# GRAPHIC ENGINE INPUT/OUTPUT

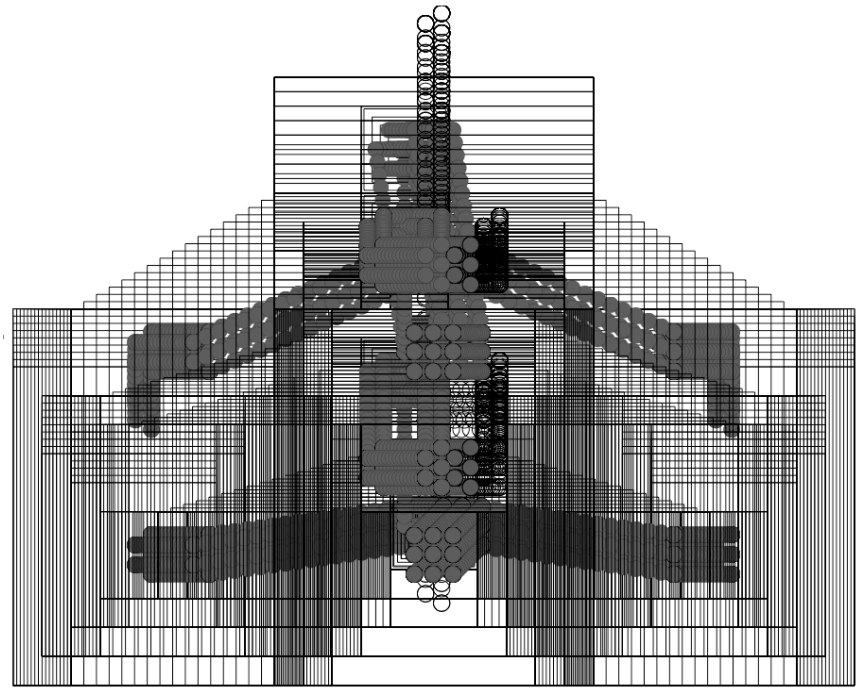
The reduction engine outputs an animation specification, i.e. a sequence of image specifications that can be tweened to create the illusion of smooth motion. In Mma I built the beginning and end points and then iterated over these points (using the number of steps wanted for timing and flow as a parameter) to build the animation steps.

The sequences below show the two base systems as examples of the generic transformations. Each base shows a comparable transformation sequence. The frame number is included with each picture.

I'll use 18–9+18 for BASE2, and 50050–5005+50050 for BASE10unit (um, so nesting depths align and 5+5 simulates binary 1+1).

Mma generates 298 frames, but for this example I'll just highlight the transition points. Both animations include a minor design error that makes the transformation sequence too long. And in general, the sequences are not cleanly recursive (but can be).

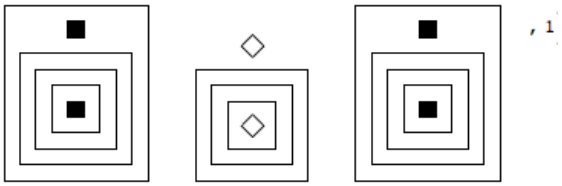
Here's an interesting image, it's all the transformations (for 50050–5005+50050 in BASE10dot) overlaid. I use this to determine the animation bounding box and to make sure that animation moves are centered.

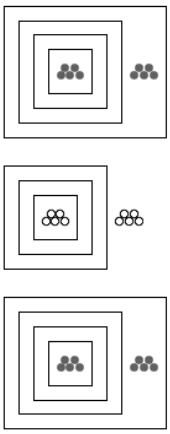


# ANIMATION SEQUENCES AND THE TRANSFORMATION RULES

The addition shown on the left is base-2 is 18–9+18. On the right, base-10-units follow the same reduction sequence; the sum is 50050–5005+50050.

