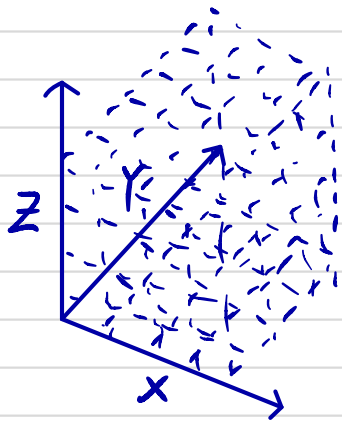
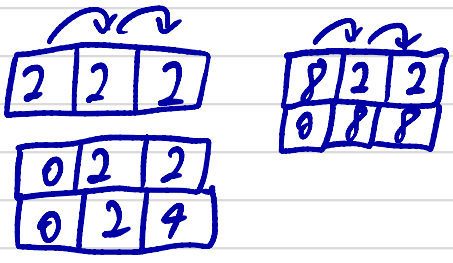
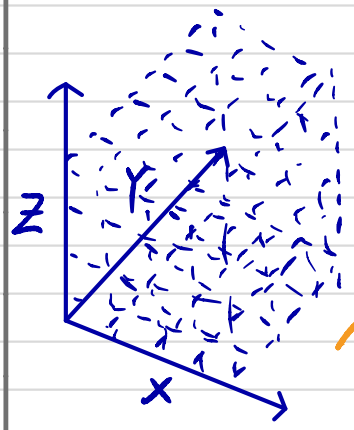


Requirements:
- Check if game is over ✓
  - no spaces
  - no adjacent tiles of same value
- Generate new game ✓
- Add new tile ✓
- Move all tiles in given direction & add tiles } see next page
  - find all tile translations
  - find all new tiles - which replace a tile
- Calculate new score
  - score += total value of new tiles made

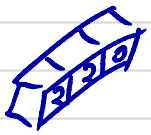
'find all tile translations'
- Starting from the direction tiles move towards, look for new tile
- use 'translations' to remove all 'start' points, before adding all 'finish' points





user choice

1) Split into groups of three tiles:



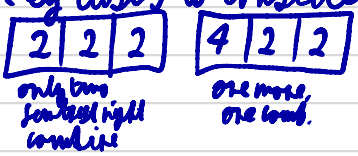
X 4

2) Stored as arrays made by appending nodes and moving in user choice (direction):

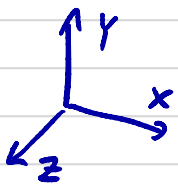
[(start: (x,y,z), finish: (x,y,z), val: 2, merged: false), ...]

3) Starting from the second highest index:
- check if tile can move higher, until max. index reached
- can move higher if
  - value at higher pos. is zero, or
  - value at higher pos. is the same
- this is programatically achieved by
  - using a list of tuples with values, starting & finishing coordinates, mergeBool
  - if a tile moves on zero: swap order in list and update finish
  - if a tile moves on the same value: update finish, remove from list, add to list 'removed', replace with zero, update value, mark merged - IF NOT MERGED
- extract translations looking for non-zero values with different start and finish, checking 'current' and 'removed' tiles.
- extract new tiles by looking for those marked 'merged'.

Key cases to consider:



In Xcode



Order of tasks:
- translate to merged regions to new coordinates
- remove so new tiles aren't removed
- add new tiles