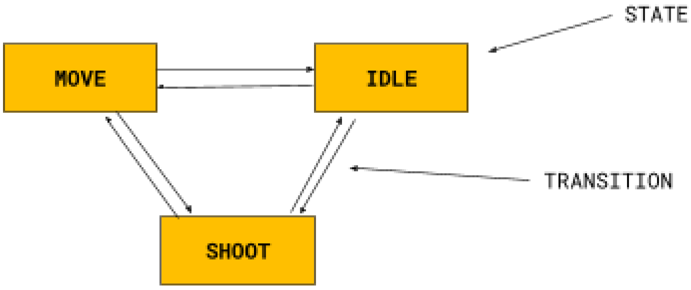
Finite State Machines

General concepts

A Finite State Machine is a system that uses States and Transitions to describe complex patterns that can be used in creating AI behavior, animations and more.

A FSM is one of the common ways to implement decision logic for the AI.

Each state represents an action (in the case of an AI) or an animation. They are interconnected by transitions and to move from one to another state there needs to be a fulfilled condition.



Behind the scenes, a FSM is a graph-like structure where its states are the nodes and its transitions, the graph's edges.

Finite State Machines are one of the first ways to implement AI Agents that appeared. It’s commonly used today as it fits a lot of cases. It does have PROs/CONs.

PROs:

- Easy to use & understand
- Fast

CONs:

- Can get out of hand quickly if a lot of states are added
- It is hard to maintain for complex structures
- It is not reusable, as its parts are tightly linked together

There are ways to mitigate these issues, using Hierarchical State Machines. The main difference between the classic ones is that the Hierarchical ones have nested states inside states. Let’s call them state ception.

For the final project we will use a finite state machine from a library but for the next example we will implement a simple one from scratch.

It’s important to implement a FSM to understand its inner workings. Let’s find them out right now.