Finite state machine...

...and some retrogaming

Gabriele Falasca
FOSDEM 2023

What is it

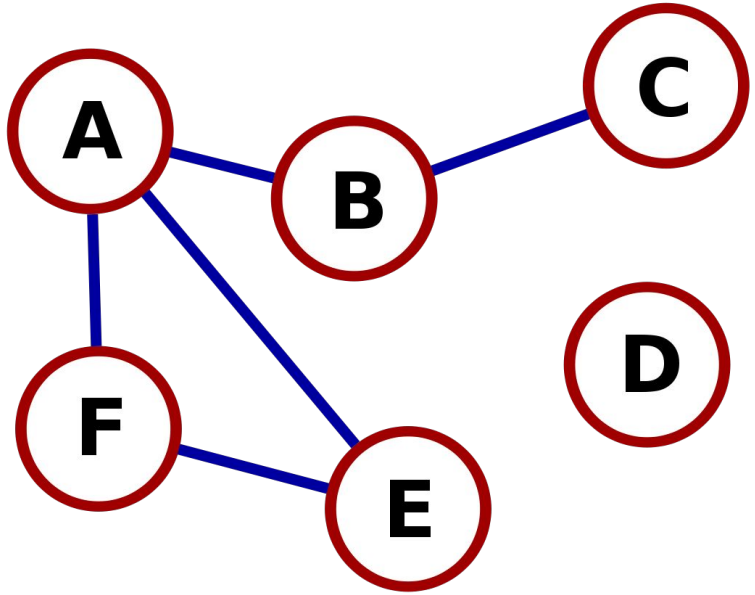
It is an abstract machine that can be in exactly one of a finite states at any given time.

It can change from a state to another in response to some inputs.

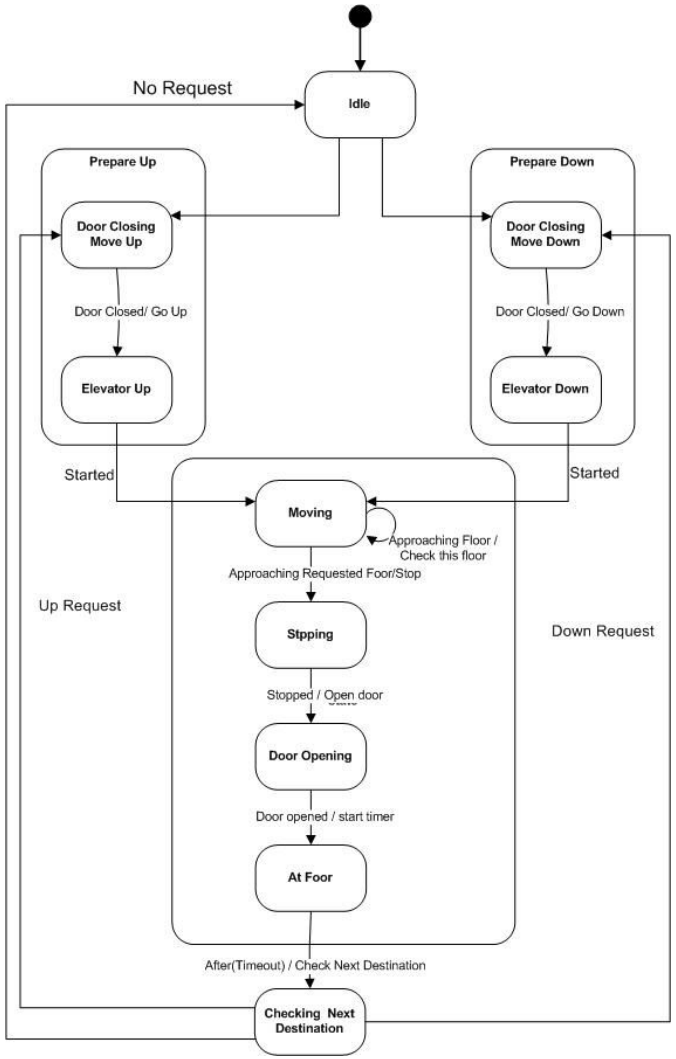
credits: Wikipedia

Statecharts

A FSM can be represented by a connected graph, called statechart where the nodes are the states, and the links are the transitions.



