Generación procedural de mapas en videojuegos

Salazar Arias, Luis Santiago de Blass

Bravo Ocampo, David Aquiles

Flores Grados, Andrea Hilda

Narvaez Garriazo, y Cristel Margarita

Universidad San Ignacio de Loyola

FC-PREISF03Z01M(H): TEORÍA DE COMPUTACIÓN

ROBERTO JOSUE RODRIGUEZ URQUIAGA

2023

Índice

Índice de figuras

Índice de tablas

Agradecimiento

Queremos expresar nuestro más sincero agradecimiento a todas las personas que nos han brindado su apoyo durante el desarrollo de este proyecto. En primer lugar, deseamos agradecer a nuestro querido profesor, quien generosamente compartió sus conocimientos y nos enseñó a lo largo de todo el ciclo de desarrollo. Su guía y apoyo han sido fundamentales para el éxito de este proyecto. También queremos extender nuestro agradecimiento a nuestros amigos y familiares, quienes nos han brindado su aliento constante y comprensión en cada etapa de este proceso. Su apoyo incondicional nos ha permitido dedicar las horas necesarias y superar los desafíos que se presentaron en el camino. Por último, pero no menos importante, queremos expresar nuestro agradecimiento a todos los miembros de la comunidad de programación en línea. Sus valiosas contribuciones en los diferentes foros han enriquecido nuestro aprendizaje y nos han proporcionado una visión más amplia del mundo de la programación.

Introduccion

Los videojuegos son una forma de entretenimiento que combina elementos visuales, de audio e interactivos para crear experiencias lúdicas y narrativas. Para lograrlo, los desarrolladores de videojuegos deben diseñar y programar los escenarios, personajes, objetos y eventos que conforman el mundo virtual del juego. Sin embargo, este proceso puede resultar muy costoso y requerir mucha mano de obra, especialmente cuando se crea algo grande, variado y detallado. Una solución alternativa a este problema es la generación de etiquetas procedimentales, que implica el uso de algoritmos y reglas matemáticas para generar contenido de forma automática o semiautomática sin la necesidad de un diseño manual. Esta técnica permite crear mapas más grandes, más dinámicos y personalizados, ahorrando tiempo y recursos. Algunos ejemplos de videojuegos que utilizan generación de mapas procedimentales son Minecraft, No Man's Sky y Diablo. Sin embargo, la generación de mapas de procedimientos también implica ciertos desafíos y limitaciones, como la coherencia, la unicidad y la jugabilidad del contenido generado. Además, se debe considerar la detección de colisiones, que es el proceso de determinar si dos o más objetos del juego están en contacto o se superponen. La detección de colisiones es esencial para la jugabilidad y la física del juego, ya que te permite simular el comportamiento realista de los objetos, así como las interacciones entre ellos y con el jugador. La detección de colisiones puede ser muy compleja y exigente desde el punto de vista computacional, especialmente cuando se trata de objetos en movimiento o con formas irregulares. Por este motivo, se han desarrollado muchas técnicas y algoritmos diferentes para optimizar y simplificar este proceso, como cajas de colisión, árboles BSP o propagación de rayos. Estas técnicas varían según el tipo y tamaño del objeto, el nivel de precisión requerido y el rendimiento de la máquina. Este trabajo de investigación tiene como objetivo analizar las principales técnicas de generación de mapas procedimentales y detección de colisiones en videojuegos, así como sus ventajas, desventajas y aplicaciones. Para ello se realizará una revisión bibliográfica de las fuentes más relevantes sobre el tema, se explicarán conceptos

teóricos y se presentarán ejemplos prácticos.

Marco Teorico

La generación procedural de mapas se puede definir como el uso de algoritmos para crear contenido para videojuegos sin intervención humana directa. Esta técnica se basa en la definición de reglas, parámetros y funciones que determinan las características del mapa generado, como su forma, tamaño, estructura, color o textura. Estas reglas pueden ser deterministas o aleatorias, lo que implica que el mapa generado sea siempre el mismo o diferente cada vez que se ejecuta el algoritmo. Existen diferentes tipos y niveles de generación procedural de mapas según el grado de control que tenga el desarrollador o el jugador sobre el contenido generado. Se pueden distinguir cuatro niveles: generación offline, generación online, generación interactiva y generación mixta.

En este caso estamos utilizando técnicas y algoritmos para crear contenido de forma automática o semiautomática sin la necesidad de un diseño manual. En particular, estamos utilizando la técnica de generación offline, donde el mapa se genera antes de que el juego empiece y no cambia durante la partida.

