

Kickstart-kursus i programmering 23 dag 4

Finite States

Daniel Spikol
ds@di.ku.dk

DIKU
Københavns Universitet

17 august 2023

Recap from Wednesday

- Scoping
- Conditionals
- Projects

Thursday IFOs

- Finite State Machines
- Wrapping up Things
- Projects and Demos

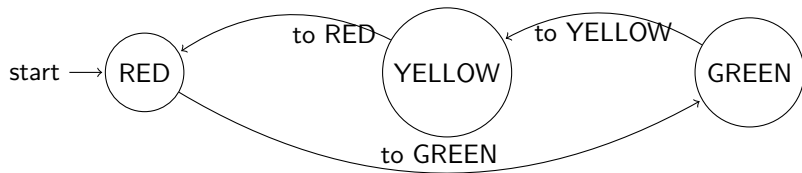
Finite State Machines- FSM

A Finite State Machine (FSM) is a mathematical model of computation used to design algorithms. In the context of computer games, FSMs are often used for character behaviour, where different states might represent actions like “dle”, “attack”, “defend”, or “flee”, and game events or conditions determine transitions between states.

The STATES in a FSM

- **Discrete States:** An FSM consists of a limited or finite number of states. It can be in just one of these states at any given moment. Transitions define how it changes from one state to another based on inputs or conditions.
- **Transitions & Triggers:** Events or conditions trigger transitions between states. Each state specifies which state the machine will move to next for each possible input.
- **Start and End States:** Among the finite states, there is one initial state where the FSM begins its operation. Additionally, there can be one or more end states where the FSM is considered to be completed or final.

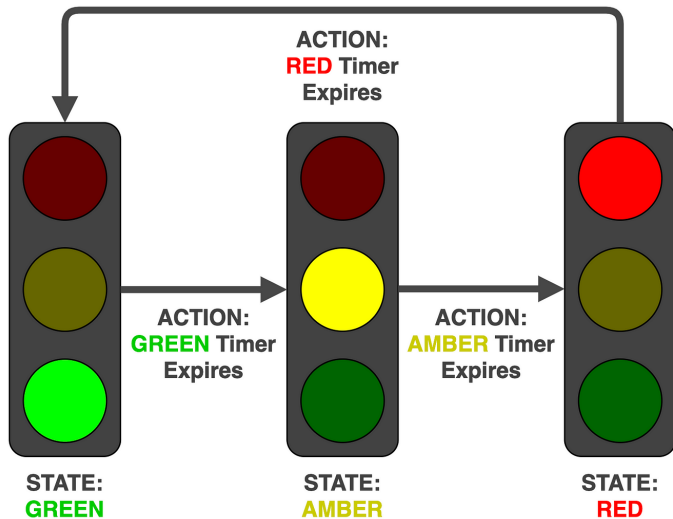
The Classic Example



Mathematical Abstraction of the FSM

- A finite state machine is a mathematical abstraction used to design algorithms. In simple terms, a state machine will read a series of inputs.
- When it reads an input, it will switch to a different state. Each state specifies which state to switch to for a given input.

IDEAL Problem Solving



Code FSM

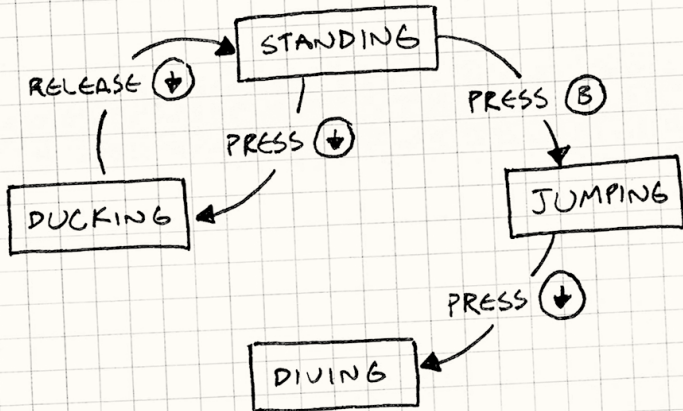
```
1 # States for our FSM
2 STATE_RED = "RED"
3 STATE_GREEN = "GREEN"
4
5 # Initial state
6 current_state = STATE_RED
7
8 def setup():
9     size(200, 200)
10    fill(255)
11
12 def draw():
13     background(200)
14
15     if current_state == STATE_RED:
16         fill(255, 0, 0) # Red color for RED state
17     elif current_state == STATE_GREEN:
18         fill(0, 255, 0) # Green color for GREEN state
19
20     ellipse(width/2, height/2, 100, 100)
21
22 def mousePressed():
23     global current_state
24     if current_state == STATE_RED:
25         current_state = STATE_GREEN
26     elif current_state == STATE_GREEN:
27         current_state = STATE_RED
```