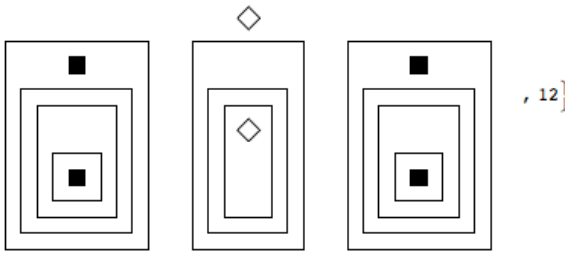
FRAME 1: The initial frame shows the transcription of the symbolic form into the iconic form. The small number in the middle is the frame number.

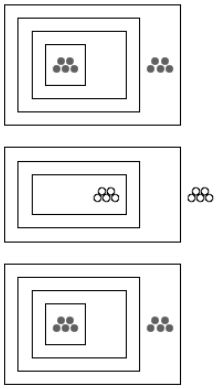


The only difference between base2 and base10is the shape of the unit(s) and the rotation. Any number of forms can add/subtract at the same time.

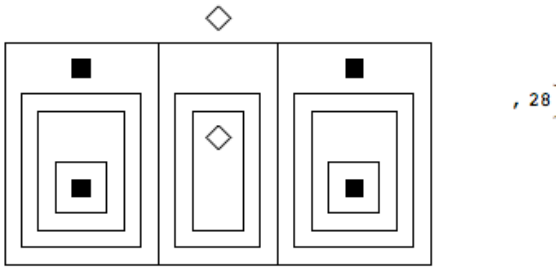
FRAME 12:

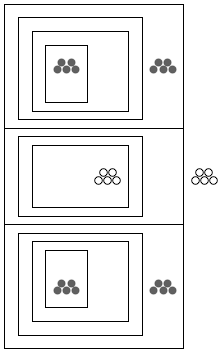


To MERGE boxes, they need to be aligned. The animation stretches the boxes so that tops align. Units also align at their depth of nesting from the top. Other boxes leave space for units when they exist.

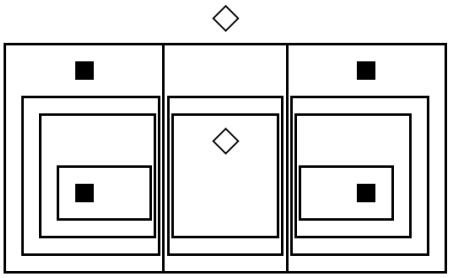
FRAME 28:

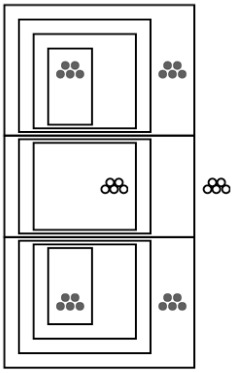


Entire forms move together in preparation for MERGING.

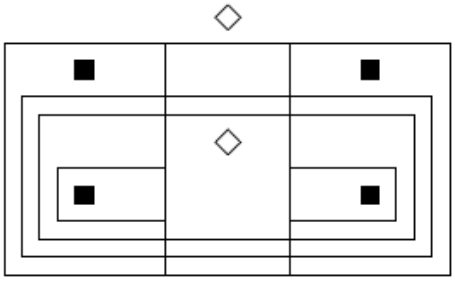
FRAME 33:



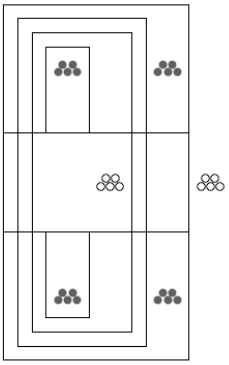


Center lines converge to illustrate joining boxes with common boundary.

