Hello Thursday

Kickstart-kursus i programmering dag 4

Daniel Spikol

ds@di.ku.dk

DIKU \ Københavns Universitet

14. august 2025

Recap from Wednesday

- Scoping
- Conditionals
- Projects

Thursday IFOs

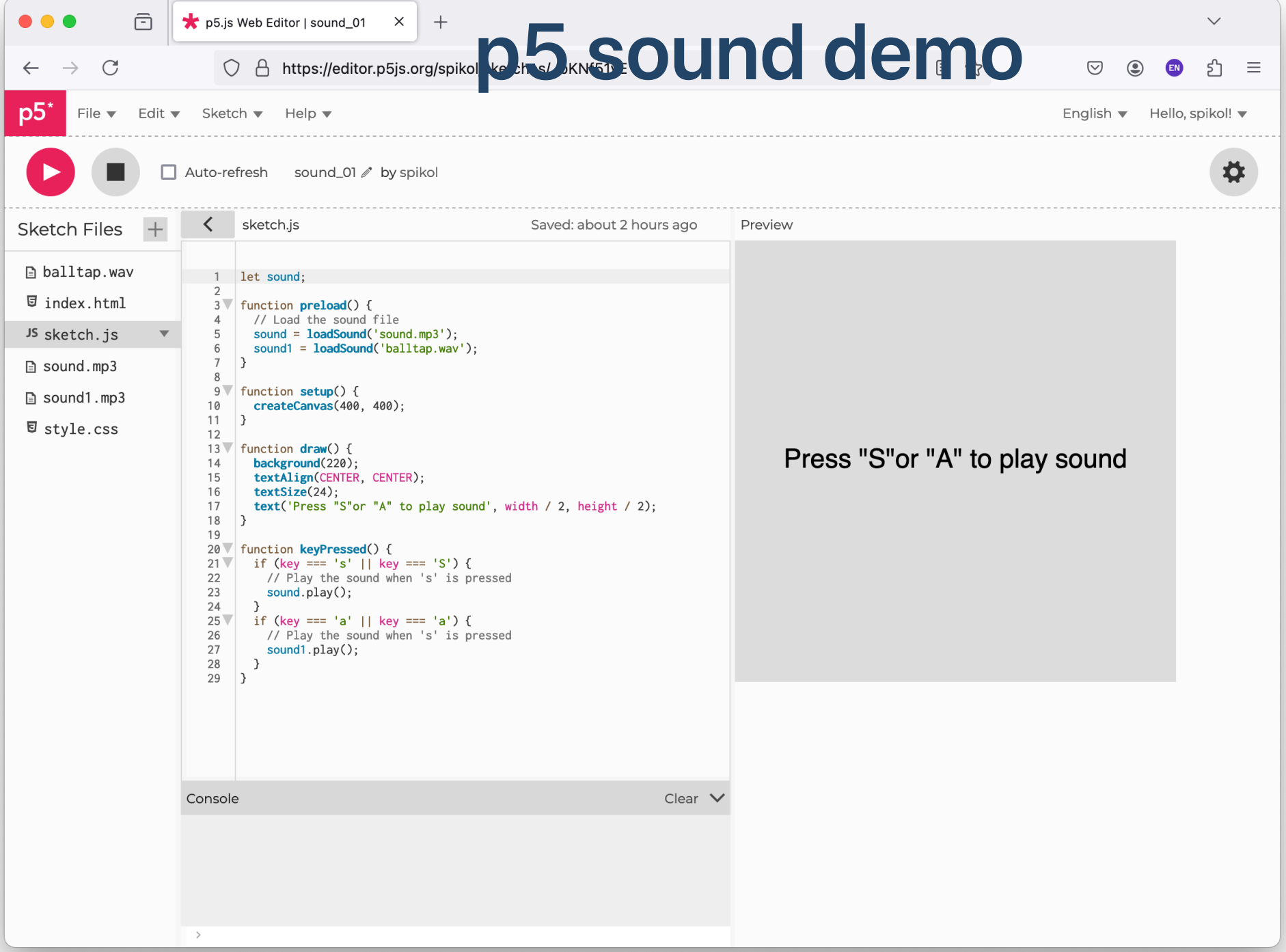
- Sounds
- Finite State Machines
- Wrapping up Things
- Projects and Demos

