

Virus fight

A multi-gamesystem game by Jorge Arroyo

Version 1.0, Feb 2008. (c) 2007-8 Jorge Arroyo.

Email: trozo@makasoft.net Web: <http://george.makasoft.net>

Released under the Creative Commons by-nc-nd License.

Written using NeoOffice/J 2.2.1

2 players: 1 piecepack set OR 2 Treehouse sets OR 2 decks of cards.

3-4 players: 2 piecepack sets OR 3-4 Treehouse sets OR 3-4 decks of cards.

15-30 minutes.

- **Introduction**

This was my second Icehouse design. At some point I realized it didn't really require Icehouse pieces to be played, in fact it could perfectly be played with a piecepack set (2 sets for a 3-4 player game) or a deck of cards.

The game is a luckless abstract with a light theme of computer programming. Each player builds a small program that then modifies itself and the other programs on the board, trying to isolate the instruction marker of the other players so that it's the only running program in the memory.

- **Specific Game System Information**

Here I'll describe the components of the game for each game system. In the rest of the rules, the descriptions will be neutral.

- IF YOU'RE PLAYING WITH ICEHOUSE PIECES:

- Computer Instructions = Icehouse Pyramids: For each player in the game, you need one Treehouse set (either rainbow or xeno). It's better if all the sets are of the same type, but the game is playable with mixed types. Here's how you map Icehouse colors to Instruction types:
 - YELLOW / ORANGE = MOVE
 - GREEN / CYAN = WRITE
 - BLUE / PURPLE = JUMP
 - RED / CLEAR = ERASE

- Priority Numbers = Pips from each pyramid: The pip number from each pyramid is its priority number.
 - Instruction Marker = Small Opaque pyramid or hollow pawn: You need a hollow marker for each player. a small opaque Icehouse piece will work fine or a piece in a color not used for the instructions. Hollow plastic pawns will also do. In the end each player needs to have a unique marker that can be placed on top of the icehouse pieces.
 - Board = Volcano Board: The game uses a 5x5 board. You can use the one included with these rules or a regular 5x5 volcano board. A 3-4 game will benefit from a bigger board (6x6) or you can use the imaginary spaces outside the 5x5 board, but adjacent to it as legal spaces.
 - First player marker = Stack of left over opaque pieces: Make a stack with the opaque pieces that won't be used in the game. This will be used to mark the first player for each round.
- IF YOU'RE PLAYING WITH A PIECEPACK SET:
- Computer Instructions = Piecepack Coins: Each player will need three coins of each of the four suits. You can play with standard suits or expansion suits (or mix both on a 3-4 player game). Here's how you map piecepack suits to Instruction types:
 - MOON / FALL / DIAMONDS = MOVE
 - SUNS / SUMMER / HEARTS = WRITE
 - CROWNS / SPRING / CLUBS = JUMP
 - ARMS / WINTER / SPADES = ERASE
 - Priority Numbers = Coin Angle on the board: Each coin must be placed at an angle throughout the game. Each direction equals one priority number (North = 1, East = 2, South = 3). The players must display all their coins at all times and they must have one coin for each direction. So, for a two player game there will always be two coins looking north, two looking east and two looking south. It's important to maintain the directions, especially when moving coins on the board and when capturing and re-using coins.
 - Instruction Marker = Piecepack Pawns: Each player chooses a different pawn as their instruction marker. The suits / colors on the pawns are unrelated to the type of instructions for the coins, and will only be used to identify each player's marker.

- Board = Piecepack Tiles: For a 2 player game, build a 3x3 tiled board and use the 5x5 inside intersections as spaces to place the instructions. For a 3-4 board build use the small squares that form a 6x6 board to place the coins.
- First player marker = Stack of left over tiles: Make a stack with some left over tiles that won't be used in the game. This will be used to mark the first player for each round.