FRAME 36:

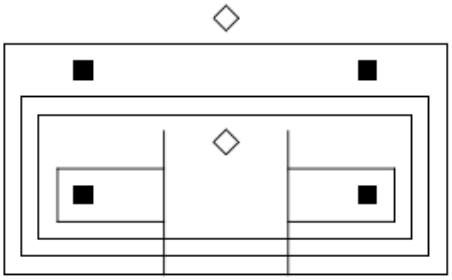


, 36

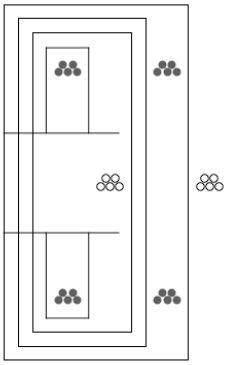


Boxes and their inner boundaries are aligned, ready to illustrate MERGE. Note that all levels merge at the same time

FRAME 40:

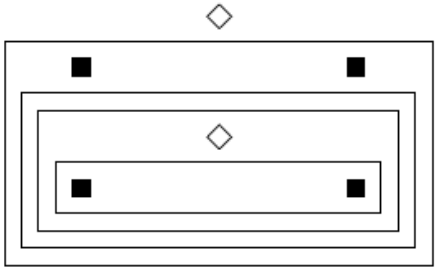


, 40

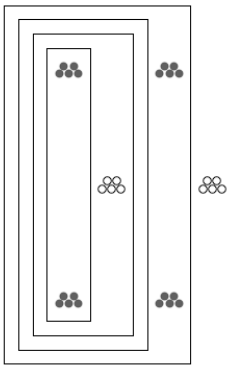


Middle boundaries recede revealing completely MERGED and aligned boxes. The boxes are rectangular so that this type of merging is easier to show.

FRAME 47:

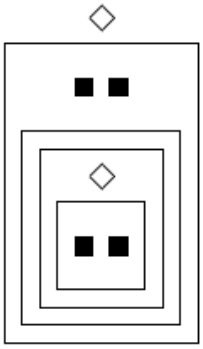


, 47]

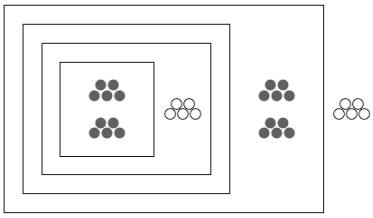


Now the addition/subtraction is complete, via the Additive Principle for each level of nesting.

FRAME 64:

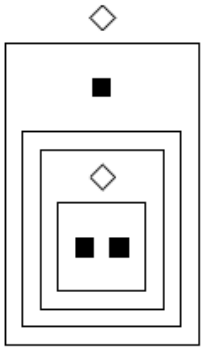


, 64]

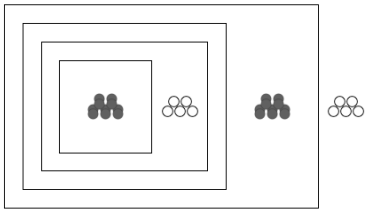


This is a tidying step, to normalize the shape, to make the next piece of code easier, and to make sure there's room for the next GROUPING step.

FRAME 72:



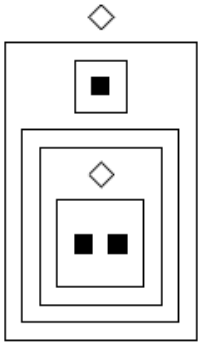
, 72}



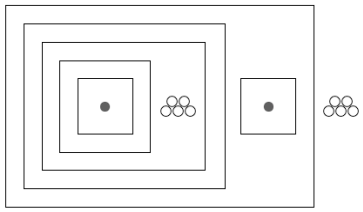
The two units in binary and the ten units in decimal move together to form a base group. This is midway in the GROUP transformation, just before they grow a boundary to express (●) in the respective base.

There's a divergence in the algorithms here (both are in error). When there is a negative unit at a shallower depth, the positive units should not merge because they just have to unmerge in order to "borrow" and cancel the outer negative unit.

FRAME 84:

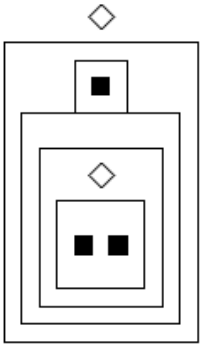


, 84}

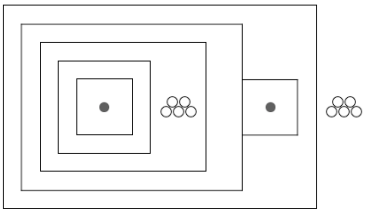


In base2, one pair has merged, in base10 both have merged. Actually neither should merge. There are lots of little planning gotchas that the prototype exposed.

