Dijkstra's algorithm in video games

Introduction

Dijkstra's algorithm is a technique used to find the shortest path between two points in a graph. In the context of video games, it is mainly used for navigating non-playable characters (NPCs), generating optimal paths, and solving movement problems in complex scenarios.

Algorithm Fundamentals

Dijkstra's algorithm operates on a weighted graph, where:

- Each node represents a location on the game map.
- Each edge represents a path between locations.
- Each weight on the edge indicates the cost of moving from one node to another (it can be distance, time, terrain difficulty, etc.).

The algorithm follows these steps:

1. All distances are initialized as infinite, except for the start node, which is set to zero.
2. The node with the shortest known distance is processed, and the distances of its neighbors are updated if a shorter path is found.
3. The process is repeated until the target node is reached or all accessible nodes are visited.
4. The optimal path is reconstructed by following the processed nodes.

Applications in Video Games

Dijkstra is fundamental in video games for his ability to find efficient routes on large and complex maps. Some of its main applications include:

1. NPC (Artificial Intelligence) Navigation

- NPCs use Dijkstra to move from one point to another avoiding obstacles.
- It is used in open-world games and real-time strategy (RTS) to optimize unit movement.

2. Routes on Open Maps

- In exploration games, the player can select a destination and the algorithm finds the shortest route.
- It is used in self-navigation systems in role-playing games (RPGs) and open worlds.

3. Difficulty Systems and Land Costs

- In strategy games, different terrains may have different movement costs (mountains, rivers, roads, etc.).
- Dijkstra allows you to calculate optimized routes considering these factors.

4. Network and Resource Path Optimization

- In management and simulation games, such as "SimCity" or "Age of Empires", the algorithm optimizes the distribution of resources.

Advantages and Disadvantages

✅ Advantages

- Guarantees to find the shortest path on maps with positive weights.
- It is accurate and deterministic, which means that it always gives the same result for the same data.
- It can be adapted to work with dynamic systems where costs change.

❌ Disadvantages

- Its performance can be slow on very large maps, especially if there are many connections.
- It does not work well in graphs with negative weights, so in those cases other algorithms such as Bellman-Ford are used.
- It can be inefficient on open maps where *A (A-Star)** is a better alternative because of its heuristic approach.

Conclusion

Dijkstra's algorithm is a powerful tool for video game development, as it allows optimal routes to be calculated efficiently. Its application in NPC navigation, open maps, and strategy systems demonstrates its versatility. However, in more complex scenarios, techniques such as *A or algorithms based on hierarchical graphs** are sometimes combined to improve performance.

In short, Dijkstra remains a key solution in virtual world design and artificial intelligence development in modern video games.