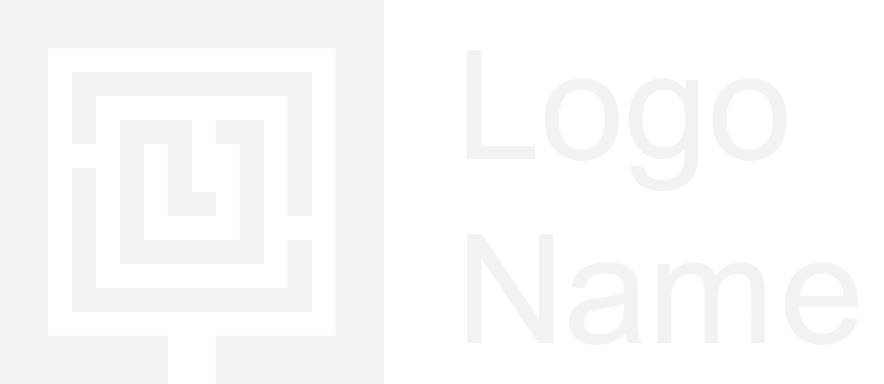


|  |
| --- |
| GDD |
|  |
| noviembre 6  Caja Negra |



# Intereses y propósitos generales

|  |
| --- |
| ¿Qué es lo que realmente atrapa? Fuera de las mecánicas y diseño del juego, siento que la verdadera gracia se basa en el **multiplayer.** Es una realidad que a la vez esto condiciona el rendimiento (cantidad de datos transferidos) y tiempo del juego (tiempo de desarrollo), por lo que pienso que la mejor idea sea limitar el diseño a algo en **2D**.  Entrando un poco más en las **mecánicas** la idea es hacer un **RPG**, todavía no hay nada muy marcado sobre esto, pero sacaría ideas de Hollow Knight, Terraria, Rain World. |
| Arte Sabiendo que estamos limitados al **2D**, y teniendo en cuenta la falta de personal de nuestra empresa, caería directamente sobre el **pixelArt**. A continuación, dejo un par de ejemplos de lo que tengo en mente. |

<https://0x72.itch.io/microfantasy>

(en general arte de 0x72)

