

Procedural Content Creation

With Unity

Creación Contenido Procedural

Consiste en general niveles o elementos del juego por medio de código en lugar de hacerlo en modo manual.

Suele poseer un factor randómico pero controlado por algoritmos para generar un diseño particular.

Suele poseer parámetros que controlan el diseño e imponen límites o ajustan probabilidades.

Creación Contenido Procedural

De este modo es posible generar una gran variedad de contenido sin requerir para ello un esfuerzo lineal.

Creación Contenido Procedural

La contra es que posiblemente sea muy complejo generar un “orden” dentro del contenido que no delate su origen randómico.

Variantes

- Crear contenido en tiempo de edición
- Crear contenido en tiempo de ejecución

Contenido en Tiempo de Edición

Unity nos permite generar contenido en tiempo de edición en una escena.

Podemos aprovechar esta posibilidad para luego realizar retoques o una decoración final de modo manual. Es decir, nos deja la posibilidad de sumarle una acción manual al contenido generado proceduralmente.

+ Aprovechamos al 100% las bondades del editor de Unity.

Contenido en Tiempo de Edición

- La desventaja de esta estrategia es que el juego pesará más en Mb dado que los niveles estarán descritos en su totalidad dentro del mismo cuando este sea distribuido.

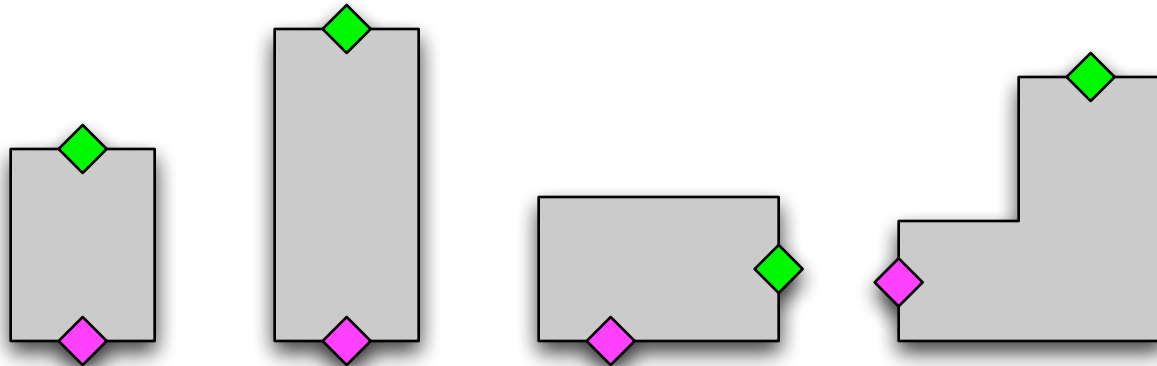
Contenido en Tiempo de Ejecución

- + El juego será más liviano (menor tamaño) dado que la ubicación de elementos en un nivel se especifica en runtime.
- + Es posible tener infinitos niveles en un juego.
- No utilizamos el editor de Unity dado que el contenido se genera 100% en runtime.
- No es posible controlar la experiencia del jugador al 100%.

Estrategia para la creación procedural de niveles

Una manera de facilitar la creación de niveles es la siguiente:

- Crear chunks o pedazos de nivel interconectables.