- **IF YOU'RE PLAYING WITH STANDARD CARDS:**
 - Computer Instructions = Face Up Cards: Each player will need three cards of each suit. Ideally they all need to be numbered with the same numbers (from 1 to 3), so a deck of cards is needed for each player. The suits map to the instructions as in the piecepack set.
 - Priority Numbers = Numbers on the cards: The biggest the number on the card, the highest the priority for the card.
 - Instruction Marker = Pawns / Beads: Standard pawns of different colors, or beads can be used as instruction markers, one for each player.
 - Board = Face Down Cards: Before starting the game, use left over cards to build a board. 5x5 for a 2 player game and 6x6 for a 3-4 player game.
 - First player marker = Stack of left over cards: Take all the cards that won't be used in the game and use them to mark the first player for each round.

- **Setup**

Place a the board between the players. Each player takes their instruction marker and 3 instructions of each type. For each type, each player needs to have one instruction for each priority number (1 to 3). It's useful to order the instructions in front of each player so everyone can see which types and priority numbers everyone else has available.

Each player then chooses a unique instruction marker.

- **Definitions**

The "PRIORITY NUMBER" represents the preference of the instruction. Instructions with a bigger priority number are usually run first.

The "INSTRUCTION MARKER" is placed over one of the instructions on the board, which will be the one executed next for that player.

A "BLOCK OF INSTRUCTIONS" or "PROGRAM" is any number of instructions grouped together on the board (and separated from the rest of the instructions). Two instructions belong to the same block if it's possible to make a path from one to the other (only with orthogonal steps) that only passes through spaces occupied by other instructions. This way, two pieces that are adjacent to each other diagonally do not form a block.

The "ACTIVE BLOCK" is the block in which the player's instruction marker is currently on. Note that it is perfectly possible for a block to be the active block for two or more players. In this case, all the players sharing it, act on their turn as if it was just theirs, being able to affect it freely. (Hey, this is a game about computer viruses after all!!)

An "EMPTY BLOCK" is a block of instructions that doesn't have an instruction marker from any player.

- **Program Setup**

During the first phase of the game, the players build their initial programs in secret. They can use up to 5 instructions (of any type or priority and in any order) forming a line outside the board. Then they place their Instruction Marker on top of any instruction in their program. When all the players are done, they reveal their programs and place them on the board, on the row (from their point of view) closest to them. The program instructions must be placed together, with no spaces between any two of them, forming one block. No two programs (from different players) may touch each other at this point.

- **Game Play**

The player with the smallest program goes first. In case of a tie, the one with less total of priority numbers from their program's instructions starts, if there's still a tie, the player with the highest priority number just under their instruction marker starts. If there's yet another tie, decide randomly.

The game is played in rounds of as many turns as players playing (one for each player). After the first round, the first player for the next round may change.

After the first round, before a new round starts, look at the instruction each player is about to execute. If the current first player has the piece with the highest priority number (even if tied with other players) then the first player remains the same next round. If only one of the instructions is the highest, then the player owning that instruction will become the first player for this round. In case of a tie not involving the first player, the player closest to the first player (in clockwise direction) becomes the first player.

Using the First Player Marker to mark the first player for each round.

During their turn, a player has to execute the instruction their marker is standing on. Here's what the player can do for each type of instruction. Note that any instruction that has an instruction marker on top, cannot be affected in any way by any instruction executed:

- MOVE - The player must take one of the instructions from his active block and move it to any other empty space adjacent orthogonally to an instruction from the same block. This way, blocks can be divided or merged (even with active blocks from other players).

- WRITE - The player must take any one of his spare instructions (the ones in their play area outside the board) and place it on an empty space adjacent to an instruction from his active block. This way, blocks may also be merged (even with active blocks from other players).

- JUMP - The player must move his instruction marker to any other piece on his active block or to any empty block (not occupied by any other player).

- ERASE - The player can destroy an instruction if it is adjacent orthogonally to any ERASE instruction in their active block. An erased piece is "captured" by the player and they just place it back on their play area outside the board to be used again at a later time. Note that as long as there's one empty space adjacent to one of the ERASE instructions, the player can choose to target it and not actually erase any instruction, but if all the ERASE pieces from the active block are surrounded by other instructions (friendly or not) the player must target an actual one.

