Generación procedural de mapas y detección de colisiones 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

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 Martínez Vilar (2015). 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. Según Román-Ibáñez (2014), se pueden distinguir cuatro niveles:

Generación offline: El mapa se genera antes de que el juego empiece y no cambia durante la partida. El desarrollador tiene un control total sobre el algoritmo y sus parámetros. Un ejemplo es Civilization. Generación online: El mapa se genera durante la partida según las acciones del jugador o las condiciones del juego. El desarrollador tiene un control parcial sobre el algoritmo y sus parámetros. Un ejemplo es Spelunky. Generación interactiva: El mapa se genera durante la partida según las preferencias o decisiones del jugador. El jugador tiene un control parcial sobre el algoritmo y sus parámetros. Un ejemplo es SimCity. Generación mixta: El mapa se genera combinando elementos predefinidos con elementos procedurales. El desarrollador y el jugador tienen un control compartido sobre el algoritmo y sus parámetros. Un ejemplo es Terraria. Para generar mapas procedurales se pueden utilizar diferentes técnicas y algoritmos según el tipo y la dimensión del mapa deseado. Algunas de las técnicas más comunes son:

Perlin Noise: Es un algoritmo que genera un ruido coherente, es decir, una función que produce valores aleatorios pero suaves y continuos. Se puede utilizar para crear mapas de altura, que son matrices bidimensionales que representan la elevación del terreno en cada punto. El ruido de Perlin se puede combinar con otros algoritmos o funciones

matemáticas para modificar la escala, la frecuencia o la amplitud del mapa generado. Un ejemplo es Minecraft. Algoritmos de Laberinto: Son algoritmos que generan laberintos, es decir, estructuras compuestas por pasillos y paredes que forman un camino único entre dos puntos. Se pueden utilizar para crear mapas de mazmorras, que son espacios cerrados y complejos que contienen enemigos, objetos y salidas. El algoritmo de Prim es uno de los más utilizados para generar laberintos, ya que parte de un grafo completo y va eliminando aristas aleatoriamente hasta formar un árbol de expansión mínimo. Un ejemplo es Rogue. Algoritmos de Voronoi: Son algoritmos que generan diagramas de Voronoi, es decir, particiones del espacio en regiones poligonales según la distancia a un conjunto de puntos. Se pueden utilizar para crear mapas de islas, que son conjuntos de tierras rodeadas por agua. El algoritmo de Lloyd es uno de los más utilizados para generar diagramas de Voronoi, ya que parte de un conjunto de puntos aleatorios y los reubica iterativamente hasta que las regiones sean lo más regulares posibles. Un ejemplo es Don't Starve. Algoritmos de Autómatas Celulares: Son algoritmos que generan patrones complejos a partir de reglas simples que se aplican a cada celda de una matriz bidimensional. Se pueden utilizar para crear mapas de cuevas, que son espacios subterráneos y irregulares que contienen rocas, minerales y agua. El algoritmo de Conway es uno de los más utilizados para generar autómatas celulares, ya que parte de una matriz aleatoria y la actualiza según el número de vecinos vivos o muertos que tenga cada celda. Un ejemplo es Dwarf Fortress. La detección de colisiones se puede definir como el proceso de determinar si dos o más objetos del juego entran en contacto o se superponen Alonso Alonso (2015). Esta técnica es fundamental para el funcionamiento y la física del juego, ya que 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 con formas irregulares o en movimiento. Por ello, se han desarrollado diversas técnicas y algoritmos para optimizar y

simplificar este proceso, como las cajas de colisión, los árboles BSP o el raycasting. Estas técnicas varían según el tipo y la dimensión de los objetos, el nivel de precisión requerido y el rendimiento de la máquina.

Para detectar colisiones se pueden utilizar diferentes técnicas y algoritmos según el tipo y la forma de los objetos. Algunas de las técnicas más

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.

Título III

El resto della concluían sayo de velarte, calzas de velludo para las fiestas, con sus pantuflos de lo mesmo, y los días de entresemana se honraba con su vellorí de lo más fino.

Título IV. Tenía en su casa una ama que pasaba de los cuarenta, y una sobrina que no llegaba a los veinte, y un mozo de campo y plaza, que así ensillaba el rocín como tomaba la podadera.

Título IV ii.

Frisaba la edad de nuestro hidalgo con los cincuenta años; era de complexión recia, seco de carnes, enjuto de rostro, gran madrugador y amigo de la caza.

Título V. Quieren decir que tenía el sobrenombre de Quijada, o Quesada, que en esto hay alguna diferencia en los autores que deste caso escriben; aunque por conjeturas verosímiles se deja entender que se llamaba Quijana.

Referencias

Alonso Alonso, J. (2015). *Videojuegos 2d*. FUOC. Descargado de https://openaccess.uoc.edu/bitstream/10609/50184/2/Videojuegos%202D.pdf ([Archivo PDF])

Martínez Vilar, R. (2015). *Videojuego basado en la generación procedimental de mundos o niveles*. Descargado de https://rua.ua.es/dspace/handle/10045/48231 ([Acceso en línea])

Román-Ibáñez, V. (2014). *Creación de videojuego roguelike*. Descargado de https://rua.ua.es/dspace/handle/10045/39667 ([Acceso en línea])