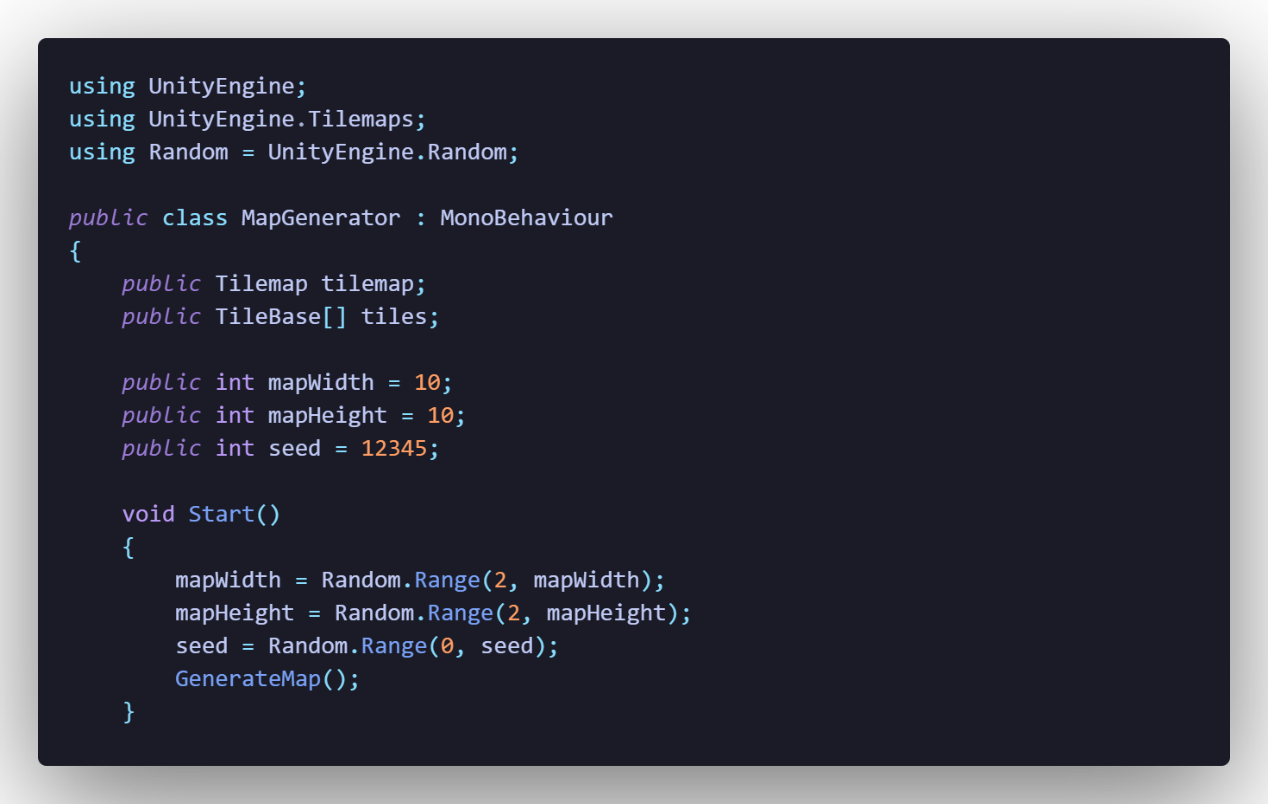
# Network

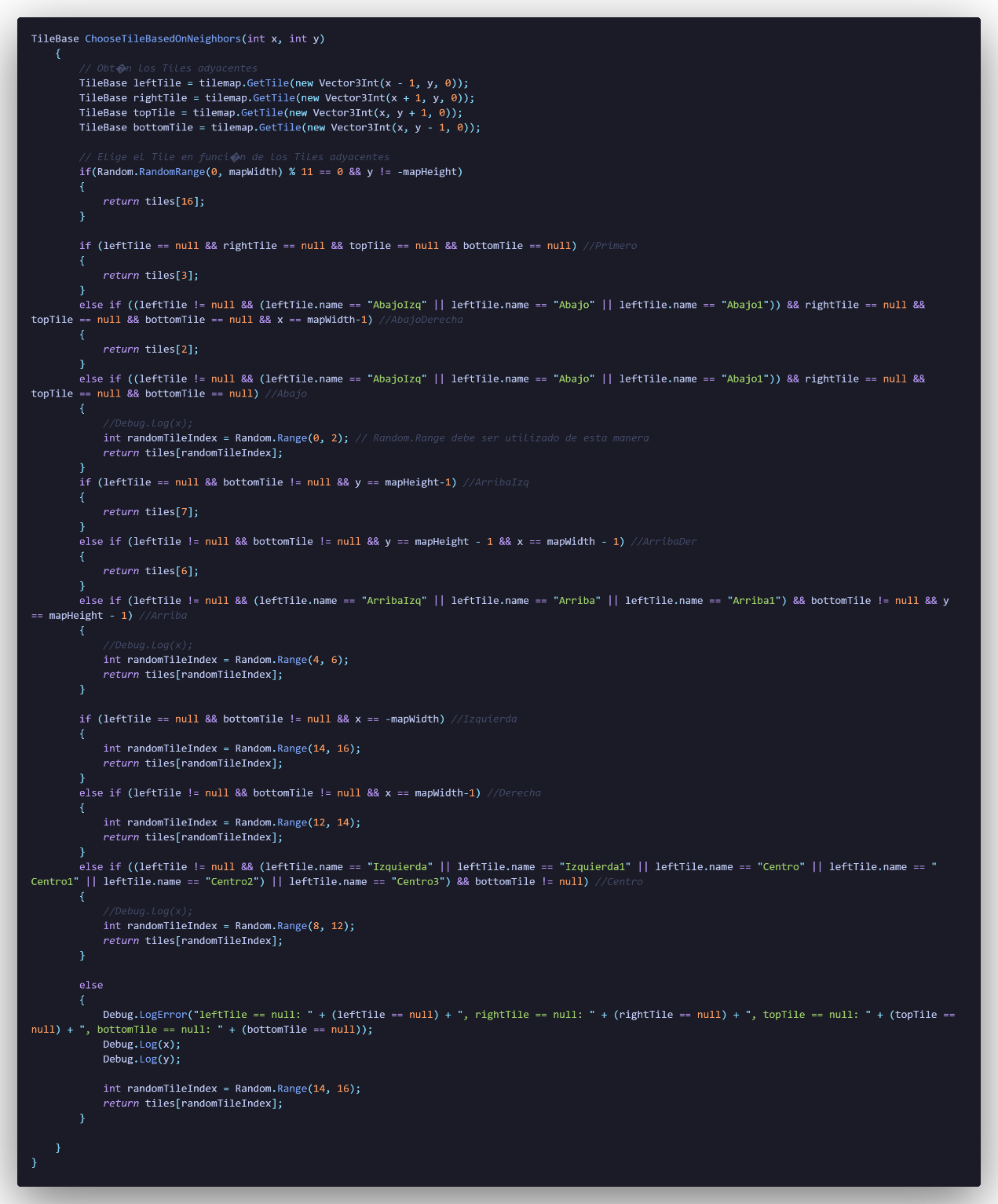
Vamos a usar steam, hay que ver como hacemos.