FRAME 93:

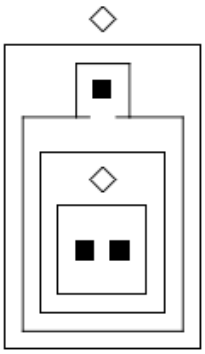


, 93}

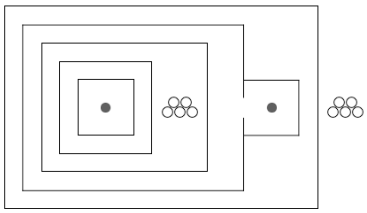


The newly formed boxes around a single unit are in the same space as another box, so the boxes MERGE. This animation is different than the first merging in FRAMEs 28-47. Now we just need get the boxed unit to the right level of nesting.

FRAME 99:

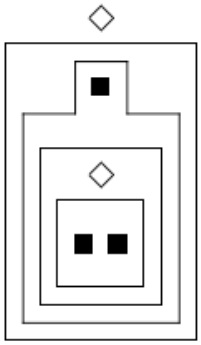


, 99}

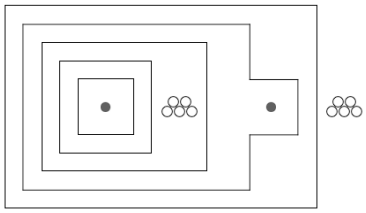


Part way during the MERGE sequence. The boxes are touching, now the boundary between them rolls back.

FRAME 105:

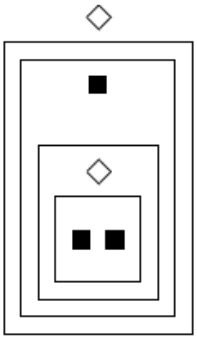


, 105}

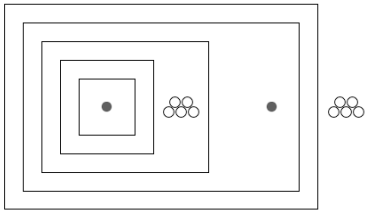


Boundaries fully MERGED.  Now its time to get everything tidy again.

FRAME 114:

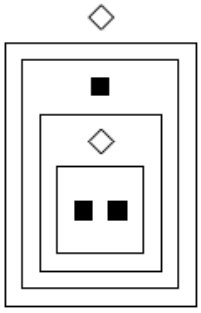


, 114}

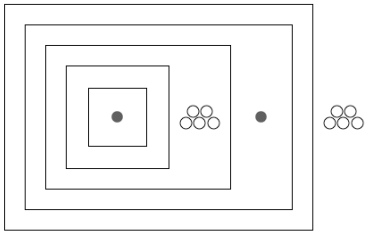


The irregular boundary moves up to form a rectangle.

FRAME 130:

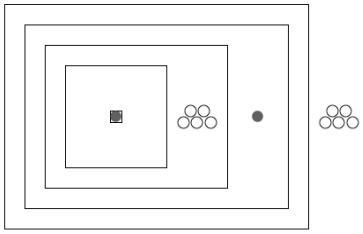


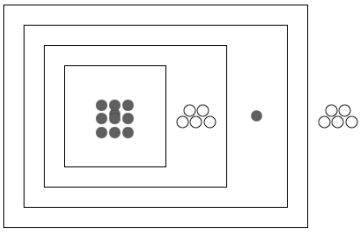
, 130}

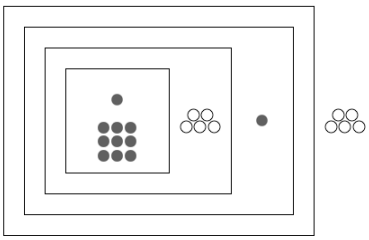


Tidy is finished.  Spacing between levels is consistent.

