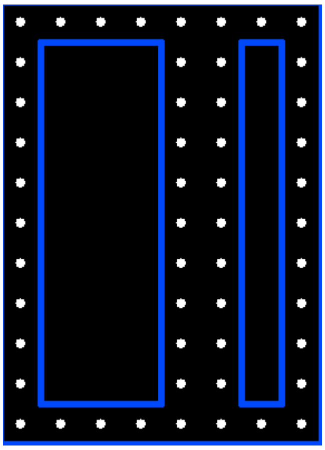
Pacman

Map generation

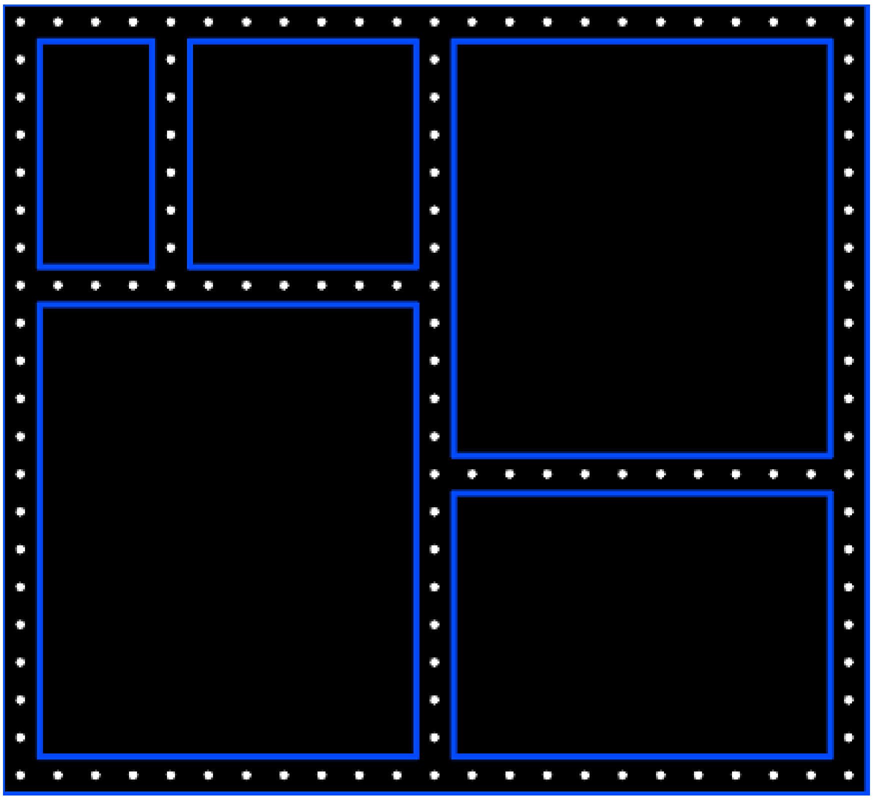
- The maps need to be at least 8x8 for the generation to be possible.

Algorithm

- The basic map consists of **zeroes representing walls**, and **ones representing free** (walkable) tiles
- The algorithm **starts with map full of walls** (zeroes) and it continuously creates walkable paths (ones) inside it
- Step 1:
    - We always create paths on the edges of map
        - This is done so that there aren't any dead ends or unwanted map configurations
- Step 2
    - We choose a random position on each of the four edges, and start filling that path with 1's
        - So for example, the top edge starts at the middle, and then goes all the way down, until it hits a wall
    - These are the four starting branches,
        - Each branch has a chance to create a subbranch for each tile it goes to
        - They can create a subbranch perpendicularly to the left or right (or both)
        - These subbranches can then create their own subbranches
        - The are two parameters:
            - MAX_BRANCHES
                - Maximum number of times a branch can branch out.
                - Branches inherit their parents branches_left minus 1.
            - BRANCH_CHANCE
                - Chance to branch out at each tile
                - It has a BRANCH_CHANCE to branch left, and independent BRANCH_CHANCE to branch right
- Since we don't want to have two paths neigbouring each other, we keep tabs of the branches created, and forbid to creation of new ones on the same/neigbouring row/column



unwanted neigbouring paths



Branch starts at the top in the middle. It than creates a subbranch on its right and than on its left. One subbranch also branches to its right

Branch starts at the top in the middle. It than creates a subbranch on its right and than on its left. One subbranch also branches to its right