# Generación automática

Vamos a hacer un mundo top-down. A continuación voy a dejar un poquito de mi progreso en el tema:

## Intento 1:

1. Primero intente crear un cuadrado con tiles que se definan en base a la anterior teniendo en cuenta que estaba usando un for loop para poder colocarlas, también tuve en cuenta la idea de utilizar seeds, este fue el resultado:



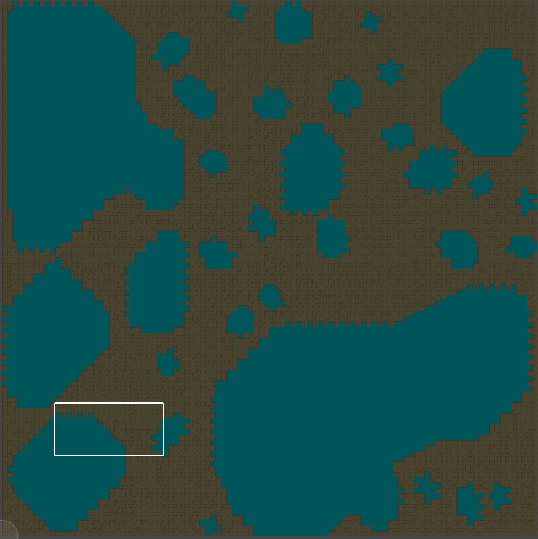
Este código tiene un montón de problemas, para empezar, es 0 polivalente, luego por como se ejecuta se hace “todo junto” lo que nos trae problemas a la hora de querer poner el siguiente tile, y por ultimo no nos deja modificar la forma del mapa. Así y todo, creo que es un buen acercamiento a lo que tratamos de hacer.

## Intento 2:

Vamos a hacer un par de cambios:

* 1. empecemos por usar noise.
  2. Algo importante a tener en cuenta es que cuando se hace un cambio hay que generarlo en otro grid, si no se rompe todo, aparte de esta forma podes acceder al resto de tiles (recordemos que por como estaba hecho el código anterior solo teníamos acceso al tile de la izquierda y al de abajo)
  3. también vamos a implementar el tema de Cellular Automata (mas info en <https://www.youtube.com/watch?v=t_HcBAO_Yas>)

ahora si quedo bonito!!



Este fue el resultado!!

La regla utilizada para el cellular Automata fue si x>4 -> x else y (donde x,y son tiles)