ADDITIONAL FRAMES FOR BASE10

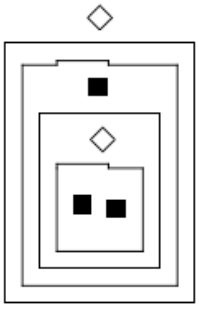




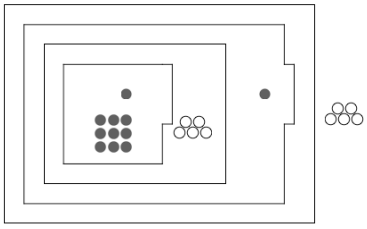


Base10 is beginning a single borrow (reverting to a prior form), the innermost box has shrunk down onto the innermost unit, which then converts into ten units. One unit splits off and prepares to migrate upwards to cancel the negative units at one level shallower.  All boxes expand upwards to make room for this.

FRAME 133:

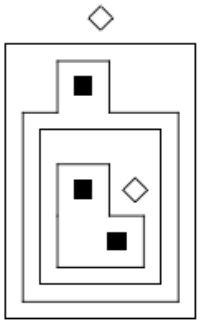


, 133}

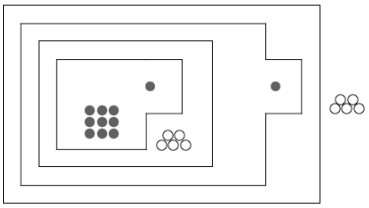


Both bases now begin a parallel borrow by migrating one positive unit up to the negative units to cancel them. Had the center ten units in FRAME 72 not GROUPED together, the additional frames to break them apart again would not have been necessary. In the parallel borrow, the negative units need to move out of the way of the box growing into their space. As well the entire figure needs to grow upward to make room for the new interior boxes.

FRAME 144:

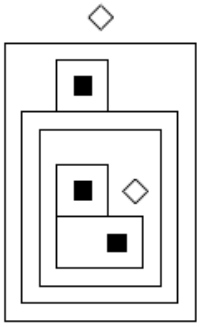


, 144}

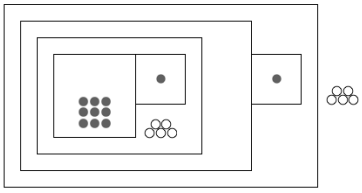


Migrating units are in position, now its time to separate their boxes. Below, the boxes separate (UNMERGE) and begin to shrink in preparation for converting into units.

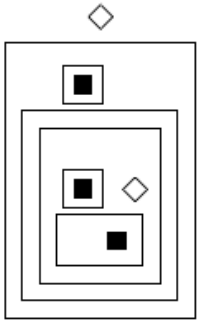
FRAME 148:



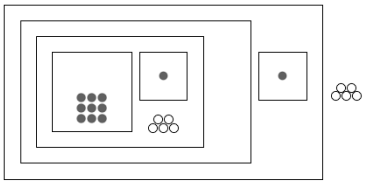
, 148}



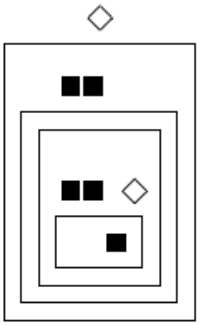
FRAME 150:

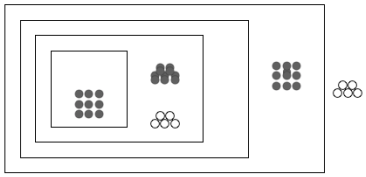


, 150}



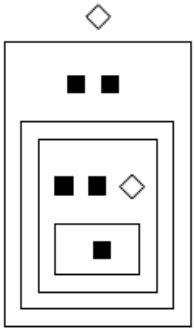
FRAME 159:

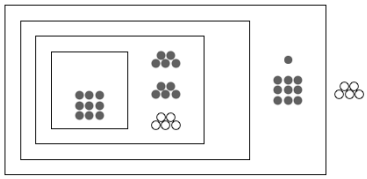




The pair of migrating boxes for each base convert into their base units, either 2 or 10. Now the innermost units can CANCEL. The outermost units still have another boundary to migrate outwards across.