Para generar los mapas, estamos utilizando diferentes técnicas y algoritmos. Combinando estas técnicas con otro algoritmo de pathfinding para buscar el "mejor mapa posible". Para mejorar la coherencia y la unicidad del contenido generado, estamos definiendo reglas y parámetros claros y coherentes que determinan las características del mapa generado. También estamos considerando la detección de colisiones, que es esencial para la jugabilidad y la física del juego, y utilizando diferentes técnicas y algoritmos para optimizar y simplificar este proceso. En resumen, nuestro trabajo se enfoca en la generación procedural de mapas en videojuegos utilizando diferentes técnicas y algoritmos para crear contenido de forma automática o semiautomática, y mejorando la coherencia y la unicidad del contenido generado.

Metodología

En el desarrollo de este proyecto, se utilizó una metodología que incluyó diferentes etapas, desde el diseño del estudio hasta el análisis de datos. A continuación, se detalla cada una de estas etapas:

Diseño del estudio: En esta etapa, se indagó información en papers y artículos para la generación de mapas. Entre las diferentes opciones consideradas, se decidió utilizar el algoritmo de "drunkard's walk"debido a su facilidad de uso. Además, se agregó una variante del algoritmo de búsqueda en amplitud ("breadth first search") para buscar el mapa más "óptimo". Para la lógica del juego en general, se tomaron como referencia algunos video tutoriales, utilizando la librería de Python llamada pygame para crear el juego.

Recopilación de datos: En esta etapa, se investigaron los papers relacionados con la generación de mapas procedurales. Se tomó la decisión de enfocarse en la generación de mapas en 2D, ya que era más seguro de programar en el nivel y tiempo disponibles. Durante la recopilación de datos, se realizaron resúmenes y lecturas rápidas para seleccionar el código del algoritmo de "drunkard's walkz el algoritmo de búsqueda en amplitud ("breadth first search"), ya que eran aplicables al proyecto. Por otro lado, se decidió descartar el uso del algoritmo de "perlin noise"debido a que no se ajustaba a las necesidades del proyecto.

Análisis de datos: En esta etapa, se desarrollaron bocetos e ideas para la creación de un videojuego en 2D. Esto llevó a analizar los papers y video tutoriales que se enfocaban en el desarrollo de juegos RPG de vista superior ("top down"). Gracias a este análisis, se logró construir e implementar el código necesario para la generación de mapas y caminos en el juego. **?**

Objetivos

Objetivo general

Analizar las principales técnicas de generación de mapas procedimentales y detección de colisiones en videojuegos, así como sus ventajas, desventajas y aplicaciones. Para ello, se realizará una revisión bibliográfica de las fuentes más relevantes sobre el tema, se explicarán conceptos teóricos y se presentarán ejemplos prácticos.

Objetivos específicos

Explicar la técnica de generación de mapas procedimentales y cómo se utiliza para crear mapas más grandes, dinámicos y personalizados en videojuegos. Describir la detección de colisiones y su importancia para la jugabilidad y la física del juego. Identificar los desafíos y limitaciones de la detección de colisiones, especialmente cuando se trata de objetos en movimiento o con formas irregulares. Presentar diferentes técnicas y algoritmos utilizados para optimizar y simplificar la detección de colisiones, como cajas de colisión, árboles BSP o propagación de rayos. Analizar las ventajas, desventajas y aplicaciones de las técnicas de generación de mapas procedimentales y detección de colisiones en videojuegos.

Diseño de estudio:

En esta fase, se investigó y buscó información en documentos y artículos para la generación de mapas. Se utilizaron el algoritmo "drunkard's walkƶ el algoritmo breadth first search. Se eligieron estos algoritmos por su funcionalidad y aplicabilidad en el contexto de los videojuegos 2D. Junto con estos algoritmos, se utilizó la librería pygame, que proporciona funciones y herramientas que facilitan la interacción con el desarrollo de juegos 2D, como la reproducción de sonido, la detección de eventos de teclado y la manipulación de gráficos.

Recopilación de datos:

En la recopilación de datos, se investigaron papers y artículos referentes a la generación procedural de mapas y la detección de colisiones. Se realizaron bocetos,

utilizando técnicas y conceptos involucrados en estos procesos. Durante esta estapa se descartó el uso del algoritmo de "perlin noise"debido a que no se ajustaba a las necesidades del proyecto. En vez de él se utilizaron los algoritmos de "drunkard's walkz el algoritmo de búsqueda en amplitud ("breadth first search"), ya que eran aplicables para el desarrollo de juegos 2D.

Análisis de datos:

Después de la recopilación de datos, se procedió a realizar un análisis exhaustivo y una discusión detallada de los hallazgos más relevantes encontrados en los papers seleccionados. Este análisis permitió obtener una comprensión más profunda de los conceptos teóricos y las técnicas más prometedoras para el proyecto. Se examinaron los diferentes enfoques y se evaluaron sus ventajas y limitaciones en relación con los objetivos y requisitos del proyecto.