Sounds

Sound in p5.js

- Let's keep it simple, but if interested explore the p5 reference
- We are going to load and play sounds, plenty other ways to work with sound.
- SoundFile object with a path to a file.
- The p5.SoundFile may not be available immediately because it loads the file information asynchronously.
- To do something with the sound as soon as it loads pass the name of a function as the second parameter.
- Only one file path is required. However, audio file formats (i.e. mp3, ogg, wav and m4a/aac) are not supported by all web browsers.

p5 sound demo



Code

```
function preload() {
  // Load the sound file
  sound = loadSound('sound.mp3');
  sound1 = loadSound('balltap.wav');
}

function setup() {
  createCanvas(400, 400);
}

function draw() {
  background(220);
  textAlign(CENTER, CENTER);
  textSize(24);
  text('Press "S"or "A" to play sound', width / 2, height / 2);
}

function keyPressed() {
  if (key === 's' || key === 'S') {
    // Play the sound when 's' is pressed
    sound.play();
  }
  if (key === 'a' || key === 'a') {
    // Play the sound when 's' is pressed
    sound1.play();
  }
}
```

Finite State Machines- FSM

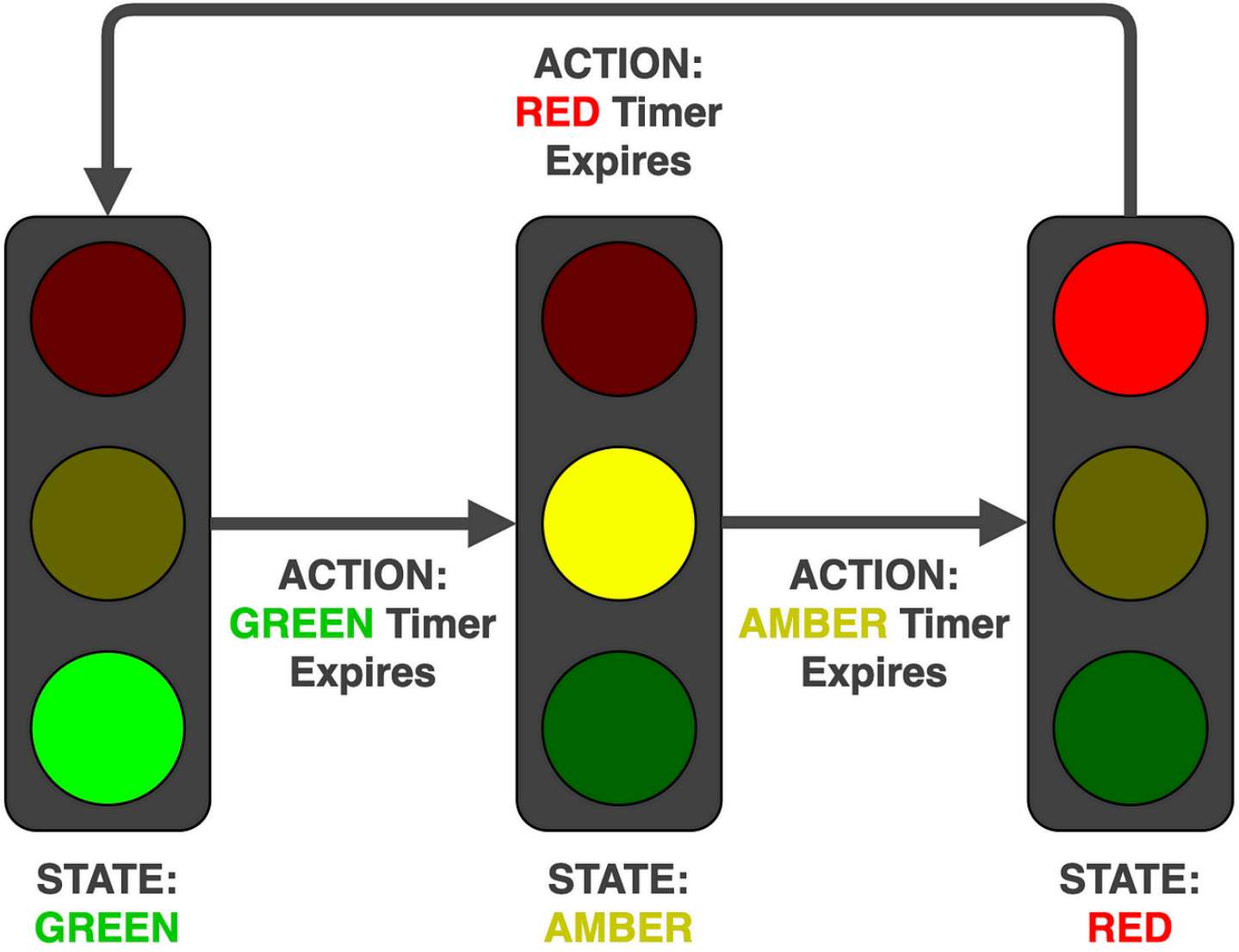
- A Finite State Machine (FSM) is a mathematical model of computation used to design algorithms.
- In the context of computer games, FSMs are often used for character behaviour, where different states might represent actions like "idle", "attack", "defend", or "flee", and game events or conditions determine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

