A Constraint Programming Solution for the Military Unit Path Finding Problem

Leenen Louise

Terlunen Alexander

Le Roux Herman

In this chapter we present an algorithm to solve the Dynamic Military Unit Path Finding Problem (DMUPFP) which is based on Stentz's well-known D* algorithm to solve dynamic path finding problems. The Military Unit Path Finding Problem (MUPFP) is the problem of finding a path from a starting point to a destination where a military unit has to move, or be moved, safely whilst avoiding threats and obstacles and minimising path cost in some digital representation of the actual terrain. Path finding is the problem of moving an object from a starting point to a destination point, whilst avoiding obstacles and minimising costs. It has many applications ranging from computer games, transportation, robotics, networks, and others. Path finding algorithms can be divided into two different types, static (or global) and dynamic algorithms. In a static algorithm, the environment is known and an optimal path is calculated before the object is moved. In a dynamic algorithm, the environment may not be completely known before the object starts moving, or it may change whilst the object is moving. In this case, the path plan has to be updated while it is being executed. The path finding problem has been well studied and there is a vast body of research in the literature. Section 2 contains an overview of methods to solve the path finding problem. During military operations, military units often encounter the path finding problem; they aim to avoid threats or at least pass threats with a minimum separating distance whilst attempting to move as quickly as possible. In the MUPFP the requirement is to maintain a balance between the two main criteria, route speed and safety. Although there are methods to solve the MUPFP, the existing approaches combine the optimisation of the various criteria in some form of an objective function. Our approach, on the other hand, is to optimise path costs while ensuring that certain criteria, such as safety requirements, are met. This means that our objective function is a pure cost function. We adopt a constraint-based approach with a clear distinction between the goal of obtaining an optimal cost and adherence to safety measures. Such an approach also allows for flexibility in terms of modeling different constraints. If the user has a new requirement in terms of adhering to safety measures or some other constraint on the problem, this simply entails the addition of constraints or modification of existing constraints: the main algorithm stays the same. Section 3 contains an overview of Constraint Programming (CP) and Constraint Satisfaction Problems (CSPs). We present a new algorithm to solve the Dynamic MUPFP (DMUPFP) in Section 4. The objective function only represents the cost of traversing a particular path, i.e. a representation of the route speed. We model the safety criteria as constraints which have to be satisfied. For example, a constraint can specify that obstacles must be avoided, or that an acceptable distance has to be observed when passing a threat such as a sniper. Previous work done revolved around a static constraint-based approach which was based on the A* algorithm.