tuts+

Envato Elements   Envato Studio   Community   Forum   Help   Sign Up | Sign In |

How-To Tutorials ⌄   Courses ⌄   eBooks ⌄   Pricing   Search Envato Tuts+   🔍

GAME DEVELOPMENT > ARTIFICIAL INTELLIGENCE

Finite-State Machines: Theory and Implementation

by Fernando Bevilacqua   24 Oct 2013

Difficulty: Intermediate   Length: Long   Languages:

| English ⇅ |

Artificial Intelligence | Finite State Machine | Programming | Platform Agnostic

Flash | AS3 | Algorithms

A finite-state machine is a model used to represent and control execution flow. It is perfect for implementing AI in games, producing great results without a complex code. This tutorial describes the theory, implementation and use of simple and stack-based finite-state machines.

All icons made by Lorc, and available on http://game-icons.net*.*

Note: *Although this tutorial is written using AS3 and Flash, you should be able to use the same techniques and concepts in almost any game development environment.*

What Is a Finite-State Machine?

A finite-state machine, or FSM for short, is a model of computation based on a hypothetical machine made of one or more states. Only a single state can be active at the same time, so the machine must transition from one state to another in order to perform different actions.

FSMs are commonly used to organize and represent an execution flow, which is useful to implement AI in games. The "brain" of an enemy, for instance, can be implemented using a FSM: every state represents an action, such as `attack` or `evade` :
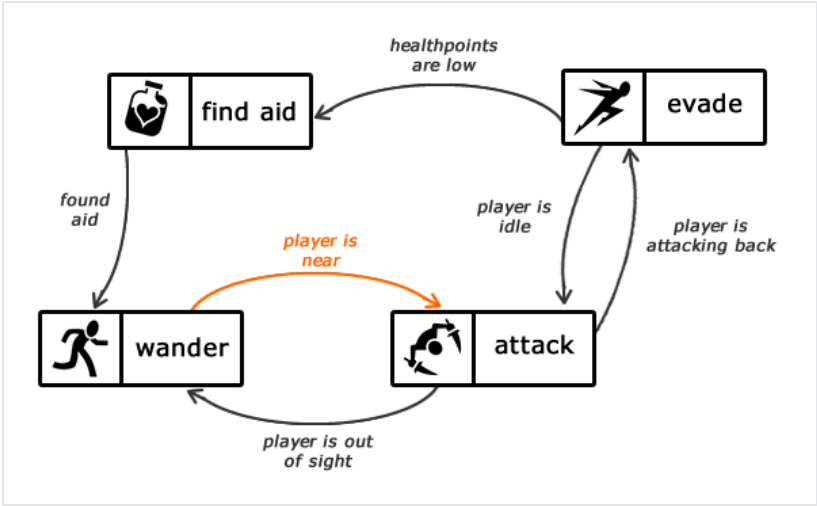
FSM representing the brain of an enemy.

An FSM can be represented by a graph, where the nodes are the states and the edges are the transitions. Each edge has a label informing when the transition should happen, like the `player is near` label in the figure above, which indicates that the machine will transition from `wander` to `attack` if the player is near.

Planning States and Their Transitions