Finite State Machines

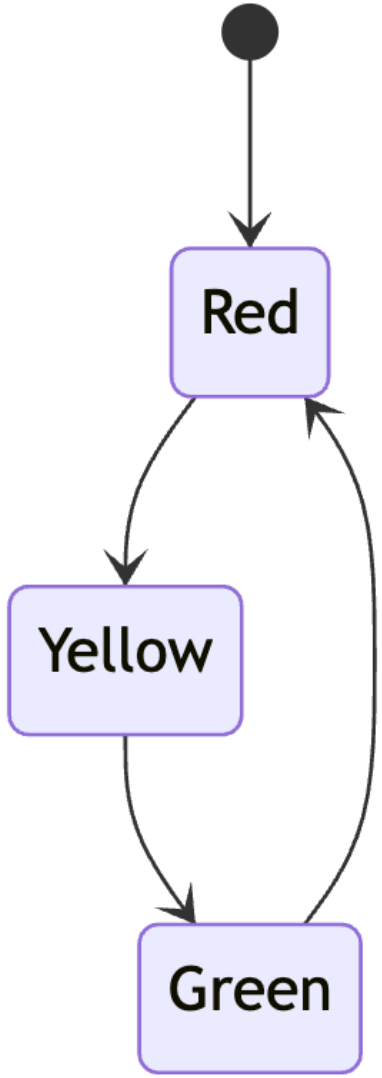
The STATES in a FSM

- **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



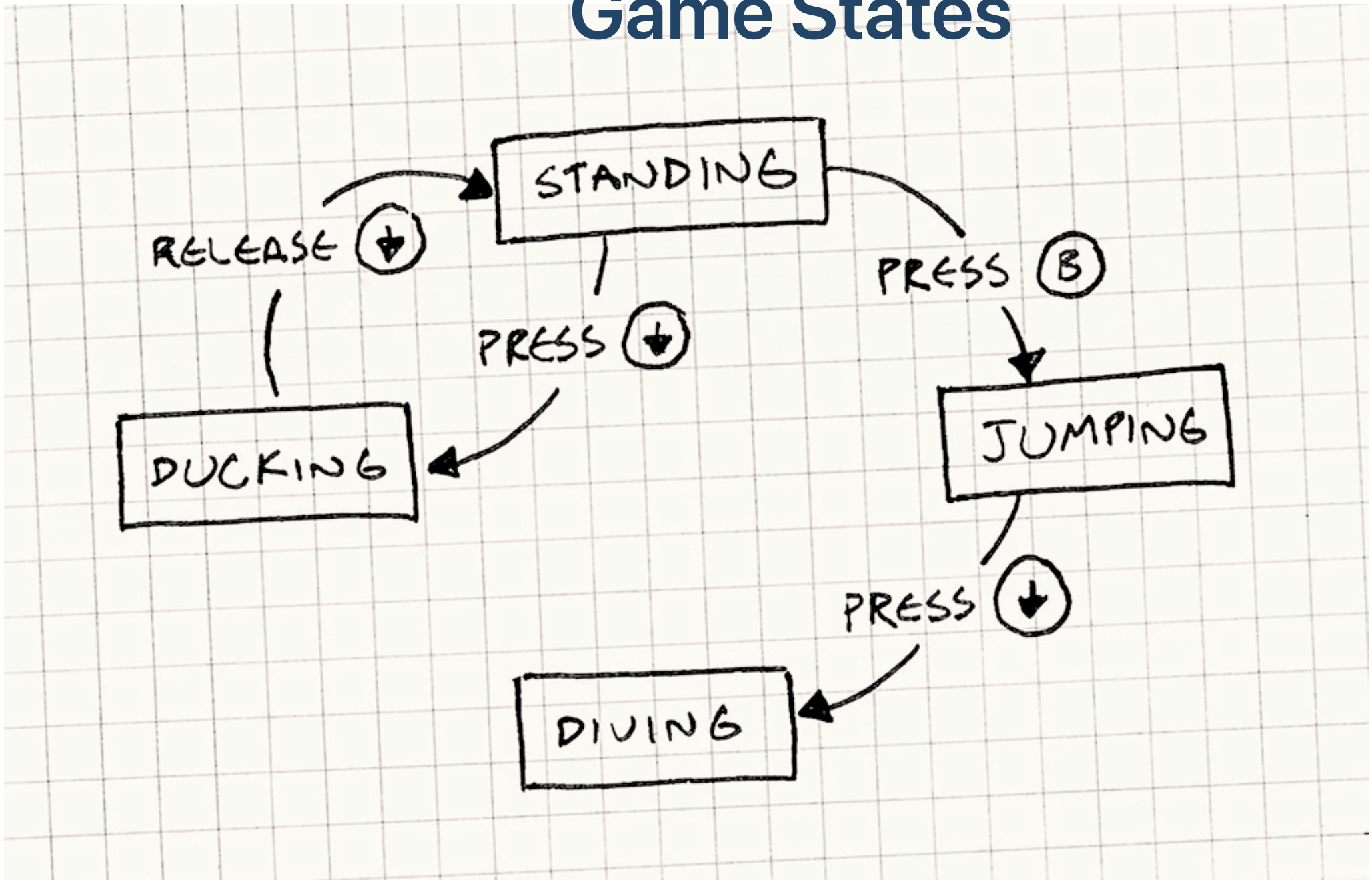
State Diagram



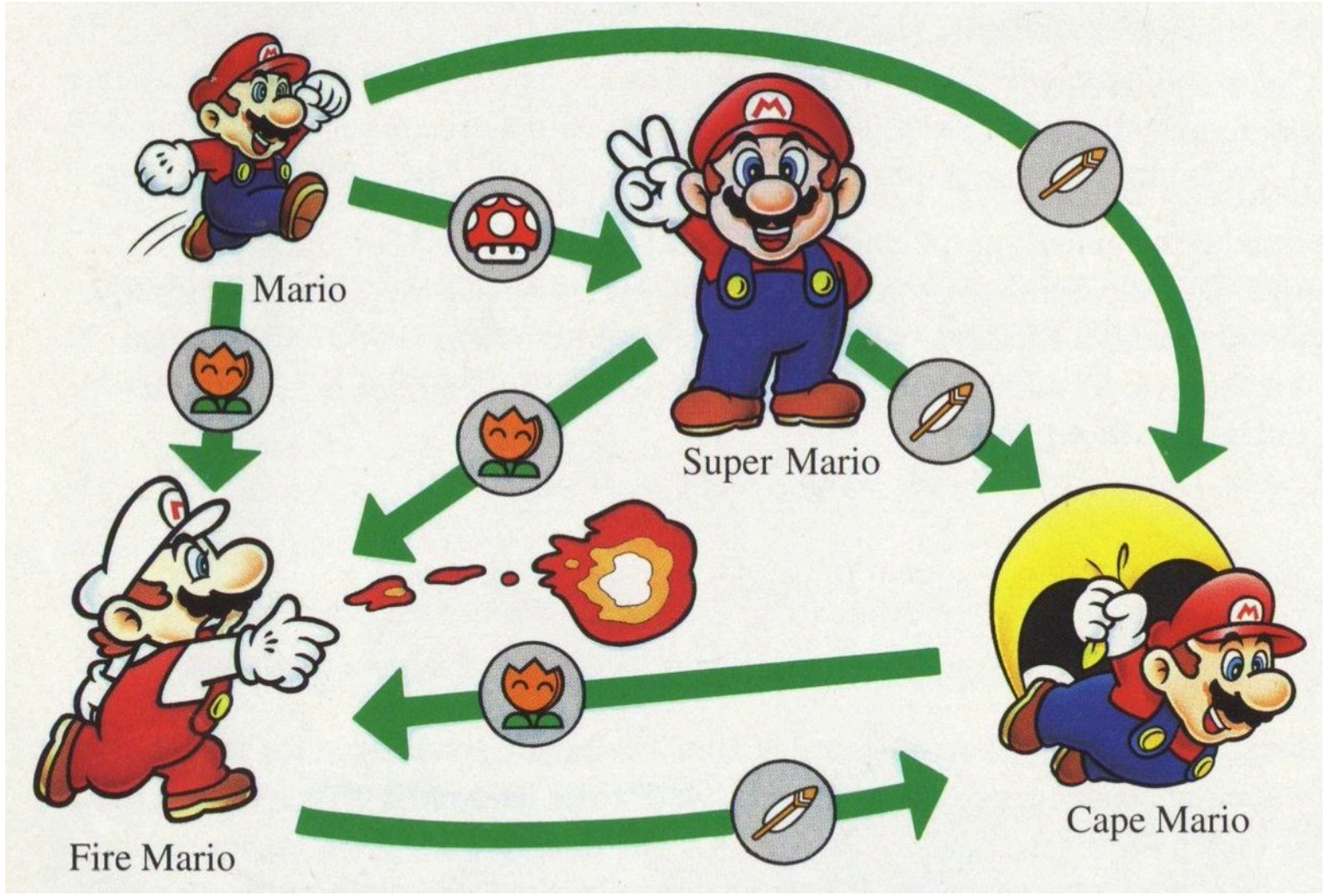
Mathematical Abstraction of the FSM

- 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.-

Game States


STANDING
RELEASE
PRESS
PRESS
B
DUCKING
JUMPING
PRESS
DIVING

Mario States



FSM Code Example

```
//global vars
var state_on = "ON";
var state_off = "OFF";

//initial state
current_state = state_off;

function setup() {
  createCanvas(400, 400);
}

function draw() {
  background(220);

  if (current_state == state_on) {
    fill(0, 255, 0); // green is for on
  } else if (current_state == state_off) {
    fill(255, 0, 0); // red is for off
  }
  ellipse(width / 2, height / 2, 100, 100);
}

function mousePressed() {
  if (current_state == state_off) {
    current_state = state_on;
  } else if (current_state == state_on) {
    current_state = state_off;
  }
}
```

How do you add Yellow?