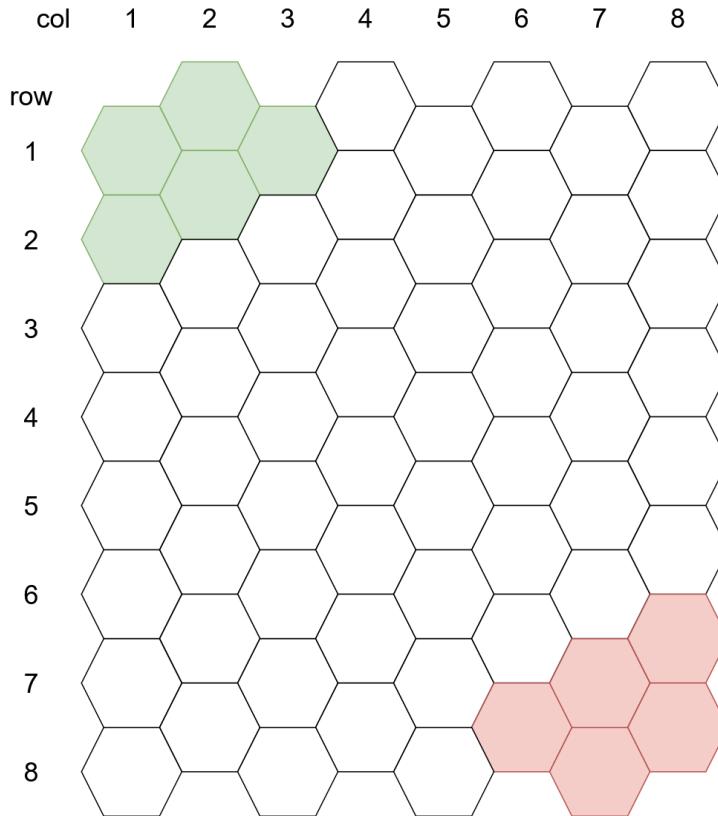
During setup, after agreeing on minion kinds, each player must spawn one minion of choice, for free. The [Gameplay section](#) describes valid spawn locations for each player.

Each player (including the bot) will be given an equal amount of initial budget as specified in the configuration file. Then, it is the first player's turn to start the game.

## Gameplay

At the beginning of each turn, the current player (including the bot) is given additional budget as specified in the [configuration file](#). Next, the overall budget accrues interest. If the resulting budget exceeds the maximum allowed budget as specified in the configuration file, the excess budget is forfeited. Then, the player will have an opportunity to spawn a new minion in a designated area. Initially, the designated area of the first player is the five hexes at the top-left corner of the territory; the designated area of the second player (or the bot) is the five hexes at the bottom-right corner of the territory, as illustrated in the following figure. The player can only spawn a new minion in an empty hex within the designated area. If all such hexes are occupied, the player loses this opportunity.

Each player can spawn new minions up to the number specified in the configuration file. Additional spawns are prohibited after this number is reached.



Additionally, before spawning a minion, the player has an opportunity to purchase a new hex so as to spawn new minions therein. This new hex must be adjacent to the existing hexes that can spawn new minions; attempts to purchase a hex elsewhere are not permitted.

The newly spawned minion will have an initial HP as specified in the configuration file. Purchasing a new spawnable hex and spawning a new minion incur costs as specified in the configuration file.

Once the player has finished spawning a new minion (if possible), all the minions belonging to the player will execute its own strategy, one minion at a time. The oldest minion executes its strategy first; the most recently spawned minion does so last.

After that, the other player's turn begins.

### Summary of the order of actions in each turn

1. The player's budget increases by a set amount and accrues interest according to the [calculated interest rates](#).
2. The player may choose to buy a new spawnable hex adjacent to the existing spawnable hexes.
3. The player may choose to spawn a new minion of a chosen kind on a spawnable hex.
4. Each minion, from oldest to newest, executes its own strategy.

## Sample minion strategy

# indicates the beginning of a comment, and is not part of the minion strategy. You can choose to support end-of-line comments in your tokenizer if desired.

```
t = t + 1 # keeping track of the turn number
m = 0 # number of random moves this turn
while (3 - m) { # made less than 3 random moves
    if (budget - 100) then {} else done # too poor to do anything else
    opponentLoc = opponent
    if (opponentLoc / 10 - 1)
        then # opponent afar
            if (opponentLoc % 10 - 5) then move downleft
            else if (opponentLoc % 10 - 4) then move down
            else if (opponentLoc % 10 - 3) then move downright
            else if (opponentLoc % 10 - 2) then move right
            else if (opponentLoc % 10 - 1) then move upright
            else move up
    else if (opponentLoc)
        then # opponent adjacent to this minion
            if (opponentLoc % 10 - 5) then {
                cost = 10 ^ (nearby upleft % 100 + 1)
                if (budget - cost) then shoot upleft cost else {}
            }
            else if (opponentLoc % 10 - 4) then {
                cost = 10 ^ (nearby downleft % 100 + 1)
                if (budget - cost) then shoot downleft cost else {}
            }
            else if (opponentLoc % 10 - 3) then {
                cost = 10 ^ (nearby down % 100 + 1)
                if (budget - cost) then shoot down cost else {}
            }
            else if (opponentLoc % 10 - 2) then {
                cost = 10 ^ (nearby downright % 100 + 1)
                if (budget - cost) then shoot downright cost else {}
            }
            else if (opponentLoc % 10 - 1) then {
                cost = 10 ^ (nearby upright % 100 + 1)
                if (budget - cost) then shoot upright cost else {}
            }
        else {
            cost = 10 ^ (nearby up % 100 + 1)
            if (budget - cost) then shoot up cost else {}
        }
    else { # no visible opponent; move in a random direction
        try = 0 # keep track of number of attempts
        while (3 - try) { # no more than 3 attempts
            success = 1
```