We Code a bit

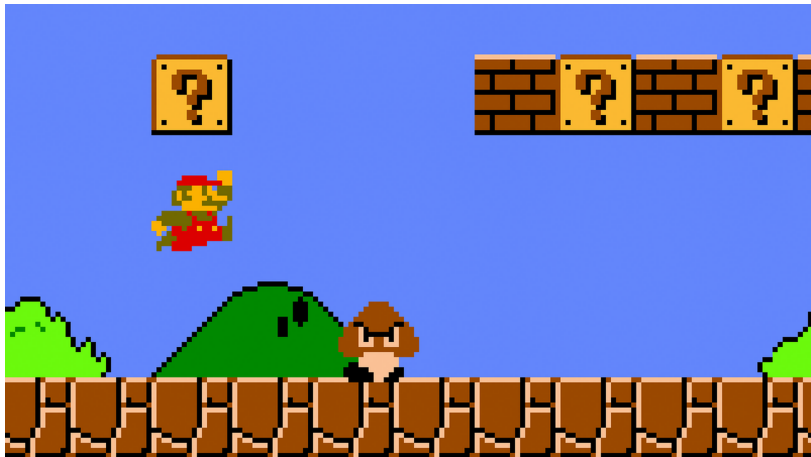
We make a Traffic Light

- How do we add the Yellow?
- Let's do it together...

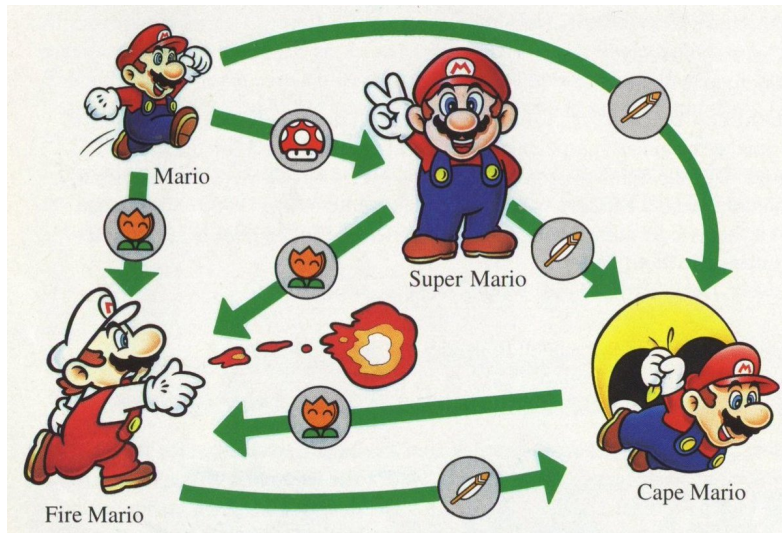
Game States



IDEAL Problem Solving



IDEAL Problem Solving



Game States

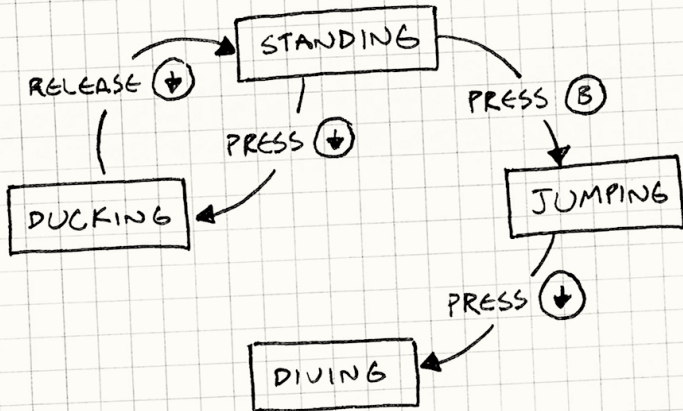
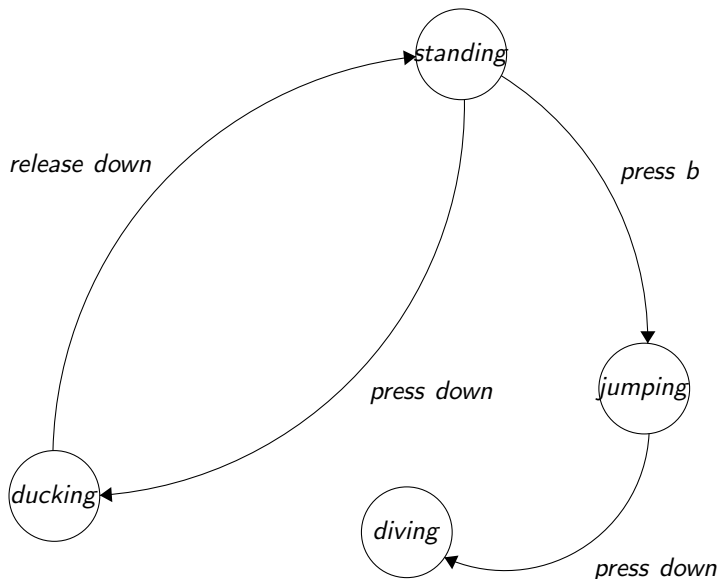


Diagram It



Let's make some FSM diagrams

<https://markusfeng.com/projects/graph/>

Today' Recap

- Conditionals
- FSM
- Projects

Tomorrow

- Morning code
- Afternoon Show and Talk