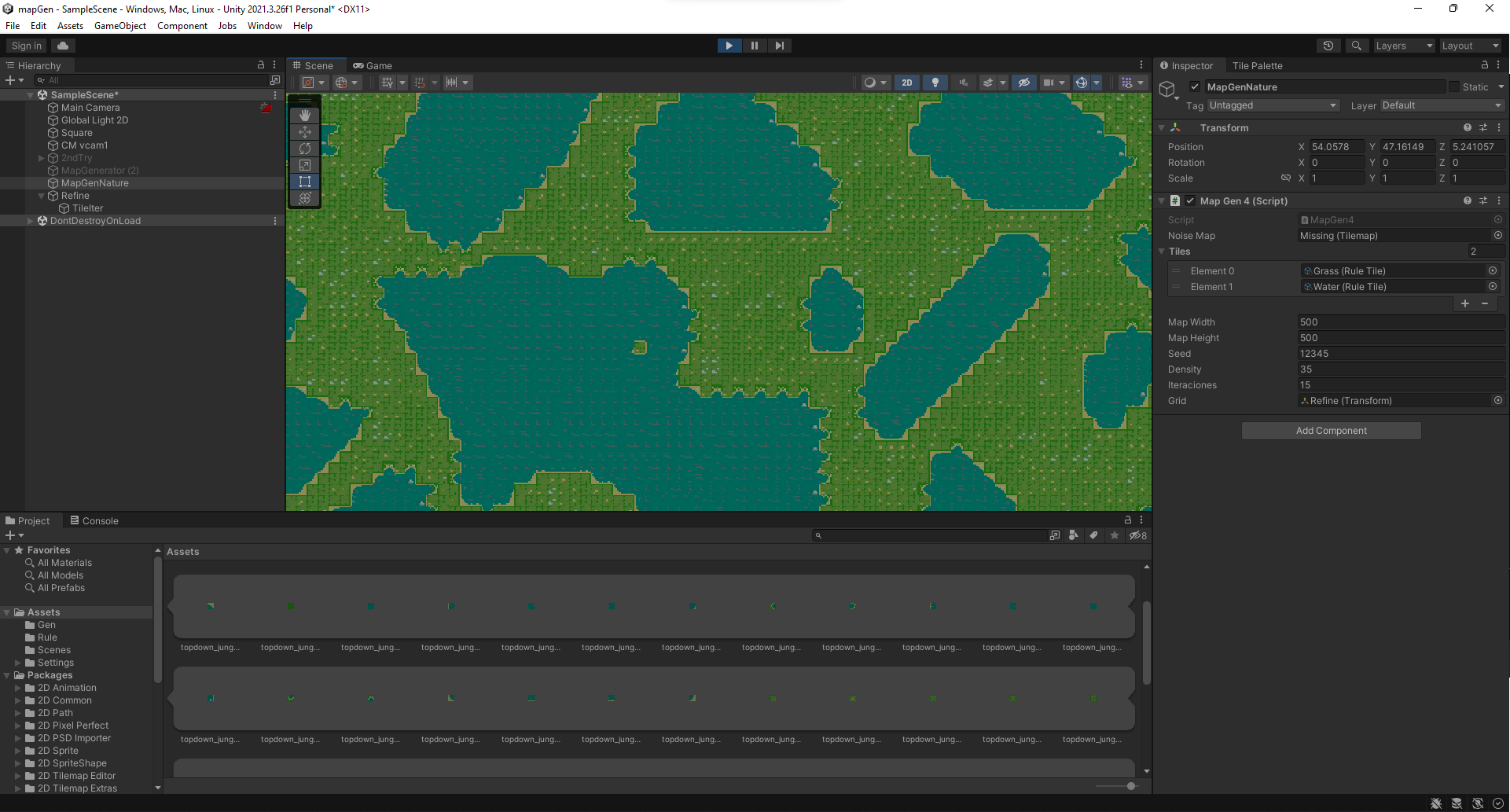
El código esta en <https://github.com/matifere/MapGen>, el script sería MapGen4

Ultimas boludeces a cambiar:

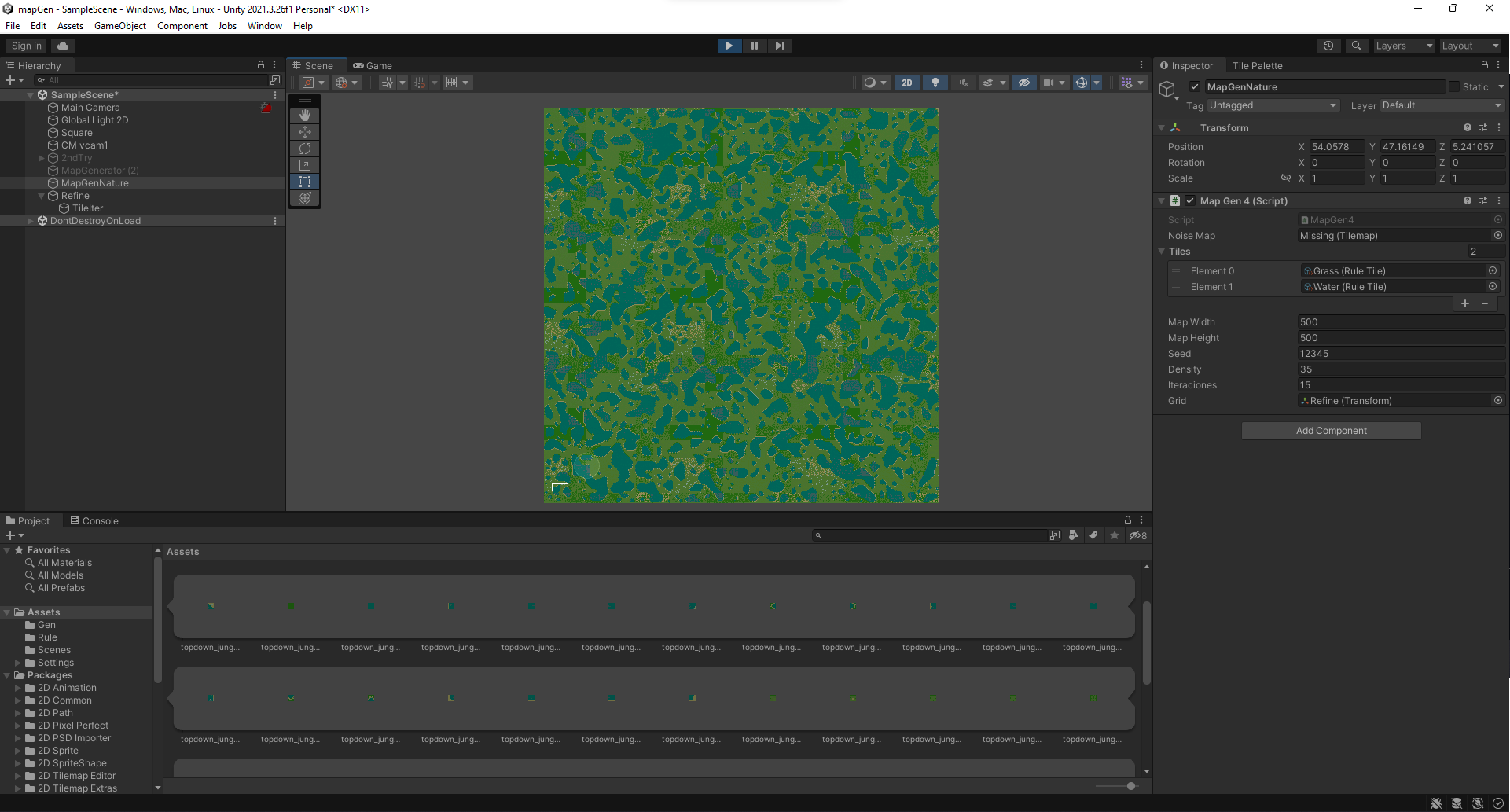
1. Parece que existe una herramienta (tilemap extras en el registro de unity) que coloca los tiles según reglas, lo que me ahorra como la mitad del código que hice antes.
2. Dentro de estas herramientas hay una que te permite agregar animaciones.

Acá el resultado:

De cerca:



De lejos:



Además, resolví como poner flores de forma aleatoria en el mapa, pero no lo voy a poner en este documento porque me parece bastante boba la implementación.

Algunos problemas que NO se pueden resolver con este tipo de generación.

