

## 1 โครงสร้างการทำงานของเกม (Game Architecture)

เกมนี้เป็น Web-based Game ใช้

HTML + CSS + JavaScript (Vanilla JS)

ทำงานทั้งหมดฝั่ง Client (ไม่ต้องมี Server)

### 1.1 โครงสร้างหลักของไฟล์

HTML

- โครงสร้างหน้าเกม (grid, canvas, modal)
- ปุ่มควบคุม
- script (Game Logic)

CSS

- Layout และ UI
- Animation / Highlight
- Modal ชชนะเกม

JavaScript

- การจัดการกระดานเกม
- ตรรกะการจับคู่ (Path Finding)
- การวาดเส้นเชื่อม
- การตรวจชนะ
- ระบบเสียง (Web Audio API)

## 2 โครงสร้าง HTML (View Layer)

องค์ประกอบหลัก

```
<div class="game">
  <canvas id="lineCanvas"></canvas> <!-- วาดเส้นเชื่อม -->
  <div class="grid" id="grid"></div> <!-- กระดานเกม -->
</div>
```

```
<button>เริ่มเกมใหม่</button>
```

```
<div id="modal">คุณชนะ</div>
```

หน้าที่

Element

หน้าที่

.grid แสดงการ์ด Emoji 8x8

Element	หน้าที่
.card	ช่องแต่ละช่องของเกม
canvas	วาดเส้นเชื่อมเมื่อจับคู่ได้
modal	แจ้งเตือนเมื่อชนะเกม
button	รีเซ็ตเกม

---

## 3 โครงสร้าง CSS (Presentation Layer)

### หน้าที่หลัก

- จัด layout แบบ grid
- กำหนดสี พื้นหลัง เอฟเฟกต์
- แสดงสถานะของการ์ด

### State ของการ์ด

```
.card          /* ปกติ */
.card.selected /* ถูกเลือก */
.card.matched  /* จับคู่แล้ว (ซ่อน) */
```

---

## 4 โครงสร้าง JavaScript (Game Logic Layer)

### 4.1 ตัวแปรหลัก (Game State)

```
board = [[]] // กระดานเกม 2 มิติ
selected = [] // การ์ดที่ถูกเลือก (สูงสุด 2 ใบ)
```

ตัวแปร	ความหมาย
board[r][c]	emoji หรือ null (ถ้าจับคู่แล้ว)
selected	การ์ดที่ผู้เล่นเลือก

---

### 4.2 วงจรชีวิตของเกม (Game Flow)

```
startGame()
  ↓
แสดงกระดาน
  ↓
ผู้เล่นคลิกการ์ด
  ↓
selectCard()
  ↓
```

```
checkMatch()
  ↓
  ┌─ จับคู่ไม่ได้ → reset()
  └─ จับคู่ได้
    ↓
    drawPath()
    ↓
    ลบการ์ดออกจาก board
    ↓
    checkWin()
```

---

### 4.3 การเริ่มเกม

startGame()

หน้าที่:

- ทุ่ม emoji
  - สร้าง DOM `.card`
  - reset state ทั้งหมด
- 

### 4.4 การเลือกการ์ด

selectCard(card)

เงื่อนไข:

- เลือกได้เฉพาะการ์ดที่ยังไม่ถูกจับคู่
  - เลือกได้ครั้งละ 2 ใบ
- 

### 4.5 การตรวจจับคู่ (Matching Logic)

checkMatch()

ตรวจ:

1. emoji เหมือนกันหรือไม่
  2. มีเส้นเชื่อมถูกต้องหรือไม่ (Onet rule)
-

## 5 ทรรกะการเชื่อมเส้น (Path Finding)

รองรับการเชื่อมสูงสุด 2 มุม (3 เส้น)

### 5.1 เส้นตรง (0 มุม)

A ————— B

### 5.2 1 มุม

A |  
B

### 5.3 2 มุม (รวมอ้อมขอบกระดาน)

A ┌————┐ B

ฟังก์ชันสำคัญ:

```
straight()  
findPath()  
isEmpty()
```

---

## 6 Canvas Drawing System

drawPath(path)

หน้าที่:

- วัดเส้นระหว่างช่องที่จับคู่
  - ใช้ canvas ซ้อนบน grid
  - ลบเส้นเมื่อแสดงเสร็จ
- 

## 7 ระบบเสียง (Audio System)

ใช้ Web Audio API

เหตุการณ์ เสียง

จับคู่สำเร็จ โน้ตสั้น

ชนะเกม Melody สั้น

ไม่มีการใช้ไฟล์เสียง → สร้างเสียงแบบ Real-time

---

## 8 การตรวจสอบชนะเกม

checkWin()

เงื่อนไข:

- board[][] ทุกช่องเป็น null

เมื่อชนะ:

- เล่นเมโลดี้
  - แสดง modal
- 

## 9 ขอบเขตของเกม (Game Scope)

### 9.1 สิ่งของเกม “ทำได้”


- เกมจับคู่ Emoji แบบ Onet
  - กระดาน 8x8
  - เส้นเชื่อมตามกติกา Onet
  - เสียงประกอบแบบ Real-time
  - แจ้งเตือนชนะเกม
  - เริ่มเกมใหม่ได้ทันที
- 

### 9.2 สิ่ง “ยังไม่อยู่ในขอบเขต”

- ระบบคะแนน
  - เวลา / Countdown
  - ระบบใบ้ (Hint)
  - การสับไฟเมื่อไม่มีทางเล่น
  - ด่าน / Level
  - Multiplayer / Online
  - บันทึกคะแนน / Database
-

## 10 เหมาะกับการใช้ในบริบทใด

เหมาะสำหรับ:

-  โปรเจกต์เรียน Web / JavaScript
-  เกมตัวอย่าง (Prototype)
-  อธิบายอัลกอริทึม path finding
-  ฟังก์ชัน logic + event-driven programming