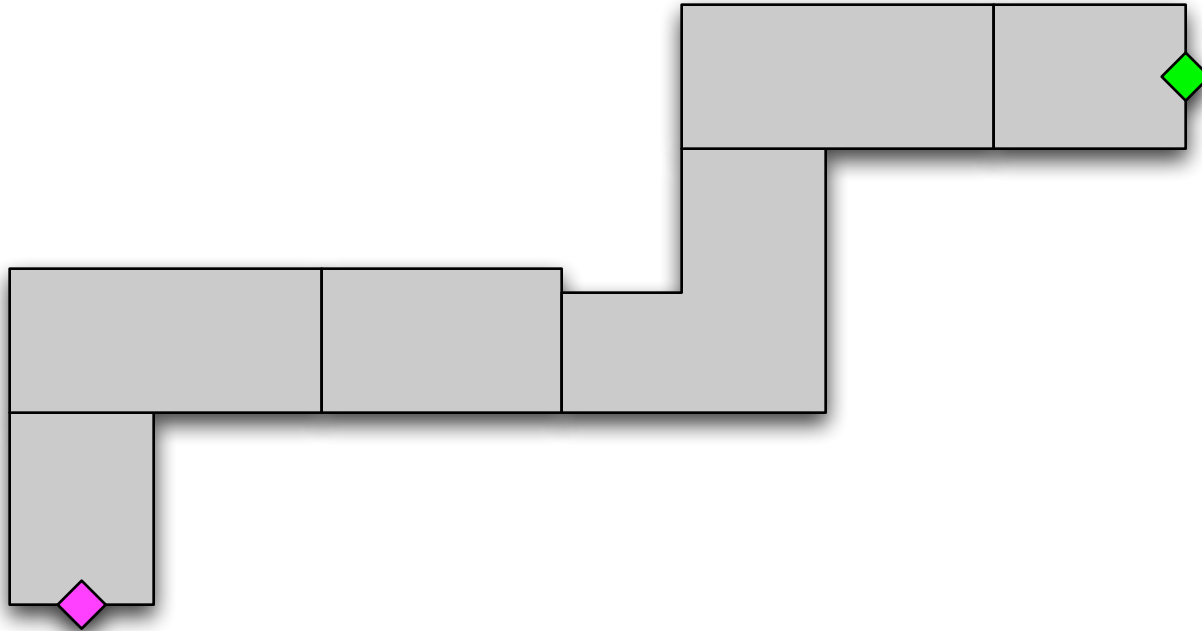


Estrategia para la creación procedural de niveles

[Estos chunks deben contener info de como encastrarlos entre sí correctamente.]

- Una función toma todos los chunks existentes, instancia N y los conecta entre sí.
- Parámetros controlan y balancean la forma y complejidad del nivel.

Estrategia para la creación procedural de niveles



Estrategia para la creación procedural de niveles

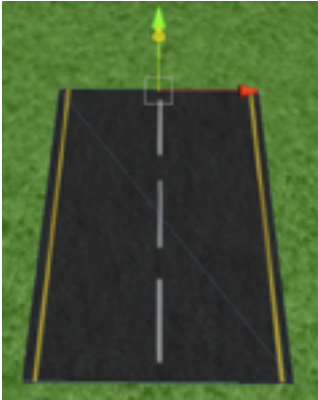
Complicaciones:

- Evitar que se generen estructuras superpuestas (si los giros podrían permitirlos).
- En ocasiones ciertos chunks no pueden encastrarse con cualquier otro.