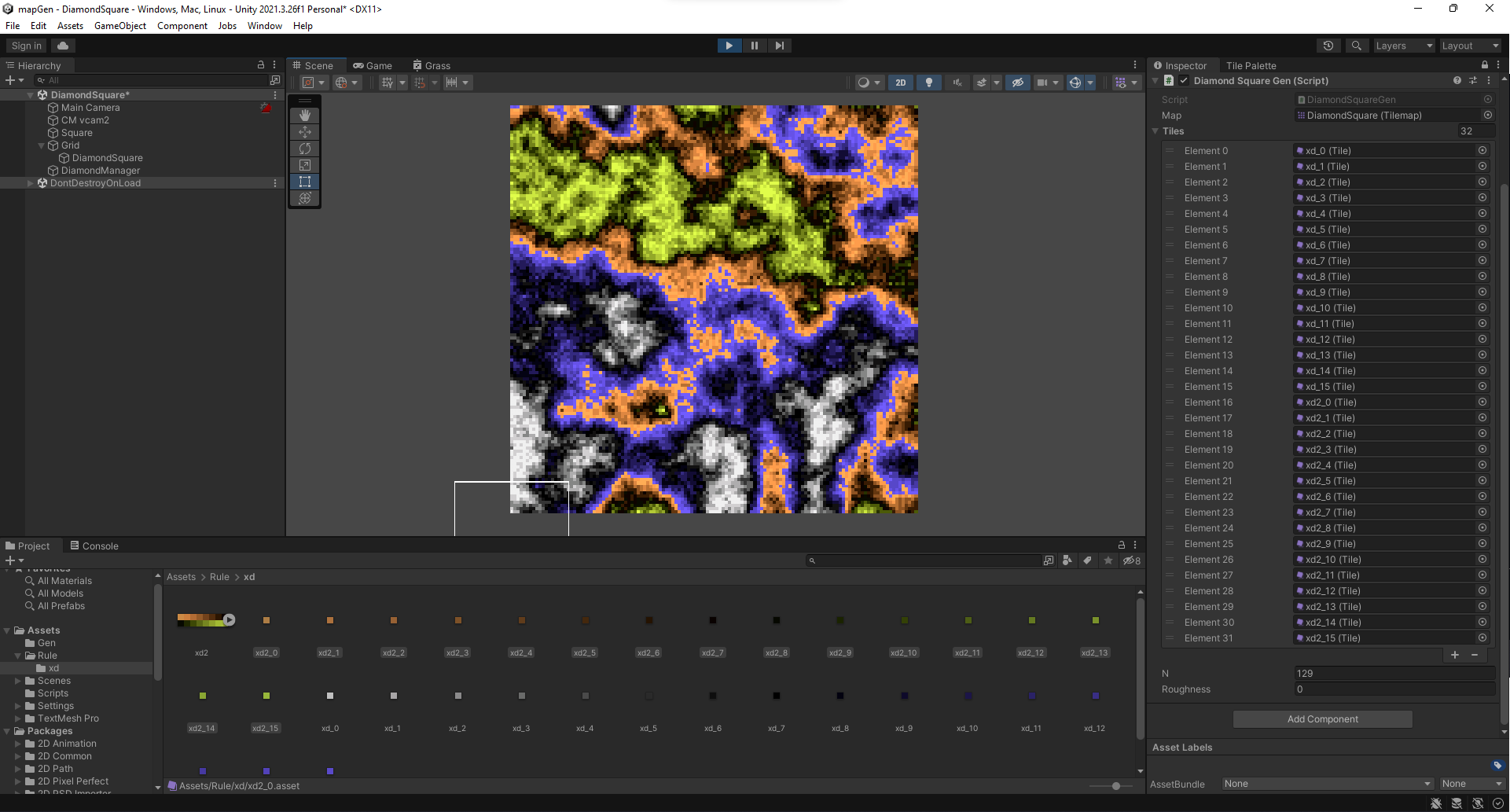
* No da la posibilidad de generar biomas
* Cuesta mucho generar el mundo

