

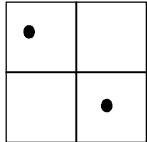
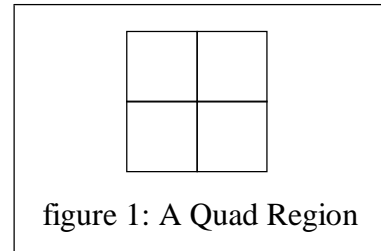
Smart Division : Dealing With Large Populations

Introduction

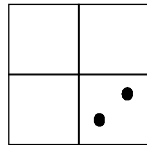
In some games, there are large numbers of objects scattered about. The problems occur when they may be scattered far apart and you simply cannot deal with searching through a large list of all the objects when only the nearby objects are relevant. How can the objects be divided into areas efficiently so that the searches can be more local in extent?

The Quad Region

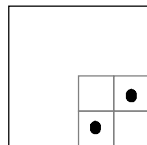
Assume a square region is divided into four equal parts as shown in *figure 1*. A good question would be, "How do you efficiently arrange objects in this region when an unknown quantity exists?"



A dot in two different quadrants is nicely divided to give each dot its own square.

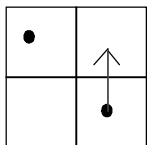


Two dots in the same quadrant, with no dots in adjacent quadrants would make the other squares wasteful. A bounding box can be set by a descend level L and a quadrant row, R , and column, C .

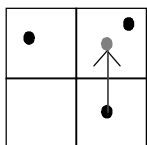


This box split into a quadrant can contain both.

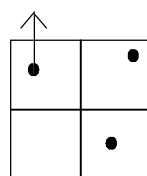
This can be done in a recursive fashion for many dots, evenly dividing them up and keeping track of them.



A dot can move into an unoccupied quadrant easily.

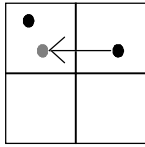


A dot can move into an occupied quadrant, with the requirement that a further subdivision operation occurs.



A dot can move outside the box, opening a can of worms.

In the case of movement outside the box, the largest containing box with quadrants should be found. If existing, add the dot to the proper quadrant, and put it into its own subdivision. If not in existence, a new box needs to be created and placed into an empty quadrant.



All three other quadrants can be left empty! This means the box is destroyed and the sub-box must be properly described using L , R , and C , replacing this box.

Steps to Reorganize

1. Shift right by $32-L$ (assuming 32bit word length) to mask
2. If move causes masking of R and C to equal another dot's masked R and C , either a descent must be done to increase the level L until they don't match, or produce a new *quad region* and separate the dots so that their R s and C s don't match after masked.
3. If move causes masking of R and C to not equal, ascend to see if they remain the same *quad region*. If not, ascend until a dot is found that does have the same *quad region* and make it, or do the above.