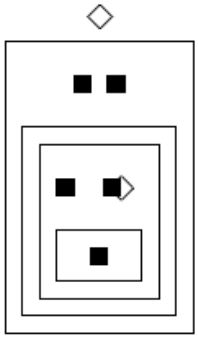
FRAME 176:

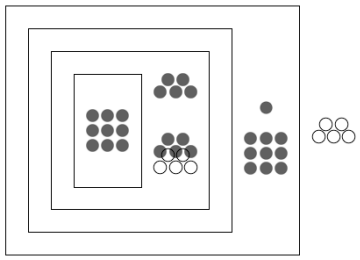




All the units get tidy, evenly spaced and aligned. Isn't quite done here.

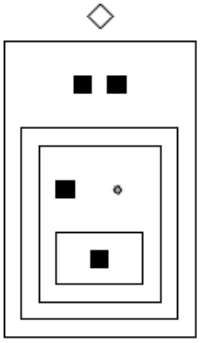
FRAME 187:

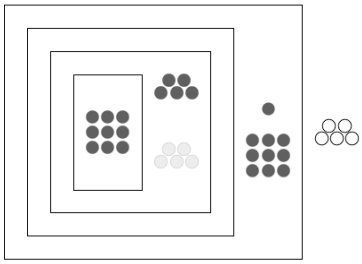




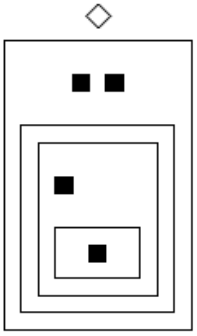
Tidied units begin to move into CANCEL position, positive directly on top of negative. Base10 fades away, base2 shrinks into nothing (below)

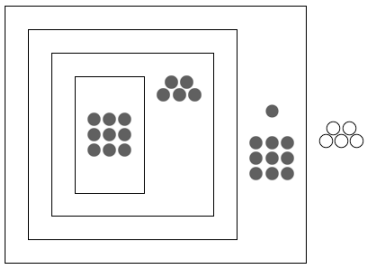
FRAME 193:





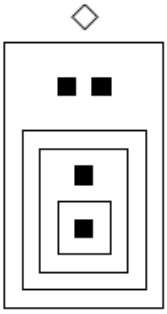
FRAME 198:

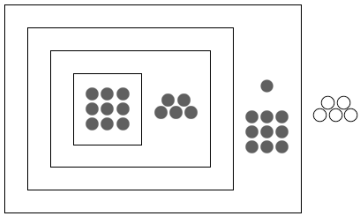




CANCEL units are completely gone.

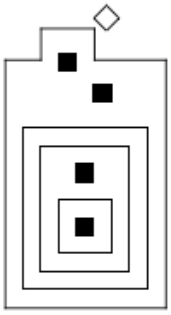
FRAME 215:

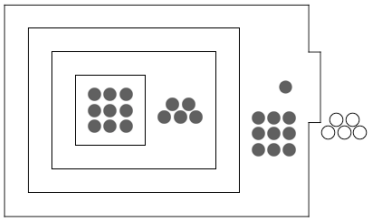




Time for another tidy. Boxes shrink, units center.

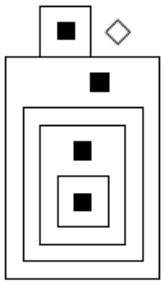
FRAME 222:

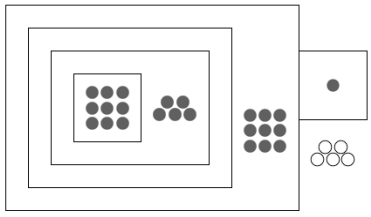




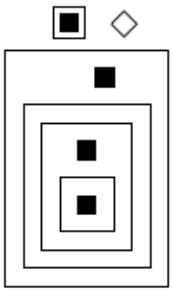
Time to begin the final migration outward to do outermost CANCEL. This could have happened in parallel to FRAME 176 CANCEL, or migrations could have finished prior to FRAME 176 beginning CANCEL. Both were hard to program given some modularity decisions I made earlier in the prototype build. Migration continues below, replicating actions of FRAMES 133-159.