#### Usando Diamond Square

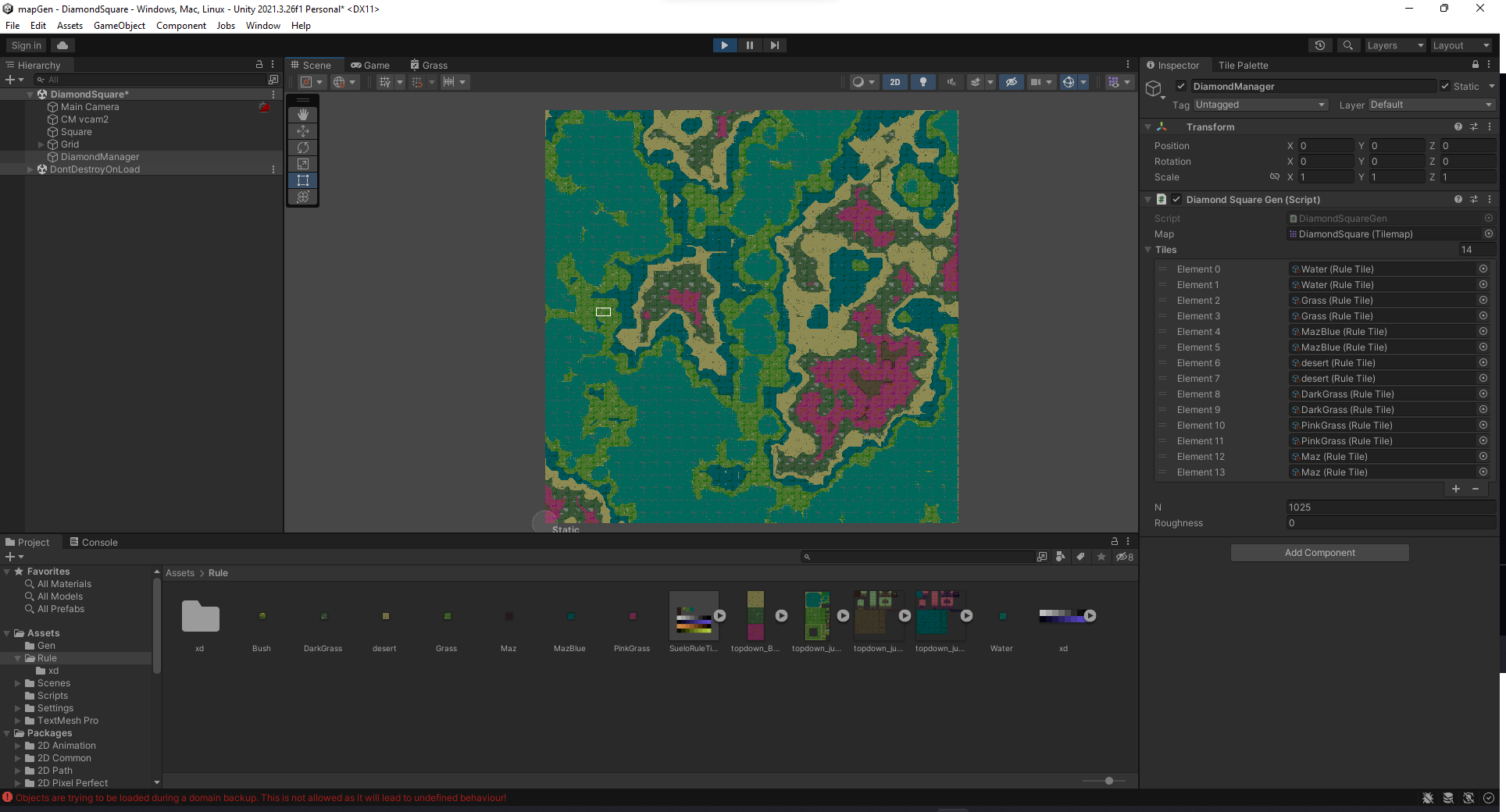
(ver <https://www.youtube.com/watch?v=4GuAV1PnurU>)

Para esto tuve que reescribir todo, pero logre que se ejecute mas rápido usando matrices y resolviendo los problemas anteriores, este es el resultado (sin tiles solo colores)

<https://github.com/matifere/MapGen> (DiamondSquare.cs)



Ahora con tiles:



Sigue sin convencerme, al ser una generación del estilo highmap no se siente muy aleatorio (medio que el jugador puede aprender a encontrar biomas por cercanía, por ejemplo, el agua siempre va a estar cerca de el pasto verde claro y el pasto verde claro siempre va a estar cerca del bloque azul, luego si el jugador encuentra un bloque azul “sabe” como llegar al agua ), aparte me parece medio feo :p

# Cosas por resolver

* Título
* Concepto principal (medio que ya está)
* Género (medio que ya está)
* Propósito y público objetivo
* Guion
* Mecánicas del juego
  + puntuación
  + guardado y carga
  + jugabilidad
  + niveles
  + intensidad
  + trampas
  + habilidades
  + recursos limitados
  + progresión del jugador
  + planificación de la batalla (o interacción con el oponente)
* Estados del juego
  + Pantalla de inicio
  + Pantalla de guardado
  + Multijugador
  + Pantalla de juego
  + Pantalla de configuración
  + Menú en pausa
* Niveles
  + Encuentro (el jugador entra en el nivel. Describe detalladamente qué ocurrirá.)
  + Objetivos
  + enemigos
  + ítems
  + personajes
  + Música y FX de sonido
* Progreso del juego
  + A través de un diagrama de flujo muestra cómo evoluciona el juego y los distintos niveles. Si el juego no es posible ser representado por un diagrama hay que buscar la mejor forma de describir el progreso de manera visual.
* Personajes
  + nombre del personaje
  + descripción: física, si es humano, animal, fantástico… incluir “concepts”
  + antagonista
  + motivación del personaje del personaje principal y del antagonista
  + encuentro: cuándo y cómo aparece en el juego
  + habilidades: enlistar y describir
  + armas: enlistar y describir
  + Ítems: enlistar y describir
  + personajes no jugables
* Logros
* Música y sonidos
* Funciones dentro del equipo (medio q todos vamos a hacer todo igual)