`flgen generate` algorithms

Algorithm 1: *generateLandscape*

Input:

    Patch area distributions for N landscapes classes: $P = \{P_1, ..., P_N\}$

    Terrain: T

    Terrain dependency: $t_d \in [0, 1]$

    Minimum distance between two patches of the same class: $d_b > 0$

    Maximum number of trials for patch generation: $m_p > 0$

    Maximum number of trials for landscape generation: $m_l > 0$

Output: A raster L with N landscape classes satisfying P

$n_l = 0$;   `// landscape generation trials counter`

L = raster with all cells initialized to -1 ;   `// output raster`

$B = \{B_1, ..., B_N\} = \{\emptyset, ..., \emptyset\}$;   `// buffer cells for each class`

hasFailed = false ;

while $\underline{n_l < m_l}$ **do**

    for $\underline{i \in [1, N]}$ **do**

        for $\underline{j \in [1, \mathrm{NP}_i]}$ **do**

            $n_p = 0$;   `// patch generation trials counter`

            $p_j = \emptyset$;   `// patch cells`

            while $\underline{n_p < m_p \wedge p_j == \emptyset}$ **do**

                $p_j = generatePatch(i, L, B_i, \mathrm{AREA}_j^i, t_d, d_b)$;   `// generate patch`

                $n_p = n_p + 1$;

            end

            if $\underline{p_j == \emptyset}$ **then**

                hasFailed = true ;

                break ;   `// break the loop if patch generation failed`

            end

            $L_x = i$ for all $x \in p_j$;   `// write to output matrix`

        end

        if $\underline{\text{hasFailed} == \text{true}}$ **then**

            break

    end

    if $\underline{\text{hasFailed} == \text{false}}$ **then**

        return L ;   `// generation was successful`

    end

    $n_l = n_l + 1$;

end

return $\emptyset$;   `// generation has failed`

Algorithm 2: $generatePatch$

Input:

    Class index: i Landscape raster: L

    Class buffer cells: B_i

    Patch area: AREA_j^i

    Terrain dependency: $t_d \in [0, 1]$

    Minimum distance between two patches of the same class: $d_b > 0$

Output: Patch cells if generation was successful, else empty set

$p = \emptyset$;

$\text{current} = randomElement(\{c \in L \mid L_c = -1 \wedge c \notin B_i\})$;

$p = p \cup \{\text{current}\}$;

$n = 1$;   `// current patch area`

while $\underline{n < \text{AREA}_j^i}$ **do**

    $\text{adj} = \bigcup_{c \in p} adjacentCells(c)$;   `// retrieve adjacent cells`

    $\text{available} = \{c \in \text{adj} \mid L_c = -1 \wedge c \notin B_i\}$;   `// only retain available ones`

    if $\underline{\text{available} == \emptyset}$ **then**

        return $\emptyset$;   `// generation has failed`

    end

    $\text{next} = randomElement(filter(\text{available}, T, t_d))$;   `// get next patch cell`

    $n = n + 1$;

end

$\text{patchBuffer} = \{c \in \bigcup_{i \in p} buffer(i, d_b) \mid L_c = -1\}$;   `// ` d_b`-wide buffer`

$B_i = B_i \cup \text{patchBuffer}$;   `// fill ` B_i ` with newly created patch buffer`

return p ;