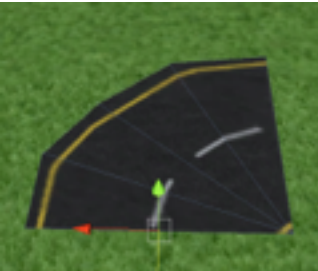
Creando una pista de carreras



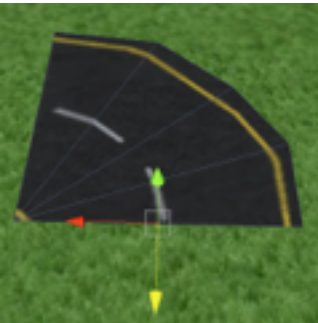
Creando una pista de carreras



Tramo recto



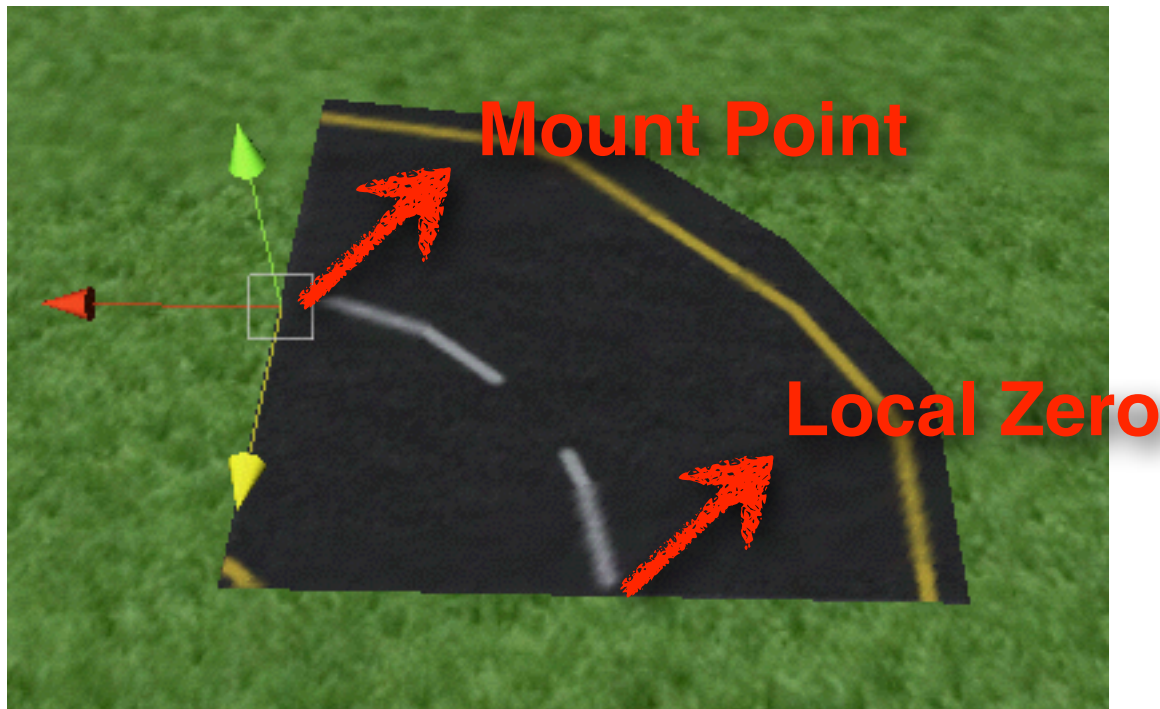
Giro a derecha



Giro a izquierda

Creando una pista de carreras

Crear prefabs a partir de estos chunks de pistas y luego agregar los mount points.



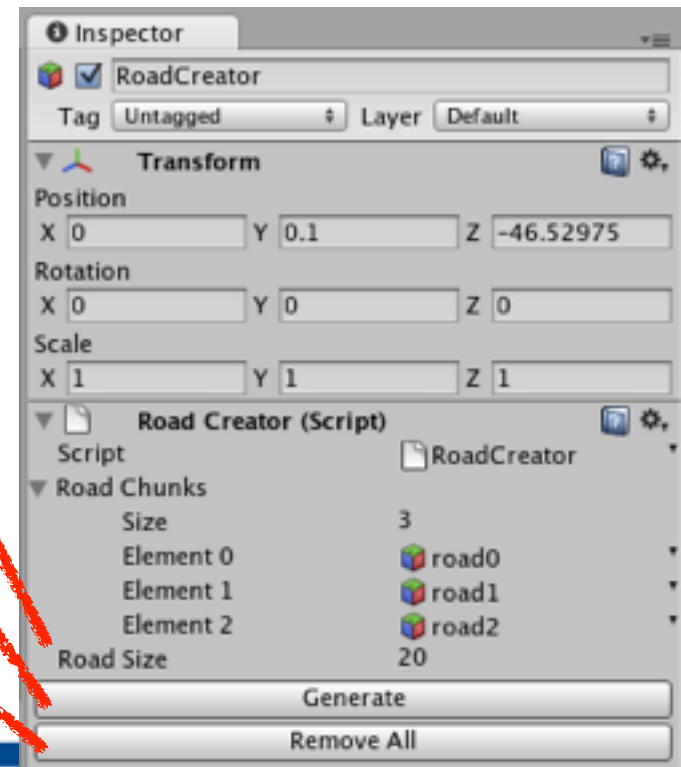
Creando una pista de carreras

Un mount point incluye una traslación y una rotación (si es que los prefabs deben ser rotados para permitir mayor diversidad).