FRAME 230:

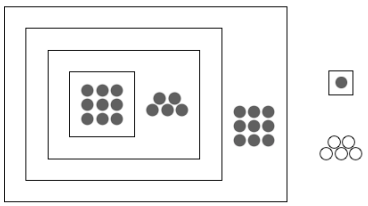




FRAME 234:

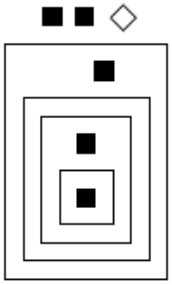


, 234}

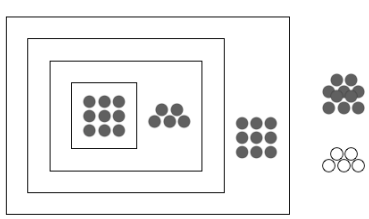


Migrated boxed units (always a single unit) have their box shrink as part of the animation of UNGROUPING.

FRAME 244:

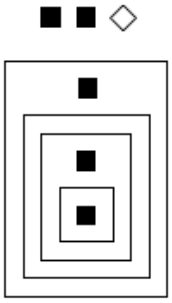


, 244}

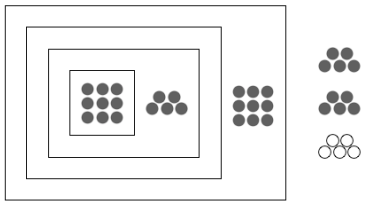


In base2 UNGROUP yields two units.  In the base10, UNGROUP splits exactly into the number of negative units needed for CANCEL.

FRAME 262:



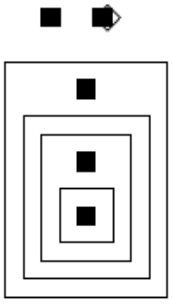
, 262}



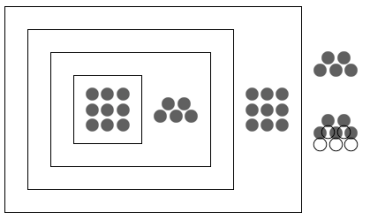
A little tidying first to get all units aligned.

The next three frames show the CANCEL process again. First positive and negative merge, then vanish, finally another tidy.

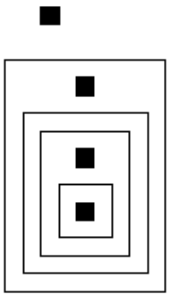
FRAME 271:



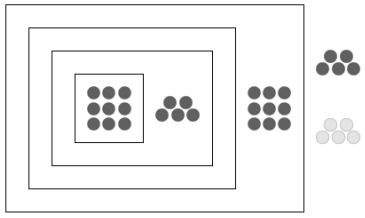
, 271}

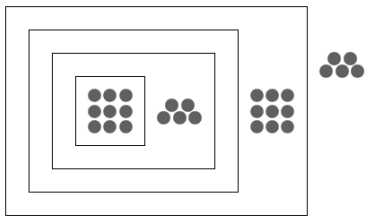


FRAME 278:

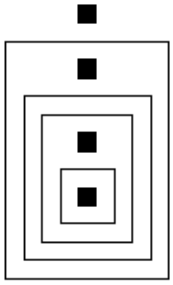


, 278

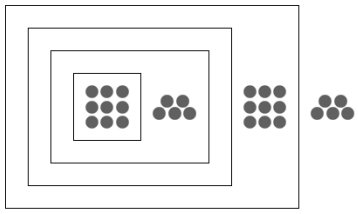




FRAME 298:



, 298



Final state.

==========