Resultados:

Como resultado de arduo trabajo hemos logrado con éxito el objetivo de generar mapas de forma procedural. La implementación clave que nos ayudo a lograrlo fue en usar el algoritmo Kruska, este enfoque ha demostrado ser efectivo para producir entornos de juego dinámicos y variados. La combinación de Kruskal y pathfinding ha permitido superar los desafiós inherentes a la generación procedural, generando mundos virtuales cohesivos que ofrecen desafiós estratégicos y una experiencia de juego envolvente. Estos resultados sientan las bases para futuras mejoras en la generación de mapas procedurales en el ámbito de los videojuegos.

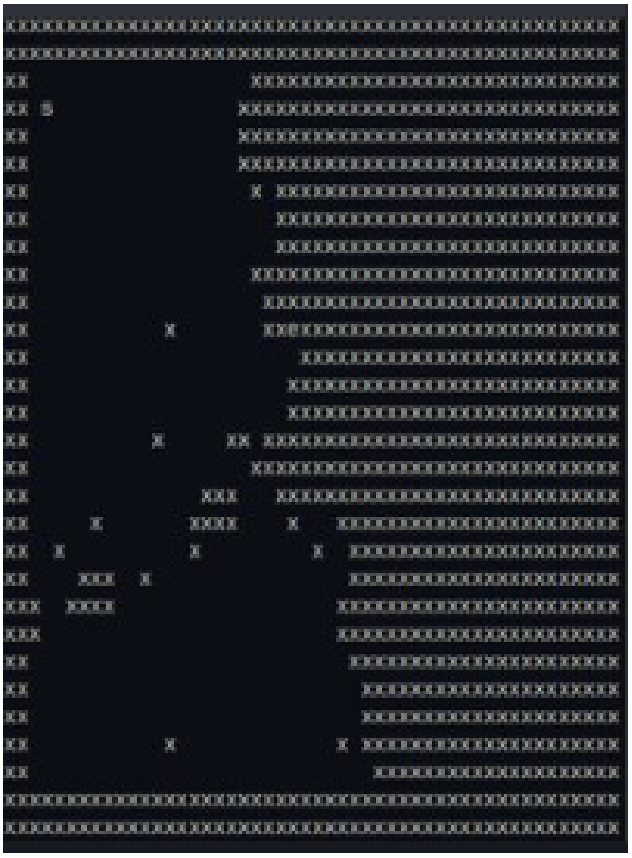

Figura 1

Primer Mapa Generado

Conclusiones:

En conclusión, este proyecto donde hicimos uso desde el diseño del estudios hasta la recopilación y análisis de datos, ha destacado la notable viabilidad y eficancia de la generación procedural de mapas, en particular, mediante la implementación exitosa de Kruskal y pathfinding, La integración de estas técnicas ha mostrado ser importante para superar los desafíos asociados con la creación automática de entornos de juego, brindando no solo eficiencia en el desarrollo, sino también una buena experiencia de juego para los usuarios.

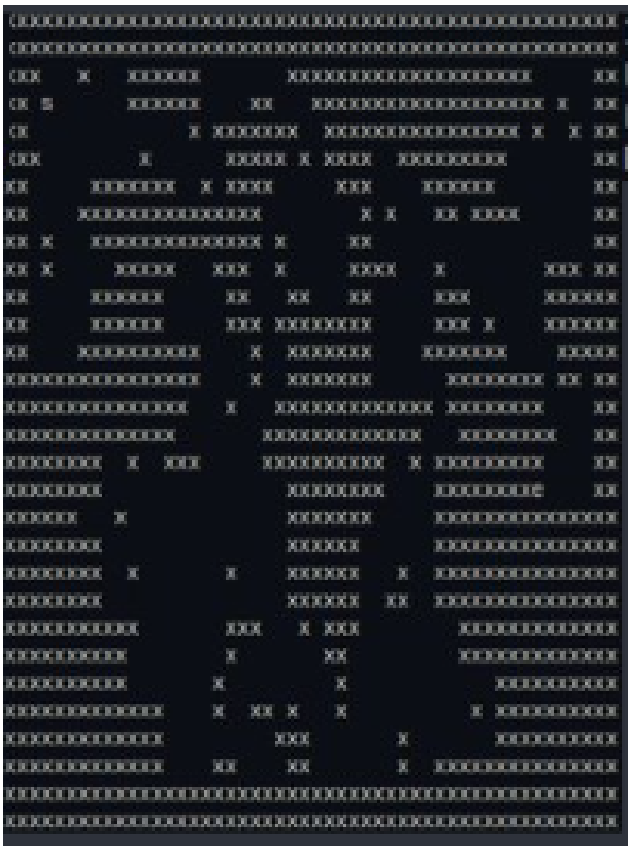

Figura 2

Segundo Mapa generado

Referencias

Koesnaedi, A., y Istiono, W. (2022). Implementation drunkard's walk algorithm to generate random level in roguelike games. *International Journal of Multidisciplinary Research and Publications (IJMRAP)*, *5*(2), 97-103. Descargado de http://ijmrap.com/wp-content/uploads/2022/07/IJMRAP-V5N2P27Y22.pdf