Example /elevator



Let's create a statechart

Create statechart / states

Idle

Create statechart / states

Idle

Boxer

Create statechart / states

Idle

Boxer

Military

Create statechart / transitions

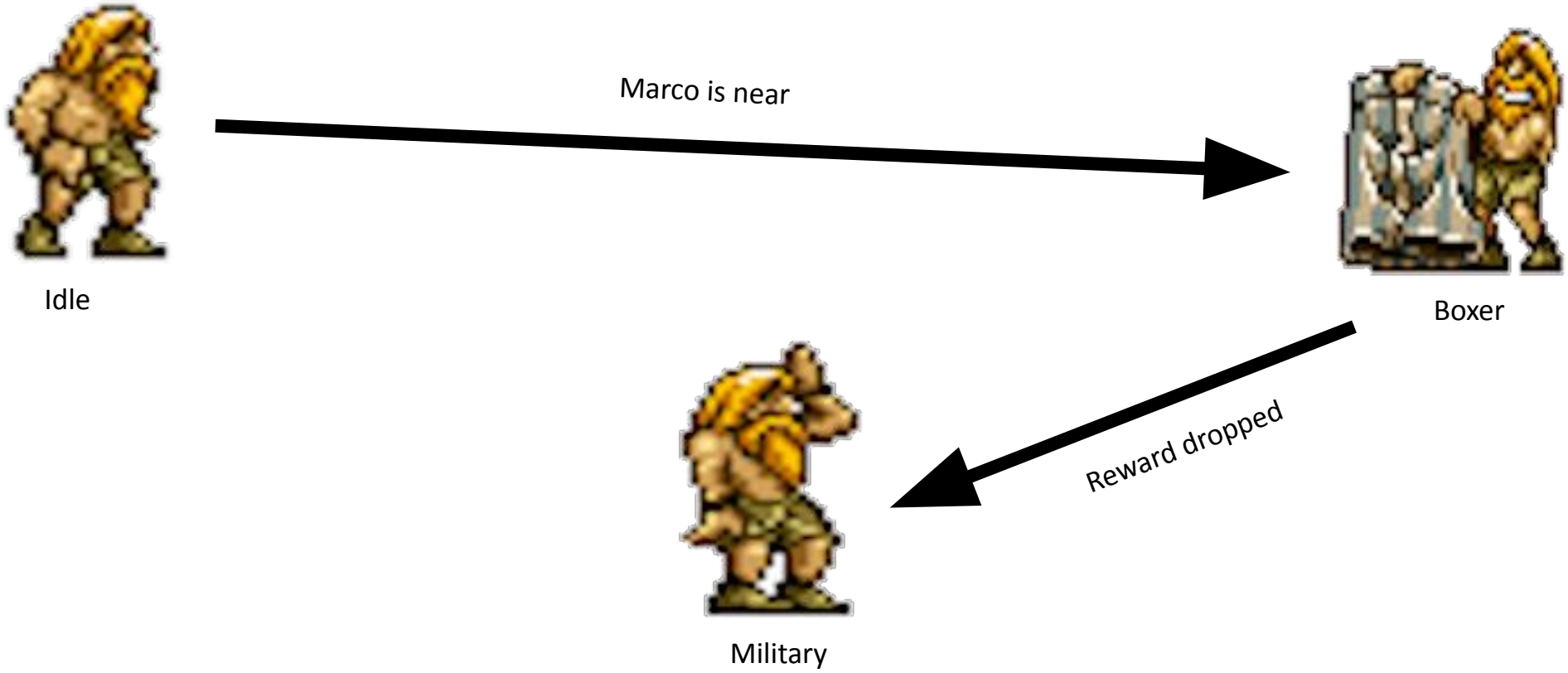
Idle

Marco is near

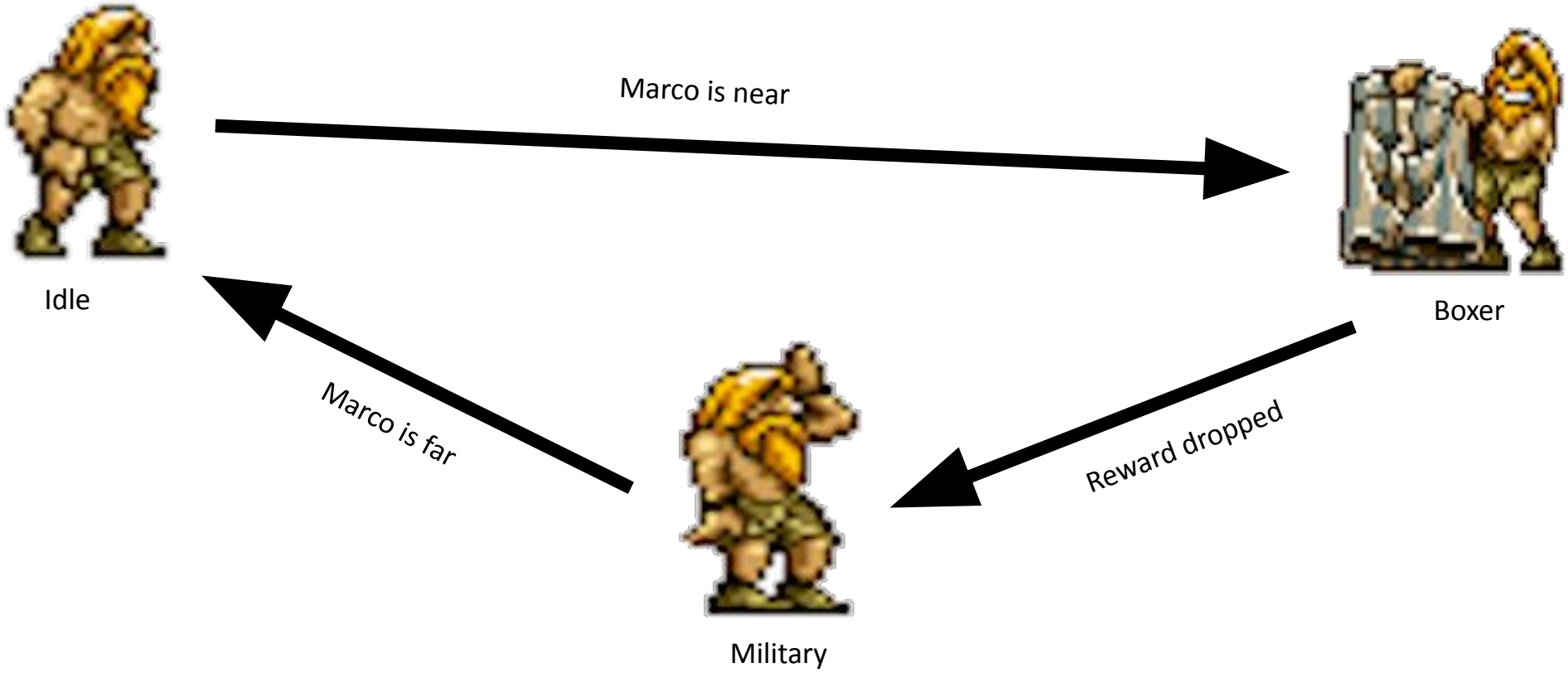
Boxer

Military

Create statechart / transitions



Create statechart / transitions



Simplest FSM code

```
// javascript

class Fsm {

  setState = (state) => {
    this.activeState = state; // activeState must be a function!
  }

  update = () => {
    if(this.activeState) {
      this.activeState();
    }
  }
}

export default Fsm;

```

Stack based FSM

- Stack of states instead of active state
- Active state is the one on top of the stack
- Every state must pop itself from the stack at the right time

Stack based FSM / code

```
// javascript

class FsmStack {
  constructor() {
    this.stack = [];
  }

  popState = () => this.stack.pop();

  pushState = (state) => this.stack.push(state);

  currentState = () => this.stack[0];

  update = () => {
    const active = this.currentState();
    if (active) {
      active();
    }
  }
}
```

xstate

Javascript and Typescript finite state machines and statecharts for modern web

xstate

```
import { createMachine, interpret } from 'xstate';

// Stateless machine definition
// machine.transition(...) is a pure function used by the interpreter.
const toggleMachine = createMachine({
  id: 'toggle',
  initial: 'inactive',
  states: {
    inactive: { on: { TOGGLE: 'active' } },
    active: { on: { TOGGLE: 'inactive' } }
  }
});

// Machine instance with internal state
const toggleService = interpret(toggleMachine)
  .onTransition(state => console.log(state.value))
  .start();
// => 'inactive'

toggleService.send('TOGGLE');
// => 'active'

toggleService.send('TOGGLE');
// => 'inactive'

```

xstate viz

Code to visual statechart visualizer. (let's see on the site)

Questions?

Gabriele Falasca

Frontend dev @Sourcesense

Mozilla Tech Speaker

@gabrycaos