The Defrag game — *Computer Systems*

Developed by Shadi Ibrahim, SuYu and Tim Bell

(draft version for discussion and brainstorming – please post any ideas or improvements to tim.bell@canterbury.ac.nz)

Summary

The data stored on computer disks is divided up into "blocks", and often a file (such as a word processor document or video) is too big for one block, and so is split up into several blocks. However, the blocks aren't necessarily beside each other on the computer's disk, and if you are opening a file, it will take longer if the computer has to read the file from blocks spread around the disk. To avoid this problem getting out of hand, computers need to "defragment" the files, that it, try to move all the blocks so that they are together. This game gives students some experience with the process needed to try to achieve this.

Curriculum Links

- ✓ Computing: data storage
- ✓ Maths: Problem solving

Skills

- ✓ Matching colours and numbers.
- ✓ Problem solving

Ages

- ✓ 7 years and up (the level of difficulty can be adjusted)

Materials

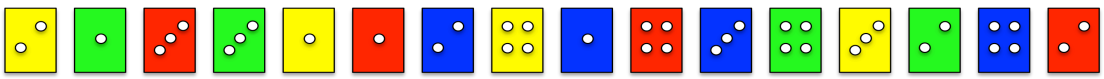
You will need:

- ✓ A set of cards for each player (either one set to share, or several sets so that the games can be played simultaneously) Typically about 4 cards each of 3 to 5 colours, with the cards numbered from 1 to 4.

Preparing the materials

The level of difficulty will depend on the number of cards in the set; around 16 cards (4 each of 4 different colours) works fairly well, but you could have up to twice as many colours, or twice as many cards of each colour.

The easiest way to make up the game is to buy index cards of different colours, and represent the numbers on them with a number of sticky dots, as below:

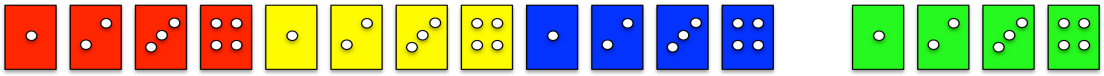


Other options are to cut up some coloured card, or to use coloured post-it notes with dots drawn on them.

Playing the game

Shuffle the cards and lay them out in a long line (if there's not enough room, you could put them in rows, and just remember that the end of one row connects to the start of the next row).

The goal is to rearrange the cards so that they are grouped in their colours, and the numbers of each colour is in order in the minimum number of moves, as in the following picture. Note that gaps between groups of colours are allowed (although there is a penalty).



The rules for moving are:

- You can only move one card at a time
- You have to move the card to somewhere that there is a space
- You can use any of the spaces before the first card and after the last card
- Each time you pick up a card and put it down, it counts as one move
- You cannot swap two cards directly (students should be able work out how to do a swap using three moves)
- At the end, you are allowed to have gaps between the groups of colours, but each gap counts as a penalty of two moves.

The game can be played in several ways:

- On your own, trying to beat your own lowest score
- Taking turns with an opponent, to see who gets the lower score
- With two card sets, and both players starting with the same random layout, to see who gets the lower score.
- With a whole class, possibly in groups, and each group tries to find the fastest way to solve the arrangement

There are some variations that add to the challenge and simulate a real defragmentation more accurately. As you add to the complexity, consider working with fewer cards!

- You could choose one colour which is "immovable"… those cards don't have to be rearranged, but they will make it harder to get all the other cards in a suitable order.
- You can have different numbers of each coloured card, some with only one or two, others with up to 5 or 6. This is more challenging because you need to work out which gaps are best for each colour
- You could have some gaps in the initial set of cards. To do this, shuffle some blank cards into the deck, and whenever a blank card comes up, leave a gap in the line of cards.

Follow up

- Discuss with the students what strategies they used to try to minimize the amount of movement.
- Were there arrangements of the cards that were particularly difficult?
- Particularly easy?

What's it all about?

Each coloured card represents a block on a disk, and the cards of the same colour represent the blocks that make up one file. The computer will perform best if the blocks for one file are all in a row. It's also desirable to have large gaps of empty space, rather than lots of small spaces on the disk, as this avoids either wasting the unused blocks, or having to split files up between the dispersed empty blocks.

The disk becomes fragmented when files are deleted or have their length changed, so after a long time the layout on the disk can be quite a mess. For example, suppose you're working on two word processor documents over a period of a couple of weeks. Every time one of the documents gets significantly larger, it will be allocated another block on the hard disk. But then the other document might get longer, and be allocated the next block. You end up with a blocks for the two documents mixed up with each other! Then, if one of the documents is deleted, there are random gaps on the disk. Soon it could be quite a mess.

Typically the blocks on a disk are 1 to 8 Kbytes. If the blocks are 4 Kbytes then a typical photograph of 1Mbyte would use about 250 blocks, a word processor file of 40 Kbytes would use 10 blocks, and a 3 minute video of 20 Mbytes would use 5000 blocks.

On some operating systems, the user has to explicitly "defrag" the disk to improve performance, and this can take several hours. Others try to be more clever about how they allocate the blocks in the first place. You can often get third-party software to perform the defragging, which might be faster or do a better job than the built-in system.

This problem becomes even more complex on "Virtual Machines" (used for "Cloud Computing"), which don't provide direct access to a disk, but simulate a computer and therefore simulate the access to the disk as well.

Computer Scientists have to work very hard to minimize fragmentation without using a lot of computing time to do it.

For more information about defragging, see the Wikipedia article at:
http://en.wikipedia.org/wiki/Defrag