

## Random Ticks

Un *chunk* es una porción del mundo representada por un prisma rectangular de dimensiones  $16 \times y \times 16$ , donde  $y$  corresponde a la altura y toma valores en el rango  $y \in \{-64, -63, \dots, 0, \dots, 318, 319\}$ . Cada *chunk* se encuentra dividido en *subchunks*, cada uno con una altura fija de 16 bloques. Dado que el rango de  $y$  abarca 384 unidades ( $320 - (-64) + 1 = 384$ ), el número total de *subchunks* por *chunk* es:  $\frac{384}{16} = 24$ .

Cada *subchunk* es, a su vez, una matriz tridimensional de dimensiones  $16 \times 16 \times 16$ , lo que equivale a un total de  $16^3 = 4096$  bloques por *subchunk*.

Cada segundo el juego procesa 20 *game ticks*, lo que equivale a un tick cada 0.05 segundos.

Durante cada *game tick*, para cada *subchunk* tenemos 3 *random ticks* en una posición  $(i, j, k)$  del mismo, donde  $0 \leq i, j, k < 16$ , llamaremos a estas posiciones  $\alpha_1, \alpha_2, \alpha_3$ .

La probabilidad de que un random tick afecta  $\alpha$  son de  $\frac{1}{4096}$  para cada random tick, y dado que tenemos 3 random ticks por game tick tendremos que

$$P(\alpha) = 1 - \left(1 - \frac{1}{4096}\right)^3 = 0.0007322430756175891$$

Si  $\alpha$  es afectada por un *random tick*, se genera un cambio de estado. Más adelante analizaremos el impacto de esto.

## Mecánica de crecimiento de StemBlock

Solo nos importa el caso en que `AGE = 7` y `mayPlaceOn = True`, no me voy a gastar en explicar como llega ahí

### Condiciones necesarias para `getFruit` (dado `StemBlock AGE = 7`)

Una vez tenemos que `StemBlock AGE = 7`, las condiciones necesarias para que `AGE = 8` y por lo tanto crezca un melón (`getFruit`), dado el `StemBlock`  $\rho = (x, y, z)$ :

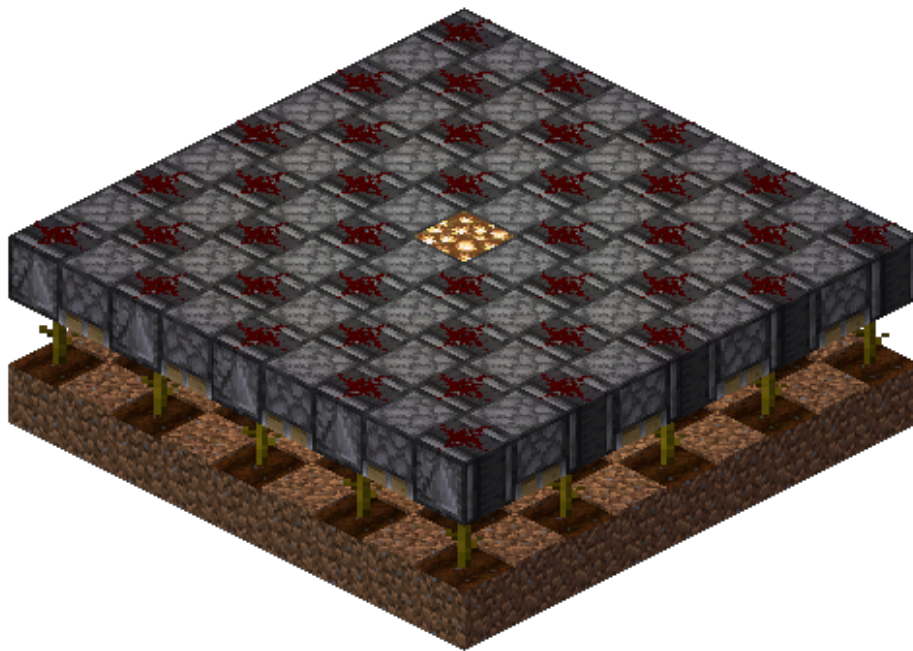
- $\rho$  es afectado por un random tick
- Nivel de luz en  $\rho \geq 9$ .
- Es aire  $(x + 1, y, z) \vee (x - 1, y, z) \vee (x, y, z + 1) \vee (x, y, z - 1)$ , esta posición la llamaremos  $(x', y', z')$
- $y' - 1$  es un bloque etiquetado como DIRT
- En el caso de que más de una posición cumpla lo anterior, selecciona una al azar.



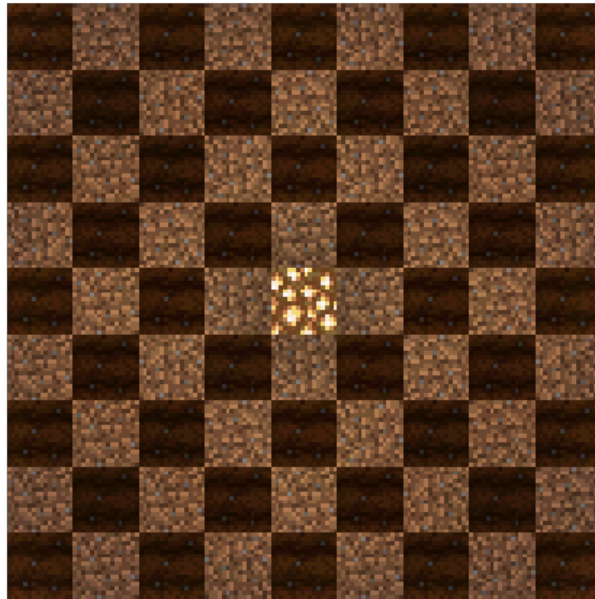
Figura 1: Ejemplos de configuraciones válidas

## Modelo clásico

El modelo clásico es un patrón de 1 y 1 entre stemBlock y posición válida, donde



El mejor patrón posible es



Donde tenemos un total de 40 stemBlocks

Cuando se actualiza el stemBlock generando un melón, tenemos que esperar un total de 5 game ticks para que ese espacio sea elegible de nuevo.

**Vamos a simularlo!**