After executing the instruction, unless it was a JUMP instruction, the active player must move his instruction marker to an orthogonally adjacent instruction that doesn't have another instruction marker on top. The marker has to be moved, otherwise the active player is eliminated from the game (their instruction marker is removed from play but all the actual instructions remain on the board).

When this is done, play passes to the player to the left (clockwise). When all the players have taken their turn, after checking for a possible first player change, a new round starts.

- **Ending the Game**

If a player cannot move their instruction marker to a new instruction at the end of their turn, then they are eliminated. The last player remaining after everyone else has been eliminated is the winner.

There are a few cases when a player might not have much chance of winning (all their WRITE pieces erased) or no chance at all (all their ERASE pieces erased). In some of those cases, it might be possible for that player to force a stalemate if the other player's position is weak enough. If it's clear that one player will eventually win, players are encouraged to concede. If it's not clear, just play and in case of a stalemate (agreed by both players) the winner is the player with more spare pieces outside the board. In case of a draw, count the total number pips.

- **Inspiration for the Game**

The idea for this game came to me as I asked myself what icehouse game would a friend of mine (who works programming) like. In the 80s this friend of mine had some magazines from his father (Maybe Scientific American) and in one, they talked about computer programs fighting in a simulated memory. The programs were made with a simple language and they could modify the simulated memory trying to disrupt the opponent programs. I even made a couple of really simple versions of this concept for the Amiga computer back in the day. I just tried adapting this concept to an icehouse game, but instead of building the program and just letting it run (not much interaction there), I gave the player control of the instruction marker, deciding which instruction would be run next (simulating possible loops and if-then conditions that the program might have). This way, the players take the part of simple computer AIs :)

VIRUS FIGHT

VIRUS FIGHT

MOVE
WRITE
JUMP
ERASE

MOVE
WRITE
JUMP
ERASE



VIRUS FIGHT

Yellow	Green	Blue	Red
Orange	Cyan	Purple	White
MOVE	WRITE	JUMP	ERASE

VIRUS FIGHT

Yellow	Green	Blue	Red
Orange	Cyan	Purple	White
MOVE	WRITE	JUMP	ERASE

VIRUS FIGHT

Yellow	Green	Blue	Red
Orange	Cyan	Purple	White
MOVE	WRITE	JUMP	ERASE

VIRUS FIGHT



MOVE



WRITE



JUMP



ERASE

VIRUS FIGHT



MOVE



WRITE



JUMP



ERASE

VIRUS FIGHT



MOVE



WRITE



JUMP

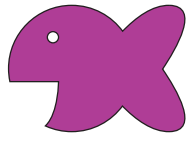


ERASE

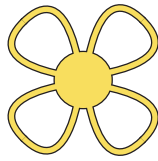
VIRUS FIGHT



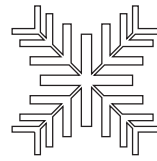
MOVE



WRITE



JUMP

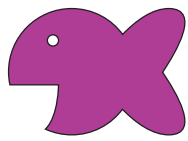


ERASE

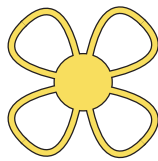
VIRUS FIGHT



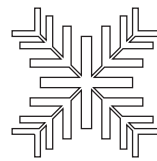
MOVE



WRITE



JUMP

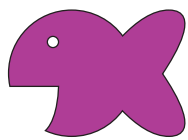


ERASE

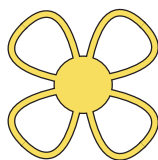
VIRUS FIGHT



MOVE



WRITE



JUMP



ERASE

VIRUS FIGHT



MOVE



WRITE



JUMP



ERASE

VIRUS FIGHT



MOVE



WRITE



JUMP



ERASE

VIRUS FIGHT



MOVE



WRITE



JUMP



ERASE