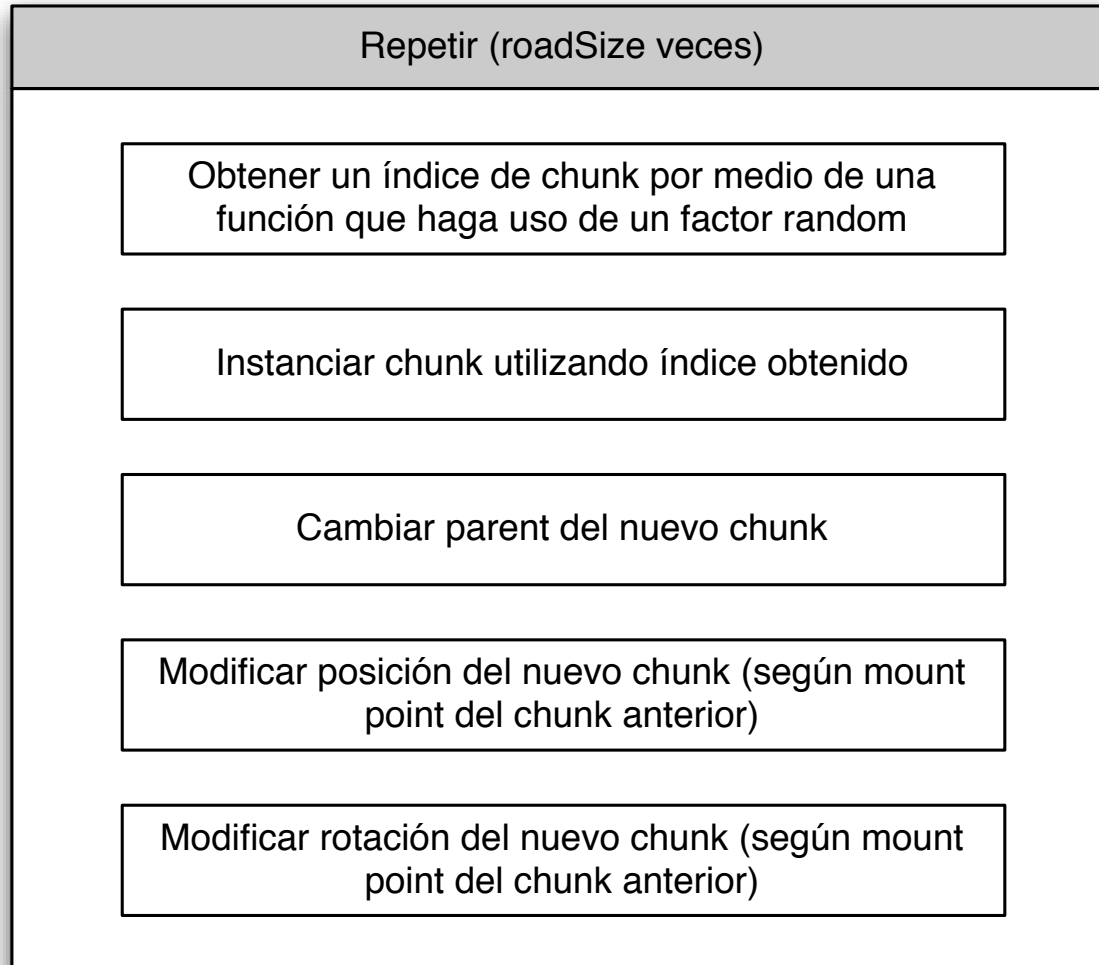
Creando una pista de carreras

Luego crearemos un game object que poseerá el script que realice la generación y será contenedor de todas las instancias de chunks creadas por dicho script.

Tamaño de la pista
Genera pista
Remover pista



Creando una pista de carreras



Creando una pista de carreras

Es clave que la función de obtención del índice del chunk no permita que la pista se cruce entre sí.

Para evitar cruces, es posible mantener una matriz con las posiciones ocupadas sobre el mapa o trazar un rayo para consultarlo sobre la escena misma.

En caso de cruce, se deberá realizar otra selección.

Creando una pista de carreras

Para realizar un cierre de la pista, es posible utilizar el siguiente algoritmo:

- Generar chunks al azar durante N chunks
- Cerrar luego la pista haciendo uso de un pathfinder desde el chunk último generado hacia el primero.